This is the Pre-Published Version.

WATER RESOURCES RESEARCH, VOL. 45, W08432, doi:10.1029/2007WR006737, 2009

Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques

C. L. Wu, K. W. Chau* and Y. S. Li

Dept. of Civil and Structural Engineering, Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong, People's Republic of China

*Email: cekwchau@polyu.edu.hk

ABSTRACT

In this paper, the accuracy performance of monthly streamflow forecast was discussed when employing data-driven modeling techniques on the streamflow series. A Crisp Distributed Support Vectors Regression (CDSVR) model was proposed for monthly streamflow prediction in comparison with four other models, Auto-Regressive Moving Average (ARMA), K-Nearest-Neighbors (KNN), Artificial Neural Networks (ANNs), and Crisp Distributed Artificial Neural Networks (CDANN). With respect to distributed models of CDSVR and CDANN, the Fuzzy-C-Means (FCM) clustering technique first split the flow data into three subsets (low, medium and high levels) according to the magnitudes of the data, and then three single SVRs (or ANNs) were fitted to three subsets. This paper gave a detailed analysis on reconstruction of dynamics which was used to identify the configuration of all models except for ARMA. To improve the model performance, the data preprocessing techniques of Singular Spectrum Analysis (SSA) and/or Moving Average (MA) were coupled with all five models. Some discussions were presented (1) on the number of neighbors in KNN; (2) on the configuration of ANN; and (3) on the investigation of effects of MA and SSA. Two streamflow series from different locations of People's Republic of China (Xiangjiaba and Danjiangkou) were applied for the analysis of forecasting. Forecasts were conducted at four different horizons (one-, three-, six-, and twelve-month-ahead forecasts). The results showed that models fed by pre-processed data performed better than models fed by original data, and CDSVR outperformed other models except for at six-month-ahead horizon for Danjiangkou. For the perspective of streamflow series, the SSA exhibited better effects on Danjingkou data because its raw discharge series was more complex than the discharge of Xiangjiaba. The MA considerably improved the performance of ANN, CDANN, and CDSVR by adjusting the correlation relationship between input components and output of models. It can be also found that the performance of CDSVR deteriorated with the increase of the forecast horizon.

KEYWORDS

Monthly streamflow forecast, Distributed Support Vector Regression, Reconstruction of dynamics, Singular Spectrum Analysis, False Nearest Neighbors, Moving average, Fuzzy-C-Means

1. Introduction

As an important issue in hydrology time series, streamflow (or discharge) series prediction was of great concern over the past few decades. Numerous data-driven modeling techniques were proposed for the forecast and simulation of the streamflow series (*Carlson et al., 1970; Jain et al. 1999; cannas, et al., 2006; Wang et al, 2006b; Lin et al., 2006*). The linear autoregressive models were based on the assumption that hydrological time series is originated from a stochastic process with an infinite number of degrees of freedom. Linear models such as AutoRegressive (AR), AutoRegressive Moving Average (ARMA), AutoRegressive Integrated Moving Average (ARIMA), and Seasonal ARIMA (SARIMA) had made a great success in river flow prediction (*Carlson et al., 1970; Salas et al., 1985; Haltiner and Salas, 1988; Yu and Tseng, 1996; Kothyari and Singh, 1999; Huang et al., 2004; Mar*í*a et al., 2004*). Meanwhile, chaos-based streamflow predictions have gained increasing interests of the hydrology community in the past two decades (*Jayawardena and Lai, 1994; Jayawardena and Gurung, 2000; Elshorbagy et al., 2002; Sivakumar et al.,2002*) in which a streamflow series was assumed to be derived from a nonlinear deterministic dynamic system. The local prediction technique of K-Nearest-Neighbors (KNN) algorithm was generally used (*Farmer and Sidorowich, 1987*). However, some doubts have been raised in terms of the existence of chaos in hydrologic data (*Ghilardi and Rosso, 1990; Koutsoyiannis and Pachakis, 1996; Pasternack, 1999; Schertzer et al., 2002; Wang et al., 2006a*).

Both the above two assumptions tends to be absolute descriptions for a hydrologic streamflow series. Salas et al. (*1985*) suggested that a streamflow process should be treated as an integration of stochastic (or random) and deterministic components. Describing it as either a totally linear stochastic process or fully nonlinear deterministic chaos was not a practical approach (*Elshorbagy et al. 2002*). The model based on either of two assumptions may not be the most suitable. Therefore, various soft computing methods such as Artificial Neural Networks (ANNs) and Support Vector Regression (SVR) have been widely employed to capture the nonlinearity in a streamflow series. For example, comparisons between ANNs and AR approaches appeared in the work of Raman and Sunilkumar (*1995*), Thirumalaiah and Deo (*2000*), and Kişi (*2003; 2005*). Comparisons between ANNs and ARMA were performed in Jain et al. (*1999*), Abrahart and See (*2002*), and Castellano-Me´ndeza et al. (*2004*). Most of these studies proved that ANNs outperform traditional statistical techniques. Lin et al. (*2006*) proposed to use a SVR model to predict long-term monthly flow discharge series. It was demonstrated that SVR exhibited a better performance than ARMA. Comparison studies between SVR and ANN were also conducted in literature. Dibike et al. (*2001*) and Behzad et al. (*2009*) drew similar conclusions that SVR outperformed ANN in modeling rainfall-runoff process. Laio et al. (*2003*) undertook a comparison of KNN and ANN for flood predictions and found that KNN performed slightly better at short forecast period while the situation was reversed for longer duration. Yu et al. (*2004*) proposed a scheme that combined chaos theory and SVR to predict the daily discharge. Their results showed that KNN performed worse in comparison with ARIMA and SVR. However, Sivakumar et al. (*2002*) found that the performance of the Phase-Space Reconstruction (PSR) approach was consistently better than that of ANN in short-term river flow prediction. Obviously, it is difficult to justify which modeling technique is more suitable for a streamflow forecast from the results in the literature.

It is generally accepted that the underlying mechanisms of streamflow generation are likely to be quite different during low, medium and high flow periods. For instance, the base flow mainly contributes to low-flow events whereas intense storm rainfall gives rise to high-flow events. A single global ANN model could not predict the high- and low-runoff events satisfactorily (*Minns and Hall, 1996*). Modular models were therefore proposed where sub-processes are first of all identified and then separate models (also called local or expert model) are established for each of them (*Solomatine, and Ostfeld, 2008*). Depending on the split of training data being soft or crisp, modular models are different. The soft split means the dataset can be overlapped and the overall forecasting output is the weighted-average of each local model (*Zhang and Govindaraju, 2000; Shrestha and Solomatine, 2006; Wang et al.,2006b; Wu et al., 2008*). Zhang and Govindaraju (*2000*) examined the performance of modular networks in predicting monthly discharges based on the Bayesian concept. Wu et al. (*2008*) employed a distributed SVR (DSVR) for daily river stage prediction. The DSVR was a modified version of the DSVR designed by Cheng et al. (*2006*) in which the final output is the weighted sum of the outputs of each single SVR (hereafter referred to as Soft DSVR (SDSVR)). On the contrary, there is not any overlap of data in the crisp split and the final forecasting output is based on one of local models (*Corzo and Solomatine, 2007; Jain and Srinivasulu, 2006; See and Openshaw, 2000; Sivapragasam and Liong, 2005; Solomatine and Xue, 2004*). Solomatine and Xue (*2004*) used M5 model trees and neural networks in a flood-forecasting problem. Sivapragasam and Liong (*2005*) divided the flow range into three regions, and employed different SVR models to predict daily flows in high, medium and low regions. In fact, ANN or SVR cannot extrapolate beyond the range of the data used for training. Poor forecasts/predictions are expected when a new input data is outside of the range of those used for training. This problem in the case of SVR was indeed noticed by Wu et al. (*2008*). The DSVR derived from crisp data split is therefore proposed (hereafter referred to as Crisp DSVR (CDSVR)) to overcome the poor extrapolation in the SDSVR. Similarly, the improved DANN is called Crisp DANN (hereafter referred to as CDANN).

Suitable data preprocessing can improve the performance of data-driven models. Besides the conventional rescaling or standardization of training data used in ANN models (*Dawson and Wilby, 2001*), preprocessing methods from the perspective of signal analysis are also crucial because streamflow series may be also viewed as a quasi-periodic signal, which is contaminated by various noises at different flow levels. An important signal decomposition technique, singular spectrum analysis (SSA) was recently introduced to hydrology field by some researchers (*Lisi et al., 1995; Sivapragasam et al., 2001; Marques et al., 2006*). An analysis technique for the series may be particularly significant if it is able to expose important characteristics of the time series in order to attain predictability (*Marques et al., 2006*). The SSA method decomposes a time series into a number of components with simpler structures, such as a slowly varying trend, oscillations and noise. SSA uses the basis functions characterized by data-adaptive nature, which makes the approach suitable for the analysis of some nonlinear dynamics (*Elsner and Tsonis, 1997*). In addition, the issue of lagged predictions in the ANN model has been mentioned by some researchers (*Dawson and Wilby, 1999; Jian and Srinivasulu, 2004; de Vos and Rientjes, 2005; Muttil and Chau, 2006*). One of the reasons for lagged predictions is the use of previous observed streamflow data as ANN inputs (*de Vos* *and Rientjes, 2005*). An effective solution is to obtain new inputs of models by moving average over the original streamflow series.

The purpose of this study is to investigate the performance of five data-driven models of ARMA, KNN, ANN, CDANN and CDSVR with data-preprocessing techniques of SSA and MA on two real monthly streamflow series. This paper is organized in the following manner. Section 2 presents four sets of streamflow data used in this study and describes the principles of PSR and the identification of its parameters using the correlation integral approach and the FNN approach. This section helps to identify all model inputs except for ARMA. Section 3 describes the SSA and the choice of its parameters. The modeling methods are presented in Section 4. The implementation of the forecast models, including data preparation and selection of model parameters, is discussed in Section 5. Forecast results and some discussions are described in Section 6 and conclusions of this study are presented in Section 7.

2. Reconstruction of Dynamics

2.1 Streamflow Data

Monthly streamflow series of three watersheds and one river, i.e. Xiangjiaba, Manwan, Danjiangkou, and Yangtze River, were analyzed for reconstruction of dynamics. Considering similarity of flow characteristics amongst the former three (see relevant analysis later), only Xiangjiaba and Danjiangkou were selected for forecasting in this study.

The largest watershed, Xiangjiaba, is at the upstream of Yangtze River with average yearly discharge of 4538 m^3/s. Monthly streamflow series were taken from the hydrological station near the Xiangjiaba Dam site located in Sichuan Province. The basin area contributed to the streamflow series is around 45.88×10^4 km^2. The period of the data was from January 1940 to December 1997.

The medium watershed, Manwan, is located in the Lancang River which originates from the Qinghai-Tibet Plateau. Monthly streamflow series were taken from the hydrological station near the Manwan Dam site located in Sichuan Province. The catchment area controlled by the station is 11.45×10^4 km^2, and the average yearly discharge is 1230 m^3/s based on a statistic of 30-year data (January 1974 to December 2003).

The smallest watershed, Danjiangkou, lies at the upstream of Han River with average yearly discharge of 1203 m^3/s. Monthly streamflow data came from the hydrology station at the Danjiangkou Dam site which is located in Hubei Province. The catchment area at the dam site is around 9.5×10^4 km^2. The data range was from January 1930 to December 1981.

The last streamflow series is Yangtze River, the largest river in China. The selected monthly streamflow data were from the hydrology station of Cuntan located in the middle stream of the river. The stream flow series spanned from January 1893 to December 2007.

Four monthly streamflow series are shown in Fig. 1. Monthly streamflow data in Xiangjiaba, Manwan, and Cuntan are characterized by a smooth process whereas monthly streamflow data in Danjiangkou exhibits more complex oscillations. The linear fits (dotted lines in Fig. 1) verify the consistency of the streamflow series. All series exhibit good consistency because the linear fits are closed to horizontal. Since there was no large-scale hydraulic works such as dams built during the data collection period, the streamflow process is fairly pristine in each case.

2.2 Phase Space Reconstruction

The most frequently used reconstruction method for a univariate or multivariate time series is the delay-time method (*Takens, 1981; Farmer and Sidorowich,1987; Sauer et al. ,1991*). A dynamic univariate time series $\left\{x_1, x_2, \cdots, x_N\right\}$, may be reconstructed into a series of delay vectors of the type $\mathbf{Y}_{\mathrm{t}}=\left\{x_t, x_{t+\tau}, x_{t+2\tau}, \cdots, x_{t+(m-1)\tau}\right\}$, $t=1,2,\cdots,N-(m-1)\tau$, where $\mathbf{Y}_{\mathrm{t}} \in \mathrm{R}^m$, τ is the delay time as a multiple of the sampling period and m is the embedding dimension. Under ideal conditions of time series of infinite length, all the reconstructions would be analogous and topologically equivalent to the real system. Owing to the shortness of real time series and the inevitable presence of dynamical noise, optimal reconstruction is involved in the choice of m and τ (*Laio et al., 2003*).

The temporal evolution of the dynamic system is given as a mapping $\mathbf{Y}(\mathrm{t}) \mapsto \mathbf{Y}(\mathrm{t+T})$ (or $\mathbf{Y}_t \mapsto \mathbf{Y}_{\mathrm{t+T}}$) . The functional relationship between the current state $\mathbf{Y}(\mathrm{t})$ at time t and the predicted state $\mathbf{Y}^{\mathrm{F}}(\mathrm{t+T})$ at time $t+T$ can be written as follows:

$$\mathbf{Y}^{\mathrm{F}}(\mathrm{t+T}) = f(\mathbf{Y}(\mathrm{t})) + e_t \qquad (1)$$

where e_t is a typical noise term. In the form of time series, it can be expressed as $[x^F_{t+T}, x^F_{t+T+\tau}, \cdots, x^F_{t+T+(m-2)\tau}, x^F_{t+T+(m-1)\tau}] = f([x_t, x_{t+\tau}, \cdots x_{t+(m-2)\tau}, x_{t+(m-1)\tau}]) + e_t$. Therefore, predicting future trajectory by current trajectory becomes viable once the function $f(\bullet)$ is determined. In practice, the expression is often defined as:

$$x^F_{t+T+(m-1)\tau} = f(\mathbf{Y}(\mathrm{t})) + e_t \qquad (2)$$

where only the last component in $\mathbf{Y}^{\mathrm{F}}(\mathrm{t+T})$ is indicated since normally the prediction of this last component is of concern (*Laio et al., 2003*).

2.3 Determination of Parameters (τ , m)

The correlation exponent method and False Nearest Neighbors (FNN) method were employed to identify the parameter pair (τ , m) in this study.

(1) Correlation exponent

The method identifies the two parameters for the perspective of verifying the existence of chaos. The diagnosis of the existence of chaos can begin if the phase space has been reconstructed. The PSR requires two parameters (τ , m), where m can be identified from the plot of d_2 (the correlation dimension) versus m when d_2 reaches a saturation value D_2 (*Tsonis*, *1992*). In other words, D_2 must be obtained prior to the achievement of m . However, m is subject to τ (often called decorrelation time) and should be first determined. Also, some researchers thought that τ and m should not be determined separately. For example, Liong et al. (*2002*) used Genetic Algorithm (GA) to optimize the triplet (m, τ, k). In the present study, we tend to adopt a widely accepted method to obtain τ as follows.

The τ can be defined when the AutoCorrelation Function (ACF) attains the value of zero or below a small value, or the Average Mutual Information (AMI) reaches the first minimum (*Tsonis, 1992; Kantz and Schreiber, 2004*). The calculations of ACF and AMI are discussed in detail in the works of Fraser and Swinney (*1986*), Tsonis (*1992*), and Abarbanel et al. (*1993*). Fig. 2 displays the ACF and AMI for the four monthly streamflow series. The ACF for each series was first attained zeros at the same lag time of 3. Since the AMI gave the same estimates of τ as the ACF, τ was consistently chosen at the lag time of 3 for each case.

With the chosen τ , the correlation dimension can be computed by the correlation integral according the formula of Grassberger-Procaccia algorithm (*Grassberger and Procaccia,1983*). This original formula was modified by Theiler (*1986*) for the estimation of the correlation integral in a time series which poses serious problems of temporal correlations. Thus, the modified correlation integral $C(r)$ for a *m*-dimension phase space is defined as:

$$C(r) = \frac{2}{N_{\text{pairs}}} \sum_{i=1}^{N} \sum_{j=i+w+1}^{N-i} \mathrm{H}(r - \|\mathbf{Y}_i \text{-} \mathbf{Y}_j\|) \tag{3}$$

where $N_{\text{pairs}} = (N - w + 1)(N - w)$, w is the Theiler window excluding those points which are temporally correlated, $\mathbf{Y}_i$ and N are the same as in Section 2.2, r is the radius of a ball centered on $\mathbf{Y}_i$, H is the Heaviside step function with $\mathrm{H}(u) = 1$ if $u > 0$ and $\mathrm{H}(u) = 0$ if $u \le 0$. The correlation integral only counts the pairs ($\mathbf{Y}_i, \mathbf{Y}_j$) whose distance, in a Euclidean sense, is smaller than r. In the limit of an infinite amount of data ($N \to \infty$) and sufficiently small r, the relation of $C(r) \propto r^{D_2}$ between $C(r)$ and r is expected when m exceeds the correlation dimension of the chaos system. The correlation dimension D_2 and the correlation exponent ν can be defined as:

$$D_2 = \lim_{\substack{r \to 0 \\ N \to \infty}} \nu \text{, where } \nu = \frac{\partial \ln C(r)}{\partial \ln r} \tag{4}$$

Since D_2 is unknown before conducting the computation, the convergence of the correlation dimension D_2 in m must be examined.

The procedure of the computation is first to plot $\ln C(r)$ versus $\ln r$ with a given m. Then, the potential scaling region is determined wherever the slope (i.e. the correlation exponent ν) of the curve for the given m is approximately constant. The constant slope can be estimated by a straight line fitting of the scaling region. In general, the best way to define the scaling region is to produce another figure which demonstrates the slope of the $\ln C(r)$ as a function of $\ln r$. If a scaling region exists, a plateau should be shown in the figure. This plateau provides an estimate for d_2, a correlation dimension of the possible attractor for the present m.

If d_2 converges to a finite value D_2 (i.e. saturation value) after repeating the above procedure for successively higher m, a true attractor of dimension D_2 is formed and the system may be considered as chaos. Meanwhile, m can be identified as the value that corresponds to the first occurrence of the saturation value D_2 in the plot of d_2 versus m.

The graphs of the correlation exponent ν versus $\ln r$ for the four streamflow series are shown in Figs. 3 respectively. Fig. 3 demonstrates that the scaling region cannot be identified for any m ($m < 20$) in Danjiangkou catchment whereas the scaling region can be determined in other three cases. It can be noticed that the scaling region becomes ambiguous at $m = 8$ for Manwan and Xiangjiaba whereas a narrow scaling region can still be defined even at $m = 20$ for Cuntan.

However, an accurate estimation of d_2 requires a minimum number of points. Some researches claim that the size should be 10^A (*Procaccia, 1988*) or $10^{2+0.4m}$ (*Tsonis, 1992*), where A is the greatest integer smaller than d_2 and m ($m < 20$) is the embedding dimension used for estimating d_2 with an error less than 5%. Other research found that smaller data size is needed. For instance, the minimum data points for reliable d_2 is $10^{\mathrm{d}_2/2}$ (*Ruelle, 1990; Essex and Nerenberg, 1991*), or $\sqrt{27.5}^{d_2}$ (*Hong and Hong, 1994*) and empirical results of dimension calculations are not substantially altered by going from 3000 or 6000 points to subsets of 500 points (*Abraham et al., 1986*).

Based on the results of Fig. 3, the relationship between correlation dimension d_2 and embedding dimension m is depicted in Fig. 4. The saturation values D_2 for three streamflow series (Xiangjiaba, Cuntan and Manwan) are at interval of (1.5, 2). Generally, a sufficient condition for the smallest m is that m is an integer larger than $2D_2$. The associated m is therefore set the value of 4 for the three series. With the potential values on D_2 or m, some criteria such as 10^A can be satisfied whereas other criteria such as $10^{2+0.4m}$ cannot be satisfied. The latter criteria means that few hydrologic records can be assessed for $m > 5$ attractors since as many as 10,000 points require 27 years of daily records or around 900 years of monthly records. Thus, the three monthly series of Xiangjiaba, Cuntan and Manwan may be treated as chaotic with suggested variable $m = 4$. Danjiangkou series cannot be identified as non-chaotic or random process because its data size is as small as 624. Furthermore, the phase portraits of four streamflow series are portrayed in Fig. 5 where (τ, m) is (3,3). Obviously, the state spaces in the 3-dimensional maps are clearly unfolded for Xiangjiaba, Cuntan, and Manwan whereas no clear trajectory is revealed for Danjiangkou.

(2) FNN

The correlation integral method appears to be data intensive and certainly subjective. For simplicity, the FNN method is commonly employed for the PSR of a hydrologic streamflow series (*Wang et al., 2006b*). The FNN algorithm was originally developed for determining the number of time-delay coordinates needed to recreate autonomous dynamics directly from properties of the data itself (*Kennel et al., 1992; Abarbanel et al., 1993*). The following discussion outlines the basic concepts of the FNN algorithm. Suppose the point $\mathbf{Y}_\mathrm{i} = \left\{x_i, x_{i+\tau}, x_{i+2\tau}, \cdots, x_{i+(m-1)\tau}\right\}$ has a neighbor $\mathbf{Y}_\mathrm{j} = \left\{x_j, x_{j+\tau}, x_{j+2\tau}, \cdots, x_{j+(m-1)\tau}\right\}$, the criterion that $\mathbf{Y}_\mathrm{j}$ is viewed as a false neighbor of $\mathbf{Y}_\mathrm{i}$ is:

$$\frac{\left|x_{i+m\tau} - x_{j+m\tau}\right|}{\left\|\mathbf{Y}_\mathrm{i} \text{-} \mathbf{Y}_\mathrm{j}\right\|} > \mathrm{R}_\mathrm{tol} \qquad (5)$$

where $\| \ \|$ stands for the distance in a Euclidean sense, R_tol is some threshold with the common range of 10 to 30 (set as 15 in this study). For all points i in the vector state space, Eq. (5) is performed and then the percentage of points which have FNNs is calculated. The algorithm is repeated for increasing m until the percentage of FNNs drops to zero, or some acceptable small number such as 1%, where m is the target m .

Fig. 6 demonstrates the Percentage of FNNs (FNNP) for the four streamflow series with $\mathrm{R}_{\mathrm{tol}} = 15$ and $\tau = 3$. The identified m is 4 for Manwan and 5 for other three cases. The FNN technique seems to be unable to distinguish random process from deterministic system because a similar m was found when the FNN was applied to a random series with the same data size as Cuntan or Danjiakou. Thus, the phase space reconstructed by the FNN does not mean the unfolding of a potential attractor in a dynamic system.

In summary, the parameter pair (τ, m) in PSR is (3, 4) for Xiangjiaba, Cuntan, and Manwan and is (3, 5) for Danjiangkou when KNN is used for the streamflow prediction. The determination of m also means the achievement of the model inputs in three soft computing models of ANN, CDANN, and CDSVR. For the purpose of forecasting, the value of the lagged time τ is set to one in the three soft computing models because we tolerate having some information redundancy in preference to losing any useful information. For instance, the previous four month data is used as inputs of ANN if the one-month-ahead forecast is performed for Xiangjiaba, and the previous five month data is the inputs of ANN for Danjiangkou.

3. Singular Spectrum Analysis

3.1 Theory of SSA

The SSA employed in the study can be referred to in Vautard et al. (*1992*) and Elsner and Tsonis *(1997)*. For the sake of simplicity, a univariate time series is considered for explanation of the SSA. In general, four steps are involved in the implementation of SSA. The first step is to construct the 'trajectory matrix'. The 'trajectory matrix' results from the method of delays. In the method of delays, the coordinates of the phase space will approximate the dynamic of the system by using lagged copies of the time series. Therefore, the 'trajectory matrix' can reflect the evolution of the time series with a careful choice of (τ , m) window. In the context of SSA, the m is usually called the window length (or singular number). For time series $\{x_1, x_2, \cdots, x_N\}$, the 'trajectory matrix' is denoted by

$$\mathbf{X} = \frac{1}{\sqrt{N}} \begin{pmatrix} x_1 & x_{1+\tau} & x_{1+2\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & x_{2+2\tau} & \cdots & x_{2+(m-1)\tau} \\ x_3 & x_{3+\tau} & x_{3+2\tau} & \cdots & x_{3+(m-1)\tau} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-(m-1)\tau} & x_{N-(m-2)\tau} & x_{N-(m-3)\tau} & \cdots & x_N \end{pmatrix} \tag{6}$$

where m is the embedding dimension, τ is the lagged (or delay) time. The matrix dimension is $n \times m$ where $n = N-(m-1)\tau$. The next step is the singular value decomposition (SVD) of the trajectory matrix $\mathbf{X}$. Let $\mathbf{S} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$ (called the lagged-covariance matrix). With SVD, $\mathbf{X}$ can be written as $\mathbf{X} = \mathbf{DLE}^{\mathrm{T}}$ where $\mathbf{D}$ and $\mathbf{E}$ are left and right singular vectors of $\mathbf{X}$, and $\mathbf{L}$ is a diagonal matrix of singular values. $\mathbf{E}$ consists of orthonormal columns, and is also called the 'empirical orthonormal functions' (EOFs). Substituting $\mathbf{X}$ into the definition of $\mathbf{S}$ yields the formula of $\mathbf{S = ELD^{\mathrm{T}}DLE^{\mathrm{T}}} = \mathbf{EL^2E^{\mathrm{T}}}$. Further $\mathbf{S}$=$\mathbf{E} \wedge \mathbf{E}^{\mathrm{T}}$ since $\mathbf{L}^2 = \wedge$ where $\wedge$ is a diagonal matrix consisting of ordered values $0 \le \lambda_1 \le \lambda_2 \le \cdots \lambda_m$. Therefore, the right singular vectors of $\mathbf{X}$ are the eigenvectors of $\mathbf{S}$ (*Elsner and Tsonis, 1997*). In other words, the singular vectors $\mathbf{E}$ and singular values of $\mathbf{X}$ can be respectively attained by calculating the eigenvectors and the square roots of the eigenvalues of $\mathbf{S}$.

The first two steps make up the decomposition stage of SSA, and the next two steps belong to the recovering stage. The third step is to calculate the principal components (a_i^k's) by projecting the original time record onto the eigenvectors as follows:

$$a_i^k = \sum_{j=1}^{m} x_{i+(j-1)\tau} e_j^k \text{ , for } \quad i = 1, 2 \cdots, N-(m-1)\tau \tag{7}$$

where e_j^k represents the jth component of the kth eigenvector. As known, each principal component is a filtered process of the original series with length $N-(m-1)\tau$, not length N as desired, which poses a problem in real-time prediction.

The last step is to generate reconstruction components (RCs) whose lengths are the same as the original series. The generation of each RC depends on a convolution of one principal component with the corresponding singular vector, given by

$$x^k{}_{i+(j-1)\tau} = a_i^k e_j^k \text{ for } k = 1, 2 \cdots, m \tag{8}$$

where $i = 1, 2 \cdots, N-(m-1)\tau$, and $j = 1, 2 \cdots, m$, or given by Vautard et al. (*1992*). Therefore, The m RCs can be achieved and their sum is a complete recover of the original series if all m principal components and their associated eigenvectors are employed in the process of signal reconstruction. To reflect significant oscillatory modes, the original record can be filtered by choosing p ($< m$) RCs from all m RCs.

3.2 Determination of Parameters (τ , m)

The choice of the parameter pair (τ , m) in SSA are not as strict as in PSR. The τ and m from SSA and PSR are generally different although some authors considered them as same in application (e.g., *Sivapragasam et al., 2001*). The choice of m represents a compromise between information content and statistical confidence. As a basic requirement, the chosen m should be able to resolve obvious different oscillations hidden in the original signal. In other words, some leading eigenvalues should be identified. Figs. 7 and 8 show the sensitivities of the eigenvalue decomposition to the window length m and the lag time τ respectively. Results from Fig. 7 show that 5 leading eigenvalues stand out for Xiangjiaba, Cuntan, and Manwan whilst 7 leading eigenvalues can be identified for Danjiangkou when $m \geq 30$. Each leading eigenvalue corresponds to a distinctive oscillation. Results from Fig. 8 suggest that the curve of eigenvalues of each series is insensitive to the lagged time τ when $m = 30$. The τ was consistently chosen as 1. The m of 30 may be too large for the analysis below. For simplicity, the final parameter pair (τ, m) in SSA were (1, 5) for Xiangjiaba, Cuntan, and Manwan and (1, 7) for Danjiangkou.

Taking Danjiangkou as an example, Fig. 9 presents 7 RCs and the original series excluding the validation data. The RC1 represents the lowest frequency oscillation and the RC7 reflects the highest frequency oscillation. The high-frequency components stand for noise signals to some extent. Potentially, they may be filtered out so as to clean the original series. Fig. 10 depicts AMI and CCF between RCs and the original streamflow series. The last plot presents the average of the AMI and CCF from the previous plots. The average value indicates an overall correlation either being positive or negative. As shown in Fig. 10, the value of CCF from each RC is significantly different with the increase of the lag time. It means that the contribution of each RC to the output of models is changeable at different prediction levels. Therefore, the number p ($< m$) of RCs at different prediction levels will be chosen by trial and error according to the minimum performance of model (Root Mean Square Error was taken in this study).

4. Description of Models

Fig. 11 depicts the framework of implementation of all five data-driven models in this study. They are ARMA, KNN, ANN, CDANN, and CDSVR from the top to the bottom. The configuration of each model except for ARMA was introduced as follows. In view of the similarity of the flow characteristics among Xiangjiaba, Cuntan, and Manwan, two streamflow series, Xiangjiaba and Danjiangkou, were therefore chosen for forecasting in the rest of this paper.

4.1 KNN

The KNN was applied to estimate the function of $f(\bullet)$ in Eq. (2) in the study. The KNN algorithm depends on observations that are in some finite neighborhood of the point of estimate. The basic idea behind the KNN is to break up domain into local neighborhoods and fit parameters in each neighborhood separately. To predict $x^{F}_{t+T+(m-1)\tau}$ based on $\mathbf{Y}(\mathrm{t})$ and the past ones, k nearest neighbors of $\mathbf{Y}(\mathrm{t})$ should be found. Jayawardena and Lai (*1994*) summarized five local prediction approaches according to the k being equal to 1 or larger than 1: zeroth order approximation when $k=1$; for $k>1$, the other four are respectively average k neighbours, weighted average of k neighbours, exponential weighting k neighbours, and linear approximation via fitting k pairs $(\mathbf{Y}(\mathrm{t'_i}), \mathbf{Y}(\mathrm{t'_i}+\tau))$. In the current study, the weighted averaging k neighbors approach is adopted and the weight function is proportional to the inverse of square Euclidean distance between $\mathbf{Y}(\mathrm{t})$ and $\mathbf{Y}(\mathrm{t'})$, which can be referred to Wu et al. (*2008*).

4.2 ANN

The single ANN used in this study was a static feed-forward multilayer perceptron (MLPs). The static ANN is able to capture the dynamics of a system in the network model by using delayed time inputs. The architecture design of the ANN consists of the number of hidden layers and the number of neurons in input layer, hidden layers and output layer. ANNs with one hidden layer are commonly used in hydrologic modeling (*Dawson and Wilby, 2001; de Vos and Rientjes, 2005*) since these networks are considered to provide enough complexity to accurately simulate the dynamic and nonlinear-properties of the hydrologic process. Therefore, a three-layer static ANN was finally chosen for the present study, which comprises the input layer with m neurons, the hidden layer with h nodes, and the output layer with one node. As mentioned previously, m is 4 for Xiangjiaba and 5 for Danjiankou. The hyperbolic tangent was chosen as the activation function for the neurons of the hidden layer, which usually requires rescaling both the input and outputting data to [-1, 1]. Based on the form of Eq. (2), the prediction model is mathematically expressed as

$$x^F_{t+T+(m-1)\tau} = f(\mathbf{Y}(\mathrm{t}), w, \theta, m, h) = \theta_0 + \sum_{j=1}^{h} w^{out}_{j} \tanh(\sum_{i=1}^{m} w_{ji} x_{t+(i-1)\tau} + \theta_j) \qquad (9)$$

where $x_{t+(i-1)\tau}$ are the m components of the state $\mathbf{Y}(\mathrm{t})$; w_{ji} are the weights defining the link between the ith node of the input layer and the jth of the hidden layer; θ_j are biases associated to the jth node of the hidden layer; w^{out}_{j} are the weights associated to the connection between the jth node of the hidden layer and the node of the output layer; and θ_0 is the bias at the output node.

As mentioned in Section 2.3, m is 4 for Xiangjiaba and 5 for Danjiankou for a unique predicted output. The ensuing task is to optimize the size h of the hidden layer with the determined input and output neurons. In terms of CDANN and CDSVR, they employed the same input and output for each single model of them as the ANN.

4.3 Distributed Models

(1) FCM (*Bezdek, 1981; Wang et al., 2006b*)

The Fuzzy C-Means (FCM) method partitions a set of N vector $\mathbf{Y_j}$, j=1,...,N , into c fuzzy clusters, and each data point belongs to a cluster to a degree specified by a membership grade u_{ij}, between 0 and 1. We define a matrix $\mathbf{U}$ comprising the elements u_{ij}, and assume that the summation of degrees of belonging for a data point is equal to 1, i.e., $\sum_{i=1}^{c} u_{ij} = 1$, $\forall j = 1, \cdots, \mathrm{N}$. The goal of the FCM algorithm is to find c cluster centers so that the cost function of dissimilarity measure is minimized. The cost function can be defined by

$$J(\mathbf{U}, v_1, \cdots v_c) = \sum_{\mathrm{i}=1}^{\mathrm{c}} J_{\mathrm{i}} = \sum_{\mathrm{i}=1}^{\mathrm{c}} \sum_{\mathrm{j}=1}^{\mathrm{N}} u_{ij}^{l} d_{ij}^{2} \qquad (10)$$

where v_i is the cluster center of the fuzzy subset i ; $d_{ij} = \left\| v_i - \mathbf{Y}_j \right\|$ is the Euclidean distance between the ith cluster center and jth data point; and $l \geq 1$ is a weighting exponent, taken as 2 here so as to match the square Euclidean distance. The necessary conditions for Eq. (10) to reach its minimum are:

$$v_i = \sum_{\mathrm{j}=1}^{\mathrm{N}} u_{ij}^{l} \mathbf{Y}_j \Big/ \sum_{\mathrm{j}=1}^{\mathrm{N}} u_{ij}^{l} \qquad (11)$$

$$u_{ij} = \left[\sum_{\mathrm{k}=1}^{\mathrm{c}} \left(\frac{d_{\mathrm{ij}}}{d_{\mathrm{kj}}} \right)^{2/(l-1)} \right]^{-1} \qquad (12)$$

The FCM algorithm is an iterative procedure that satisfies Eqs. (11) and (12) to minimize Eq. (10). Implementation of the algorithm can be referred to Bezdek (*1981*) and Wang et al. (*2006b*) for details.

The number c of clusters was taken to be 3 in this study. The three subsets were expected to represent three different magnitudes of flows, i.e., low flow, medium flow, and high flow.

(2) SVR

A nonlinear SVR was used in this study. A brief introduction of SVR was presented here. The details of SVR can be referred to Kecman (*2001*), Yu et al. (*2006*) and Wu et al. (*2008*). According to equation (2), the underlying function $f(\bullet)$ in the context of a nonlinear SVR is given by

$$x^F_{t+T+(m-1)\tau} = f(\mathbf{Y}(\mathrm{t}), \omega) = \omega \cdot \phi(\mathbf{Y}(t)) + b \tag{13}$$

where the input data $\mathbf{Y}(\mathrm{t})$ in the input space is mapped to a high dimensional feature space via a nonlinear mapping function $\phi(\mathbf{Y}(\mathrm{t}))$. The objective of the SVR is to find optimal ω , b and some parameters in kernel function $\phi(\mathbf{Y}(\mathrm{t}))$ so as to construct an approximation function of the $f(\bullet)$.

When introducing Vapnik's ε-insensitivity error or loss function, the loss function $L_\varepsilon(\mathrm{y}, f(\mathbf{Y}(\mathrm{t}), \omega))$ on the underlying function can be defined as

$$L_\varepsilon(\mathrm{y}, f(\mathbf{Y}(\mathrm{t}), \omega)) = \left|\mathrm{y} - f(\mathbf{Y}(\mathrm{t}), \omega)\right|_\varepsilon = \begin{cases} 0 & if \left|\mathrm{y} - (\omega \cdot \phi(\mathbf{Y}(\mathrm{t})) + b)\right| \le \varepsilon \\ \left|\mathrm{y} - (\omega \cdot \phi(\mathbf{Y}(\mathrm{t})) + b)\right| - \varepsilon & otherwise \end{cases} \tag{14}$$

where y represents observed value. Similar to linear SVR (*Kecman, 2001; Yu et al., 2006*), the nonlinear SVR problem can be expressed as the following optimization problem:

$$\begin{aligned} &\textit{minimize} \quad R_{W,\xi_i,\xi_i^*} = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ &\textit{subject to} \quad \begin{cases} \mathrm{y_i} - f(\phi(\mathbf{Y}_\mathrm{i}),\omega) - b \le \varepsilon + \xi_i \\ f(\phi(\mathbf{Y}_i),\omega) + b - \mathrm{y_i} \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \end{aligned} \tag{15}$$

where $\mathbf{Y}_\mathrm{i}$ represents $\mathbf{Y}(\mathrm{i})$ for simplicity, the term of $\frac{1}{2}\|\omega\|^2$ reflects generalization, and the term of $C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$ stands for empirical risk. The objective in Eq. (15) is to minimize them simultaneously, which implements SVR to avoid underfitting and overfitting the training data. ξ_i and ξ_i^* are slack variables for measurements "above" and "below" an ε tube. Both slack variables are positive values. C is a positive constant that determines the degree of penalized loss when a training error occurs.

By introducing a dual set of Lagrange multipliers, α_i and α_i^*, the minimization problem can be solved in a dual space. The objective function in dual form can be represented as (*Gunn, 1998*)

$$\begin{aligned} &\textit{maximize} \quad L_d\left(\alpha,\alpha^*\right) = -\varepsilon\sum_{i=1}^{N}\left(\alpha_i^* + \alpha_i\right) + \sum_{i=1}^{N}\left(\alpha_i^* - \alpha_i\right)y_i - \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\left(\phi(\mathbf{Y}_\mathrm{i})\cdot\phi(\mathbf{Y}_\mathrm{j})\right) \\ &\textit{subject to} \quad \begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ 0 \le \alpha_i^* \le C, \qquad i = 1,\cdots,N \\ 0 \le \alpha_i \le C, \qquad i = 1,\cdots,N \end{cases} \end{aligned} \tag{16}$$

By using a "kernel" function $K(\mathbf{Y}_\mathrm{i},\mathbf{Y}_j) = \left(\phi(\mathbf{Y}_i)\cdot\phi(\mathbf{Y}_\mathrm{j})\right)$ to yield inner products in feature space, the computation in input space can be performed. In the present study, Gaussian radial basis function (RBF) was adopted in the form of $K(\mathbf{Y}_\mathrm{i},\mathbf{Y}_j) = \exp(-\left\|\mathbf{Y}_i - \mathbf{Y}_j\right\|^2 \big/ 2\sigma^2)$.

Once parameters α_i, α_i^*, and b_0 are obtained, the final approximation function of the $f(\bullet)$ becomes

$$f(\mathbf{Y}_i) = \sum_{i=1}^{N} (\alpha_k - \alpha_k^*) K(\mathbf{Y}_k \cdot \mathbf{Y}_\mathrm{i}) + b_0 \, , k = 1, \cdots, s \qquad (17)$$

where $\mathbf{Y}_\mathrm{k}$ stands for the support vector, α_k and α_k^* are parameters associated with support vector $\mathbf{Y}_\mathrm{k}$, N and s represent the number of training samples and support vectors respectively. Three parameters (C, ε, σ), however, have to be optimized first before Eq. (17) is used to perform the forecasting.

(3) CDANN and CDSVR

As depicted in Fig. 11, CDANN (or CDSVR) consists of three single ANNs (or SVR). A basic idea behind the type of model is that the training set is first split into three subsets by the FCM, and then each subset is fitted by a single ANN (or SVR). The three subsets are associated with subset1, subset2 and subset3 in Fig. 11. Generally, the final output of the distributed model is a weighted sum of the outputs of all single models when making a forecast from a new input vector, and the weights are determined according to the distance between the new input vector and each cluster center (*Cheng et al., 2006; Wu et al.,2008*). As mentioned in Introduction, the DANN (or DSVR) dependent upon weighted output was called SDANN (or SDSVR). They tend to generate poor forecasts because ANN (or SVR) is unable to extrapolate beyond the range of the data used for training. In the CDANN (or CDSVR), only one single ANN (or SVR) model, whose clustering center is the nearest to the new input, is triggered for the forecasting. As a consequence, the final output of the distributed models is directly from the output of the triggered ANN (or SVR). Therefore, the validation set is necessarily split into three parts in advance before the forecasting or simulation by CDANN (or CDSVR).

5. Application of Models

5.1 Data preparation

Streamflow series data were divided into three parts: model training, cross-validation and validation. The last ten years' streamflow data, called validation set, were set aside for validation. This validation set would not be used until all model development and training was finished completely. Of the remaining data, the first two-thirds called training set was for model training, and the other one-third called cross-validation set was for the purpose of confirming and validating the initial analysis. Models based on soft computing method have difficulties to extrapolate beyond the range of the data used for training. As a consequence, it is imperative that the training and validation sets are representative of the same population. Statistical properties (mean, deviation, range) from them are compared in order to measure the representative. According to the statistical properties of these data sets, the division of data is satisfied since the statistical parameters of the training sets for all the streamflow series are closed to the cross-validation sets and the testing sets.

With suitable data preparation beforehand, it is possible to improve the performance of models although these soft computing models (ANN, CDANN, and CDSVR) are very powerful to handle nonlinear, noisy, and non-stationary data. Data-preprocessing can ensure that all variables receive equal attention during the training process (*Dawson and Wilby, 2001*). Also, data-preprocessing may be a need to improve the training efficiency of ANN (or SVR) (*Sudheer et al., 2003*). Data-preprocessing techniques include signal filtering, data-transformation, rescaling or standardization. In this study, the signal filtering and data-transformation were respectively the SSA and Moving Average (MA) over input signals. The rescaling operation was applied to models of ANN, CDANN, and CDSVR, which rescaled the input and output data to [-1, 1].

5.2 Evaluation of model performances

The measures of evaluating model performance that are used in the present paper comprise Root Mean Square Error (RMSE), the Nash-Sutcliffe Coefficient of Efficiency (CE) (*Nash and Sutcliffe, 1970*), and the Persistence Index (PI) (*Kitanidis And Bras, 1980*). They are respectively formulated as: $\mathrm{RMSE}=\sqrt{\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{T_i}-\hat{\mathrm{T}}_\mathrm{i})^2}$), $\mathrm{CE}=1-\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{T_i}-\hat{\mathrm{T}}_\mathrm{i})^2 \Big/ \sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{T_i}-\bar{\mathrm{T}})^2$ and $\mathrm{PI}=1-\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{T_i}-\hat{\mathrm{T}}_\mathrm{i})^2 \Big/ \sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{T_i}-\mathrm{T_{i\text{-}L}})^2$. In these equations, N =number of observations, $\hat{\mathrm{T}}_\mathrm{i}$ =forecasted streamflow, $\mathrm{T_i}$ =observed streamflow, $\bar{\mathrm{T}}$ =average observed streamflow, and $\mathrm{T_{i\text{-}L}}$ is the streamflow estimate from a so-call persistence model (or called naïve model) that basically takes the last streamflow observation (at time i minus the lead time L) as a prediction. CE and PI values of 1 stand for perfect fits.

5.3 Optimization scheme of model parameters

The ARMA method was used as one of the benchmark models in the present study. Parameters (*P*, *Q*) in ARMA were estimated by trial and error with each of them varied over the range of 0 to 12, excluding the unfeasible case where both of them are simultaneously equal to 0, in view of the predominant periodicity of 12 months (Fig. 2). The best pair of (*P*, *Q*) is associated with the minimal value of Akaike Information Criterion (AIC).

The number of k in KNN in this study was chosen as $k=m+1$ which was suggested by Sugihara and Mary (*1990*). As a result, a fixed $k=5$ and $k=6$ were respectively adopted in Xiangjiaba (recall that $m=4$) and Danjiangkou (recall that $m=5$).

The optimal size h of the hidden layer was found by systematically increasing the number of hidden neurons from 1 to 10 until the network performance on the cross-validation set was no longer improved significantly. The optimization procedure can be begun if ANN is equipped with necessary training algorithm in advance. In our preliminary investigations (Wu et al., Data-driven models for monthly streamflow time series prediction, submitted to Journal of Hydrological Process, 2009; hereinafter referred to as *Wu et al., submitted manuscript, 2009*), we tested performance of three training algorithms of Levenberg-Marquardt (LM), Particle Swarm Optimization (PSO), and Levenberg Marquart combined with Genetic algorithm (LM-GA). The results showed that the LM-GA performed best. Thus, the LM-GA algorithm was used for training in this study. Finally, a 4-8-1 configuration of the ANN model with the LM-GA algorithm was designed for one-month-ahead forecast of Xiangjiaba. Similarly, a 5-5-1 configuration of the ANN model was found for one-month-ahead forecast of Danjiangkou. For other prediction levels, the configurations of ANN and CDANN were also found by the same trial and error method.

The CDSVR had the same inputs and output of model as ANN and CDANN. To conduct forecasting, three parameters (C, ε, σ) in CDSVR is necessarily determined when Gaussian RBF was adopted as the kernel function. Some techniques to determine these parameters have been recommended in literature. For instance, Cherkassky and Ma (*2004*), from the perspective of analytic 'rule-of-thumb' selection, suggested selection of parameter C based on the formula: $\frac{C}{N} = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|)$, where $\bar{y}$ and σ_y are the mean and the standard deviation of the training data y, and N is the data size of training data. ε is suggested as $\varepsilon = 3\sigma_a\sqrt{\ln N / N}$, where $\sigma_a{}^2$ is the variance of additive noise δ. Global evolutionary techniques are widely reported for optimization of these parameters. Hsu et al. (*2003*) recommend a “grid-search” on C and parameter σ in RBF using cross-validation. Yu et al. (*2004*) and Lin et al. (*2006*) employed Shuffled Complex Evolution (SCE-UA) to search these parameters. A two-step grid search method was applied to find the optimal parameter triplet (C,ε,σ) by Yu et al. (*2006*) and Wu et al. (*2008*). In the present study, the two-step GA was found the best in terms of effectiveness and efficiency of optimization compared to two-step PSO and two-step SCEUA. Thus, the two-step GA method was employed in this study.

There are other two important parameters, the number p of chosen reconstructed components and the memory length of moving average. They were chosen by trial and error and the final values of them were associated with the minimum RMSE.

6. Results and Discussion

5.1Main results

Table 1 shows the results of one-month-ahead prediction of all five models without data preprocessing of MA and SSA using the monthly data from Xiangjiaba and Danjiangkoufor respectively. These models made better accuracy for Xiangjiaba data than Danjiangkou data. ARMA gave the best performance for Xiangjiaba and the second best performance for Danjiangkou as compared to the other models, which is inconsistent with the conclusion in literature that ANN mostly outperforms ARMA (*Jain et al. 1999; Abrahart and See, 2002; Castellano-Me´ndeza et al. 2004*). This is most likely due to the ANN (CDANN or CDSVR) model not being optimal considering that the model input was identified by reconstruction of dynamics.

Figs. 12 and 13 present the scatter plots and the plots of the observed and forecasted time series of one-month-ahead forecast for Xiangjiaba and Danjiangkou using the ANN and KNN models. The biggest drawback is that each model mismatched quite a number of peak flows and the ANN model always underestimated the peak flows of both Xiangjiaba and Danjiangkou. Fig. 13 shows that the ANN model’s timing of the peaks is markedly lagged. This prediction lag effect is the result of using previously observed values as ANN inputs. The ANN gives the most weight to the latest discharge input for calculating the forecast because the forecast depends on the autocorrelation of the monthly time series. Therefore, the MA over inputs was suggested to overcome the lag effect by de Vos and Rientjes (*2005*). Compared with Fig. 12, Fig. 13 also demonstrates that it is difficult for any powerful model to fit a signal contaminated by complex noises. However, it is possible to improve the model performance by using the SSA technique to clean the original signal. This was explored in more detail in Fig. 14 by using KNN model to conduct four different lead forecasts. For Xiangjiaba data, the SSA can significantly improve RMSE of the KNN model at one- and three-month forecast horizons. For Danjiangkou data, the RMSE of the KNN model greatly decreases at all forecast horizons except for twelve-month horizon.

Fig. 15 depicts the results of one-month-ahead forecast of the ANN model with the help of MA and SSA for Xiangjiaba and Danjiangkou. Compared with the scatter plots from ANN model in Figs. 12 and 13, the current scatter plots with low spread, and the low RMSE and high CE and PI indicate better model performances. The peak flows still cannot be fitted perfectly. Also, there is an overall underestimate of the discharge of Danjiangkou by the ANN model because the scatter plot is wholly skew to the axis of observed values, which implies the identified ANN model is not optimal.

The results of one-month-ahead forecast of CDSVR with the aid of MA and SSA are presented in Fig. 16. The scatter plot with perfect match of the diagonal indicates excellent model performance. The plot of observed and forecasted discharges shows that the peak flows are also fitted perfectly. In particular, the whole forecasted error generated by ANN model in Fig. 15 for Danjiangkou is also overcome by the CDSVR model. A further discussion can be found in the later analysis of errors.

Figs. 17 and 18 show the validation data and the comparison of absolute prediction errors of various models for Xiangjiaba and Danjiangkou. Figs. 17a and 18a show that the extreme forecast errors of ARMA model occurred at the last points, which indicates the ARMA model with the current technique of SSA is not viable for a real-time prediction. Figs. 17b and 18b compare three soft computing models of ANN, CDANN and CDSRV. Error curves of ANN and CDANN have very similar trends with obvious underestimates at most points. The error curve of CDANN displays that the peak flows were mostly well simulated, and the errors tend to be random. Fig. 19 presents one-step-ahead absolute forecasting error distributions of KNN, ANN, CDANN, and CDSVR using Danjiangkou data. The histograms of errors for KNN and CDSVR are characterized by a quasi-normal distribution with mean near zero, which indicates that errors for them basically pass the test for “whiteness”. However, the histogram of errors for ANN is evidently skewed to the left which means that the ANN model was not optimal. The histogram of errors for CDANN is also skewed although there is a bit improvement compared to the ANN model. The issue often occurs in the application of ANN. In practice, it is not easy to obtain the optimal configuration of ANN although ANN is able to map any complex relationship.

Table 2 shows the one-month-ahead forecast performance in terms of RMSE, CE, and PI, and relevant parameters including model parameters and data-preprocessing parameters of all five models. The data-preprocessing parameters were decided by trial and error. Compared to the results in Table 1, data-preprocessing techniques of MA and/or SSA can considerably improve the performance of all models. Prediction was also extended to longer prediction horizons, namely $T=3$ months, 6 months, and 12 months. The static prediction method was employed for multiple step prediction, and hence relevant parameters of each model at different forecast horizons have to be optimized again. The results of three-, six-, and twelve-month-ahead forecasts are given in Tables 3-5. The CDSVR model outperforms the other four models at all prediction horizons for Xiangjiaba, and the CDSVR model performs the best among all models at all prediction horizons expect for at six-month-ahead forecast horizon for Danjiangkou. At six-month-ahead forecast horizon, the KNN method performs the best. The data-preprocessing parameters from Tables 2-5 show that MA cannot affect the model performance of ARMA and KNN whereas SSA imposes a great impact on the model performance of KNN. From the perspective of catchment data of the current study, MA is suitable for each catchment data whilst SSA is more effective for Danjiangkou data.

5.2Discussion

The following part concerns some discussions about forecasting models and the effects of data-preprocessing techniques.

(1) About parameter k in KNN

The parameter of k in KNN model poses a great impact on the performance of KNN. The choice of k should ensure the reliability of the solution although a preliminary value of k is based on $k=m+1$. The check of robustness of k in terms of RMSE of validation data was presented in Fig. 20, where k was in the interval of [2, 40]. Adopting the value of k as $m+1$ was reasonable because its RMSE was only 2.3% larger than the minimum RMSE (corresponding to $k=4$) for Xiangjiaba, and 2.5% larger than the minimum RMSE (corresponding to $k=20$) for Danjiangkou.

(2) About models of ANN, CDANN, and CDSVR

The performance of ANN (CDANN or CDSVR) is considerably affected by the identified configuration. Recall that the three soft computing models had the same inputs and output of model, which was derived from the reconstruction of dynamics. The purpose is to fairly compare these models. In terms of the one-month-ahead forecast for Danjiangkou, the configuration of ANN with 12-5-1 achieved better performance where the number of input neurons was optimized by trial and error. Obviously, the model inputs from the reconstruction of dynamics are not the optimal. However, the basic conclusion in this study remains unchanged because the three soft computing models are compared based on the same model inputs.

The distributed models improved the performance of ANN by using local fitting instead of global fitting but at the expense of the increase of the training time. The CDANN is actually three single ANNs. Hence, it has the same disadvantages as the ANN that the forecast results tend to be unstable. The CDSVR, however, exhibited more stable forecast results and better generalization but need longer training time. In view of the fact that training time will exponentially increase with the number of training samples (*Wu et al., 2008*), the SVR model is suitable for samples less than hundreds. Conversely, the ANN model prefers more samples to fewer samples so as to improve the generalization of model. Therefore, ANN and SVR are not always alternative, and are sometimes complementary.

Tables 2-5 show that the number of clustering centers for CDSVR was 3 for one-month-ahead forecast and 3 or 6 for longer forecast horizon. It was found that the performance of CDSVR can be improved at some forecast horizons if the number of clustering centers increases. In the meantime, the increase of clustering centers can remarkably promote the training efficiency of the CDSVR model.

(3) About the investigation of effects of MA and SSA

To investigate the effect of MA on the ANN model, Fig. 21 compares the Cross-Correlation Functions (CCFs) between inputs and output from ANN with/without MA for one-month-ahead forecast. For Xiangjiaba whose original data has high autocorrelation, the MA decreases the correlation relationship between the latest two input components and the output of model whereas the MA increases the correlation relationship between other three input components and the output of model. For Danjiangkou whose raw data has low autocorrelation, the MA increases the correlation relationship between each of seven input components and the output of model.

Fig. 22 investigates the combined effect of MA and SSA on the ANN model by computing the CCFs between inputs and output of model for a six-month-ahead forecast. Fig. 22 shows that there is a common characteristic for Xiangjiaba and Danjiangkou that the combined effect of MA and SSA is to mainly improve the correlation relationship between the more previous input components and the output of model. In fact, the MA and SSA decrease the contribution of the latest component to the output of model, which potentially overcomes the timing error of forecast.

7. Conclusions

In this paper, five data-driven models, ARMA, KNN, ANN, CDANN, and CDSVR, were adopted to conduct multiple-step monthly discharge forecasts by two catchment data, Xiangjiaba and Danjiangkou. In order to improve the model performance, two data-preprocessing techniques of the MA and SSA were equipped to each model. The results of the one-month-ahead forecasts showed that the MA and/or SSA can considerably improve the performance of each model. The results of the multiple-step forecasts demonstrated that CDSVR stood out from other models except for at the six-month-ahead forecast horizon on Danjiangkou data. In terms of the CDSVR model, the model performance deteriorated with the increase of the forecast horizon. For the perspective of models, the SSA had positively effects on ARMA and KNN whereas the MA had pivotal influences on the improvement of model performance of ANN, CDANN and CDSVR. For the perspective of two different catchments, the MA is suitable for each catchment whilst SSA is more effective for the data of Danjiangkou.

Some discussions were presented (1) on the number of neighbors in KNN; (2) on the configuration of ANN; and (3) on the investigation of effects of MA and SSA. First of all, adopting the number of neighbors k as $m+1$ was considered to be reasonable. Secondly, compared to CDSVR, the poor forecasts from ANN were partly due to the identified configuration of ANN not being optimal. The CDANN model should be expected to have model performance as good as that of CDSVR because it also adopted the local fitting (or distribution) approach. However, obtaining three ANN's configurations with good generalization for CDANN is difficult in view of fewer training data in this study. Conversely, the CDSVR model preferred fewer training data which exponentially improves its training speed but do not obviously affect its generalization due to the implementation of structural risk minimization (SRM) in SVR (*Kecman,2001*). Finally, the techniques of MA and SSA tremendously improved the model performance of ANN (CDANN or CDSVR) by adjusting the correlation relationship between input components and output of models (i.e., increasing the correlation relationship between the earlier input components and the output of model).

References

Abraham, N.B., A.M. Albano, B. Das, G. de Guzman, S. Yong, R.S. Gioggia, G.P. Puccioni, and J.R. Tredicce (1986), Calculating the dimension of attractors from small data. ***Phys. Lett. A***, 114, 217–221.

Abarbanel, H. D. I., R. Brown, J. J. Sidorowich, and L.S. Tsimring (1993), The analysis of observed chaotic data in physical systems, ***Reviews of Modern Physics***, 65(4), 1331-1392.

Abrahart, R.J., and L. See (2002), Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments, ***Hydrology and Earth System Sciences***, 6(4), 655-670.

Behzad, M., K. Asghari, M. Eazi, and M. Palhang (2009), Generalization performance of support vector machines and neural networks in runoff modeling, ***Expert Systems with Applications***, 36 (4), 7624-7629.

Bezdek, J.C. (1981), ***Pattern Recognition with Fuzzy Objective Function Algorithms***. Plenum Press, New York.

Cannas, B., A. Fanni, L. See, and G. Sias, (2006), Data preprocessing for river flow forecasting using neural networks: Wavelet transforms and data partitioning, ***Physics and Chemistry of the Earth***, 31, 1164-1171.

Carlson, R.F., A.J.A. MacCormick, and D.G. Watts (1970), Application of linear models to four annual streamflow series, ***Water Resour. Res.***, 6(4), 1070-1078.

Castellano-Méndeza, M., W. González-Manteigaa, M. Febrero-Bande, J. Manuel Prada-Sáncheza, and R. Lozano-Calderón (2004) Modeling of the monthly and daily behavior of the runoff of the Xallas river using Box–Jenkins and neural networks methods, ***Journal of Hydrology***, 296, 38–58.

Cheng, J., J.S. Qian, and Y.N. Guo (2006), A Distributed Support Vector Machines Architecture for Chaotic Time Series Prediction. Neural Information Processing (ICONIP 2006), Proc. 13th International Conference, Part 1, Hong Kong, China, October 3-6, 2006, Springer.

Cherkassky, V., and Y. Ma (2004), Practical selection of SVM parameters and noise estimation for SVM regression, ***Neural Networks***, 17, 113–126.

Corzo, G. and Solomatine, D. P. (2007). Baseflow separation techniques for modular artificial neural network modelling in flow forecasting. ***Hydrological Science Journal***, 52 (3), 491–507.

Dawson, C. W. and Wilby, R. L.,(1999), A comparison of artificial neural networks used for river flow forecasting, ***Hydrol. Earth Sys. Sci.***,3, 529–540.

Dawson, C. W., and R. L. Wilby (2001). Hydrological Modeling Using Artificial Neural Networks, ***Progress in Physical Geography***, 25(1), 80-108.

De Vos, N.J. and T.H.M. Rientjes (2005), Constraints of artificial neural networks for rainfall -runoff modeling: trade-offs in hydrological state representation and model evaluation, ***Hydrology and Earth System Sciences***, 9, 111-126.

Dibike, Y. B., S. Velickov, D. Solomatine, and M. B. Abbott, (2001). Model Induction with support vector machines: Introduction and Application. ***Journal of Computing in Civil Engineering***, 15(3), 208 – 216.

Elshorbagy, A., S.P. Simonovic, and U.S. Panu (2002), Estimation of missing stream flow data using principles of chaos theory, ***Journal of Hydrology***, 255, 123–133.

Elsner, J. B. and A. A. Tsonis (1997), ***Singular Spectrum Analysis: A New Tool in Time Series Analysis***. Plenum Press, New York.

Essex,C., and M.A.H. Nerenberg (1991), Proc. R. Soc. London, Series A 435, 287.

Farmer, J. D., and J. J. Sidorowich (1987), Predicting chaotic time series, ***Phys. Rev. Lett.***, 59(4), 845–848.

Fraser, A.M. and H.L. Swinney (1986), Independent coordinates for strange attractors from mutual information, ***Physical Review A***, 33(2), 1134-1140.

Ghilardi, P., and R. Rosso (1990), Comment on chaos in rainfall, ***Water Resour. Res.*** 26 (8), 1837–1839.

Grassberger, P., and I. Procaccia (1983), Measuring the strangeness of strange attractors, ***Physica 9D***, 189-208.

Haltiner, J.P., and J.D. Salas (1988), Short-term forecasting of snowmelt discharge using ARMAX models, ***Water Resources Bulletin***, 24(5), 1083-1089.

Hong, S.Z., and S.M., Hong (1994), An amendment to the fundamental limits on dimension calculations. ***Fractals*** 2 (1), 123–125.

Hsu, C. W., C.C. Chang, and C. J. Lin (2003), A practical guide to support vector classification, Available at http://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf. [Accessed on 20/6/07].

Huang, W.R., B. Xu, and A. Hilton (2004), Forecasting Flows in Apalachicola River Using Neural Networks, ***Hydrological Processes***, 18, 2545-2564.

Jain, S. K., A. Das, and D. K. Drivastava (1999). Application of ANN for reservoir inflow prediction and operation. ***J. Water. Resour. Plann. Manage.***, 125(5), 263–271.

Jain, A., and S. Srinivasulu (2006), Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques, ***Journal of Hydrology***, 317, 291–306.

Jain, A., and S. Srinivasulu (2004), Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques, ***Water Resour. Res.***, 40, W04302.

Jayawardena, A.W., and A.B. Gurung (2000), Noise reduction and prediction of hydro-meteorological time series: dynamical systems approach vs. stochastic approach, ***Journal of Hydrology***, 228, 242–264.

Jayawardena, A.W., and F. Lai (1994), Analysis and prediction of chaos in rainfall and stream flow time series, ***Journal of hydrology***, 153, 23-52.

Kantz, H., and T. Schreiber (2004). ***Nonlinear time series analysis (2nd edition)***. Springer.

Kecman, V. (2001), ***Learning and soft computing: support vector machines, neural networks, and fuzzy logic models***, MIT press, Cambridge, Massachusetts.

Kennel, M. B., R. Brown, H. D. I. Abarbanel (1992), Determining embedding dimension for phase space reconstruction using geometrical construction, ***Phy. Rev. A.***, 45(6), 3403-3411.

Kişi, O. (2003). River flow modeling using artificial neural networks, ***Journal of Hydrologic Engineering***, 9(1), 60-63.

Kişi, O. (2005). Daily river flow forecasting using artificial neural networks and auto-regressive models, ***Turkish J. Eng. Env. Sci,*** 29, 9-20.

Kitanidis, P. K. and R. L. Bras (1980), Real-time forecasting with a conceptual hydrologic model, 2, applications and results, ***Wat. Resour. Res.***, 16 (6), 1034–1044.

Kothyari, U.C., and V.P. Singh (1999), A multiple-input single-output model for flow forecasting, ***Journal of Hydrology***, 220, 12–26.

Koutsoyiannis, D., and D. Pachakis (1996), Deterministic chaos versus stochasticity in analysis and modeling of point rainfall series. ***J. Geophys. Res***. 101 (D21), 26441–26451.

Laio, F., A. Porporato, R. Revelli, and L. Ridolfi (2003), A comparison of nonlinear flood forecasting methods, ***Water Resour. Res.***, 39(5), 1129, doi:10.1029/2002WR001551.

Lin, J.Y., C.T. Cheng, and K.W. Chau (2006), Using support vector machines for long-term discharge prediction, ***Hydrological Sciences–Journal***, 51(4), 599-611.

Liong, S.Y., and C. Sivapragasam (2002), Flood stage forecasting with support vector machines, ***Journal of American Water Resour***, 38(1),173 -186.

Lisi, F., O. Nicolis, and M. Sandri (1995), Combining singular-spectrum analysis and neural networks for time series forecasting, ***Neural Processing Letters***, 2(4), 6-10.

María, C.M., G.M. Wenceslao, F.B. Manuel, M.P.S. José, and L.C. Román (2004), Modelling of the monthly and daily behaviour of the discharge of the Xallas river using Box–Jenkins and neural networks methods, ***Journal of Hydrology***, 296,38–58.

Marques, C.A.F., J. Ferreira, A. Rocha, J. Castanheira, P. Gonçalves, N. Vaz., and J.M. Dias (2006), Singular spectral analysis and forecasting of hydrological time series, ***Physics and Chemistry of the Earth***, 31,1172-1179.

Minns, A. W., and M. J. Hall (1996), Artificial neural networks as rainfall-runoff models, ***Hydrol. Sci.***, 41(3), 399–417, 1996.

Muttil, N. and K.W. Chau (2006), Neural network and genetic programming for modelling coastal algal blooms. ***Int. J. Environment and Pollution***, Vol. 28, Nos. 3/4, pp.223–238.

Nash, J. E. and J. V. Sutcliffe (1970), River flow forecasting through conceptual models; part I – a discussion of principles, ***Journal of Hydrology***, 10, 282–290, 1970.

Pasternack, G.B. (1999). Does the river run wild? Assessing chaos in hydrological systems. ***Adv. Water Resour***. 23, 253–260.

Procaccia, I. (1988). Complex or just complicated? ***Nature***, 333, 498–499.

Raman, H., and N. Sunilkumar (1995), Multivariate modeling of water resources time series using artificial neural networks, ***Hydrological Sciences Journal***, 40(2):145-163.

Ruelle, D. (1990), Proc. R. Soc. London, Ser. A 427,241.

Salas, J.D., G.Q. Tabios III, and P. Bartolini (1985), Approaches to multivariate modeling of water resources time series, ***Water Resources Bulletin***, 21(4), 683-708.

Sauer, T., J. A. Yorke, and M. Casdagli (1991), Embedology, ***J. Stat. Phys.***, 65, 579– 616.

Schertzer, D., I. Tchiguirinskaia, S. Lovejoy, P. Hubert, H. Bendjoudi, and M. Larchevêque (2002), Which chaos in the rainfall runoff process? A discussion on Evidence of chaos in the rainfall-runoff process by Sivakumar, ***Hydrological Sciences-Journal,*** 47 (1), 139–147.

See, L., and Openshaw, S. (2000). A hybrid multi-model approach to river level forecasting. ***Hydrological Sciences Journal***, 45 (3), 523–536.

Shrestha, D. L. & Solomatine, D. P. (2006). Experiments with AdaBoostRT, an improved boosting scheme for regression. ***Neural Comput.***, 17.

Sivakumar, B., A. W. Jayawardena, and T. M. K. Fernando (2002), River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches, ***Journal of Hydrology***, 265(1), 225-245.

Sivapragasam, C., and S. Y. Liong (2005). Flow categorization model for improving forecasting. ***Nordic Hydrology***, 36 (1), 37–48.

Sivapragasam, C., S.Y. Liong, and M.F.K. Pasha (2001), Rainfall and discharge forecasting with SSA-SVM approach, ***Journal of Hydroinformatics***, 3(7), 141–152.

Solomatine, D. P., & Ostfeld, A. (2008). Data-driven modelling: Some past experiences and new approaches. ***Journal of Hydroinformatics***, 10(1), 3-22.

Solomatine, D. P. and Xue, Y. I. (2004). M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai River in China. ***Journal of Hydrological Engineering***, 9 (6), 491–501.

Sudheer, K. P., P. C. Nayak, and K. S. Ramasastri (2003), Improving Peak Flow Estimates in Artificial Neural Network River Flow Models, ***Hydrological Processes***, 17, 677-686.

Sugihara, G and R.M. May (1990), Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series, ***Nature*** 344, 734– 741.

Takens, F. (1981), Dynamical systems and Turbulence, ***Lecture note in Mathematics***, Vol. 898, Springer, New York.

Theiler, J., (1986), Spurious dimension from correlation algorithms applied to limited time-series data. ***Phys. Rev. A*** 34 (3), 2427–2432.

Thirumalaiah, K., and M. C. Deo (2000). Hydrological forecasting using neural networks. ***Journalof Hydrologic Engineering***, Vol. 5, No. 2, 180-189.

Vautard, R., P. Yiou, and M. Ghil (1992), Singular-spectrum analysis: a toolkit for short, noisy and chaotic signals, **Physica D** 58, 95–126.

Wang,W., P.H.A.J.M., Van Gelder, J.K. Vrijling, and J. Ma (2006a). Testing for nonlinearity of streamflow processes at different timescales, ***Journal of Hydrology***, 322, 247-268.

Wang, W., P.H.A.J.M. van Gelder, J.K. Vrijling, and J. Ma (2006b), Forecasting Daily Streamflow Using Hybrid ANN Models, ***Journal of Hydrology***, 324, 383-399.

Wu, C.L., K.W. Chau, and Y.S. Li (2008), River stage prediction based on a distributed support vector regression, ***Journal of Hydrology***, 358, 96-111.

Yu, P.S., S.T. Chen, and I.F. Chang (2006), Support vector regression for real-time flood stage forecasting, ***Journal of hydrology***, 328,704-716.

Yu, X.Y., S.Y. Liong, and V. Babovic (2004), EC-SVM approach for real-time hydrologic forecasting, ***Journal of Hydroinformatics***, 6(3), 209-233.

Yu, P.S., and T.Y. Tseng, (1996), A model to forecast flow with uncertainty analysis, ***Hydrological Sciences-Journal***, 41(3), 327-344.

Zhang, B., and R. S. Govindaraju (2000), Prediction of watershed runoff using Bayesian concepts and modular neural networks. ***Water Resources Research***, 36(3), 753-762.

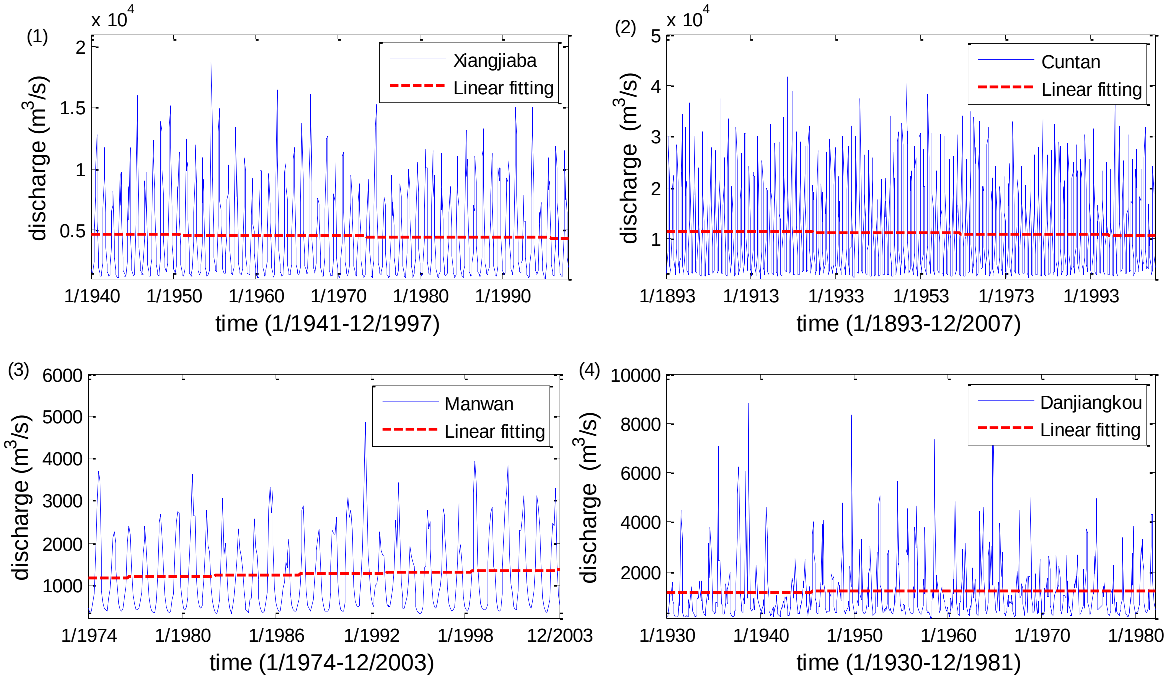


Figure 1. Monthly discharge series of (1) Xiangjiaba, (2) Cuntan, (3) Manwan, and (4) Danjiangkou

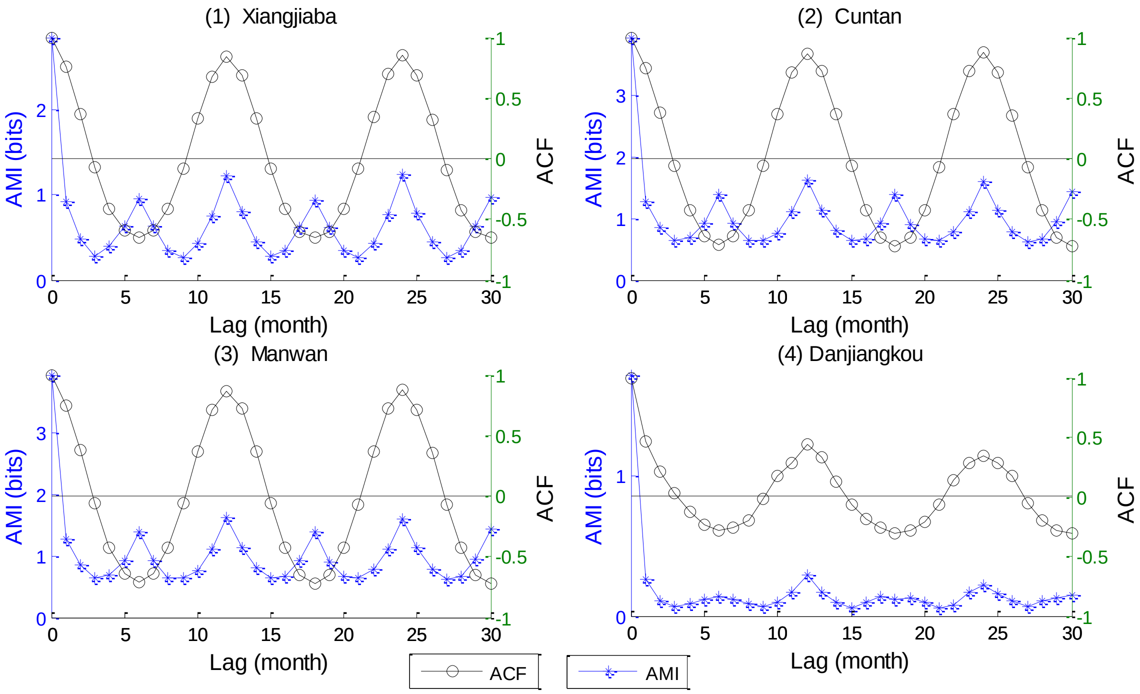


Figure 2. ACF and AMI of (1) Xiangjiaba, (2) Cuntan, (3) Manwan, and (4) Danjiangkou

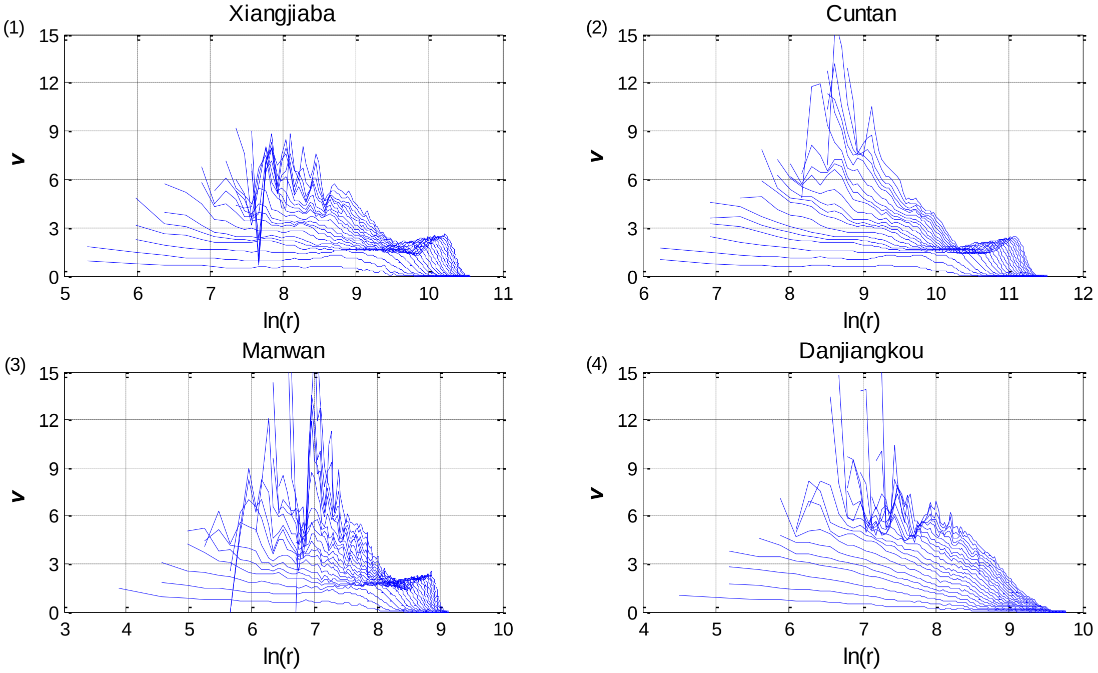


Figure 3. The estimation of correlation dimension (d_2) for (1) Xiangjiaba, (2) Cuntan (3) Manwan, and (4) Danjiangkou. $\tau = 3$ and m is at the interval of [1,20] (increasing from bottom to top in each pane).

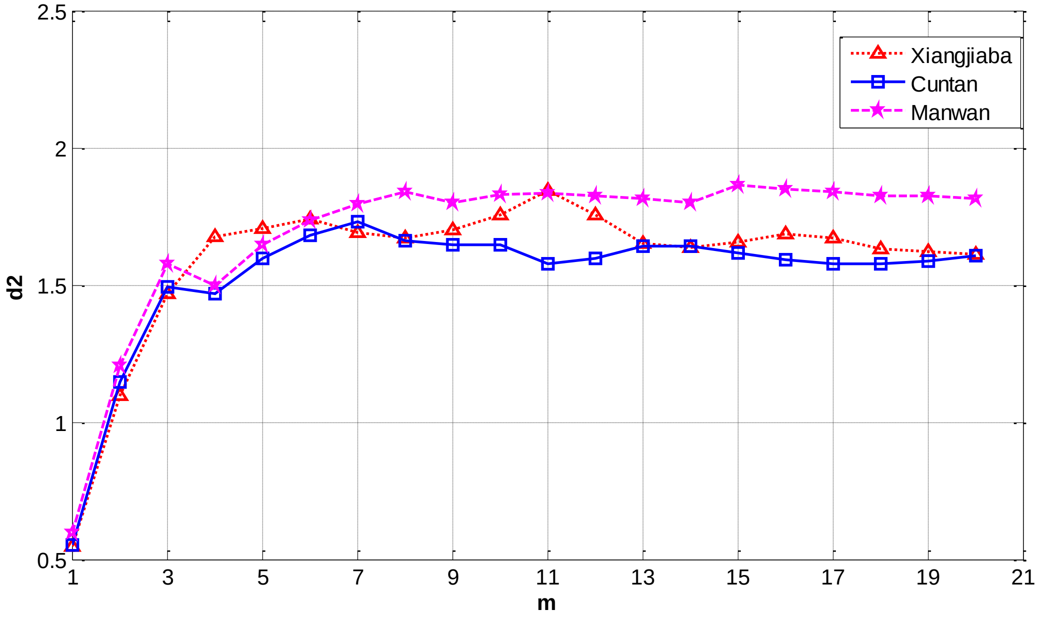


Figure 4. Relationship between d_2 and m for monthly discharges of Xiangjiaba, Cuntan, and Manwan

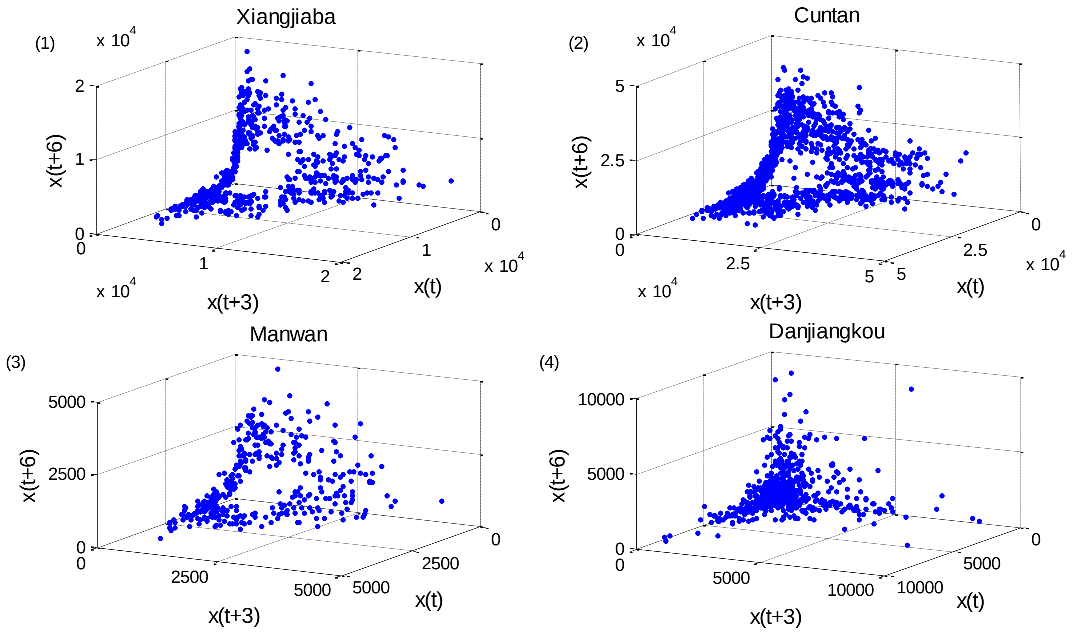


Figure 5. Phase portraits for (1) Xiangjiaba, (2) Cuntan (3) Manwan, and (4) Danjiangkou when $m = 3$ and $\tau = 3$.

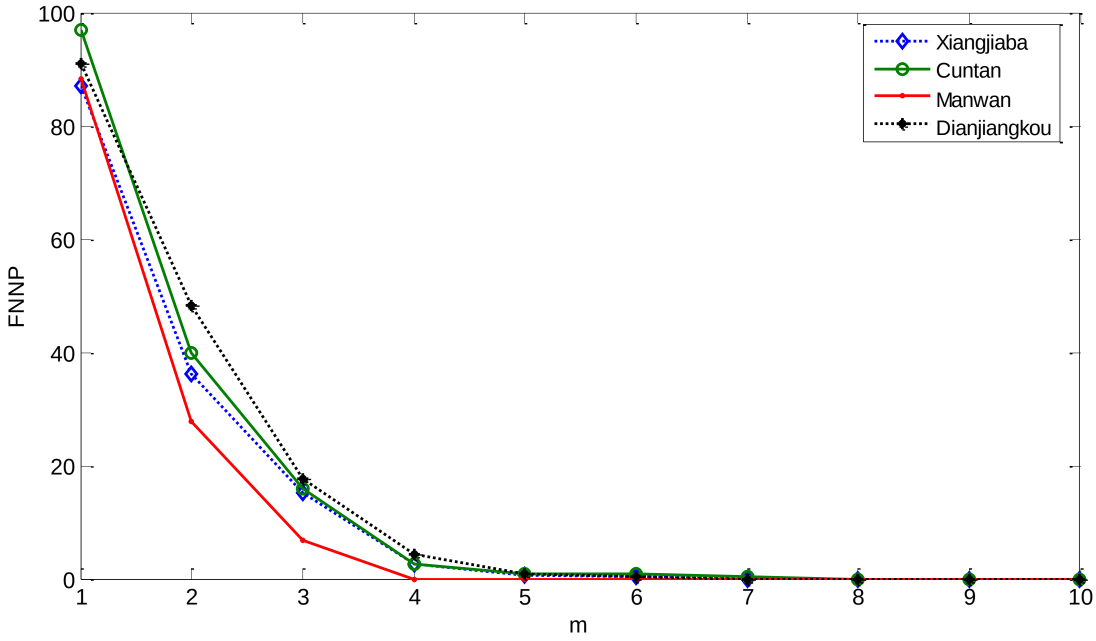


Figure 6. FNNP for Xiangjiaba, Cuntan, Manwan, and Danjiangkou when $R_{tol} = 15$ and $\tau = 3$.

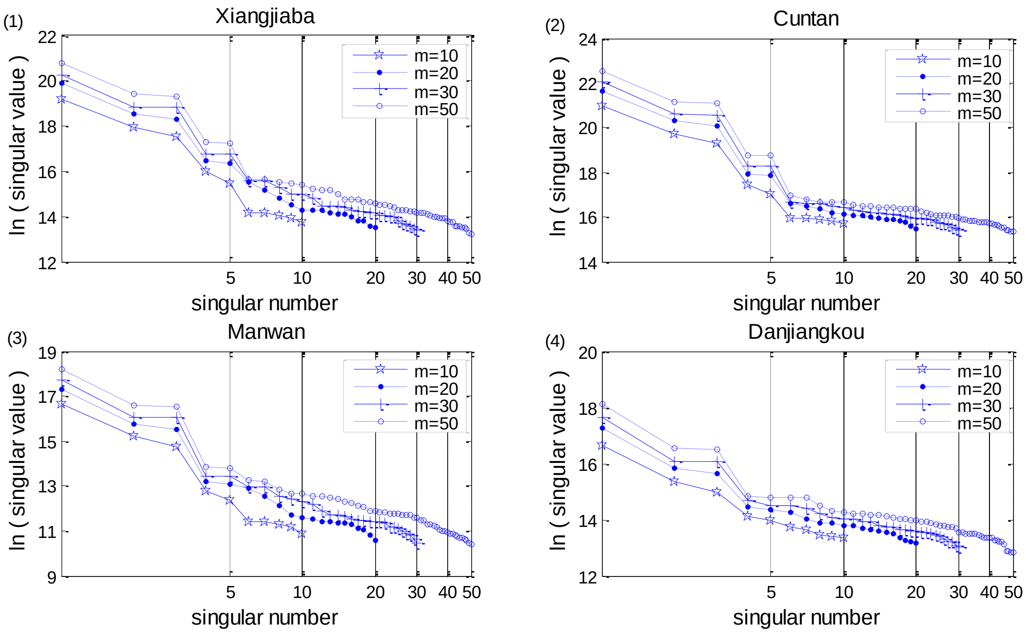


Figure 7. Singular Spectrum for (1) Xiangjiaba, (2) Cuntan (3) Manwan, and (4) Danjiangkou with different m

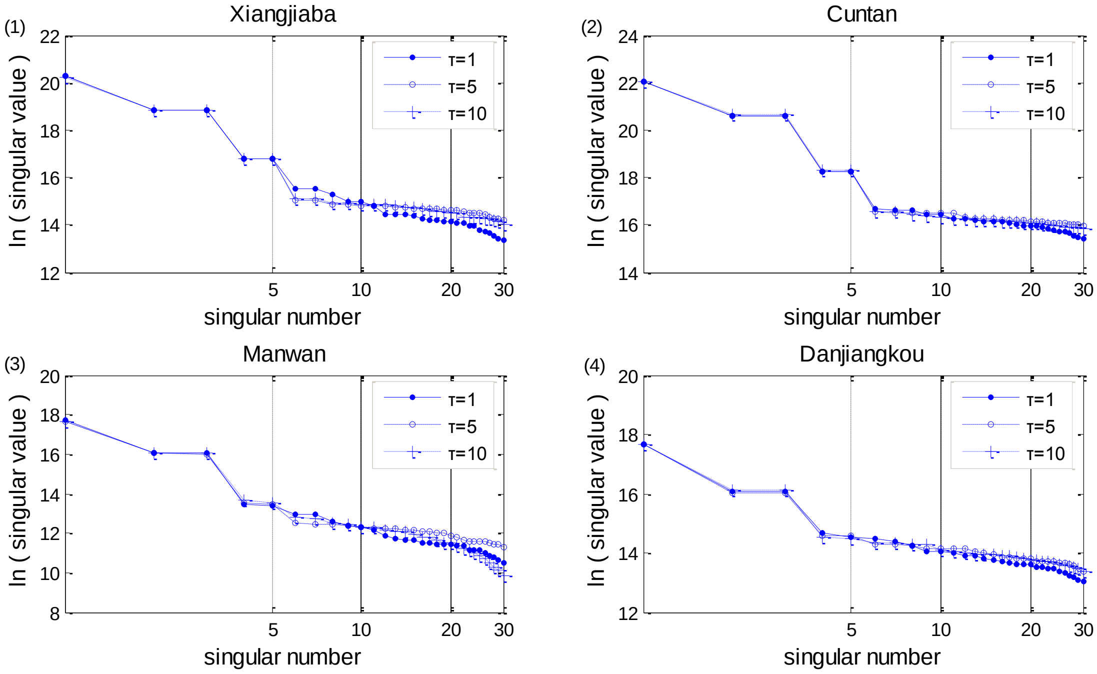
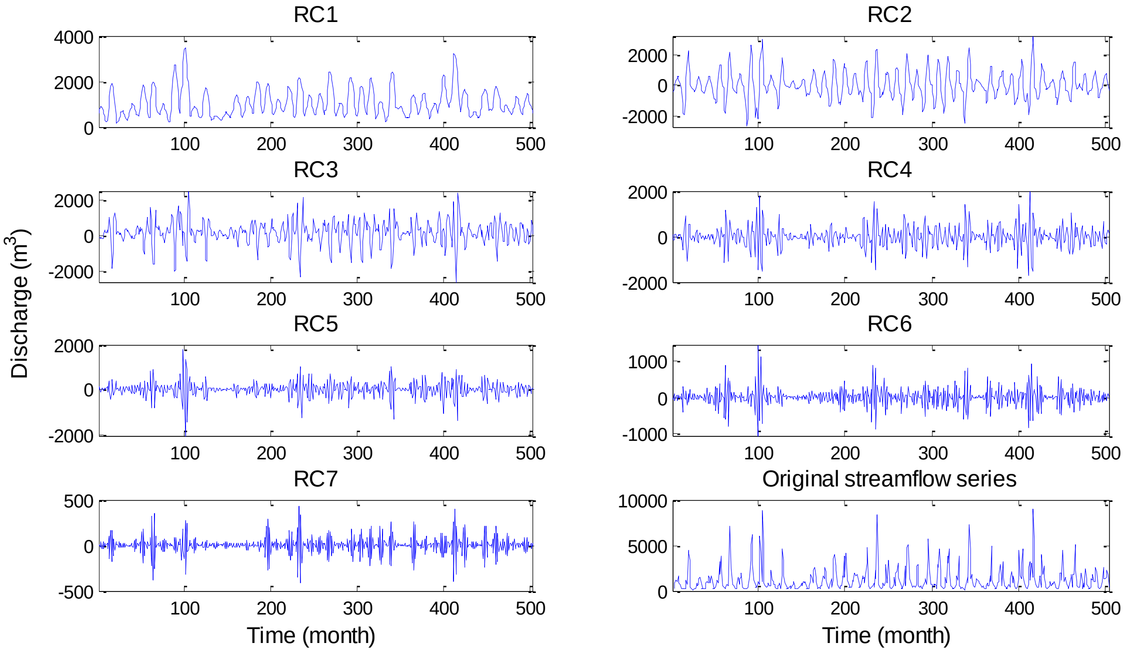
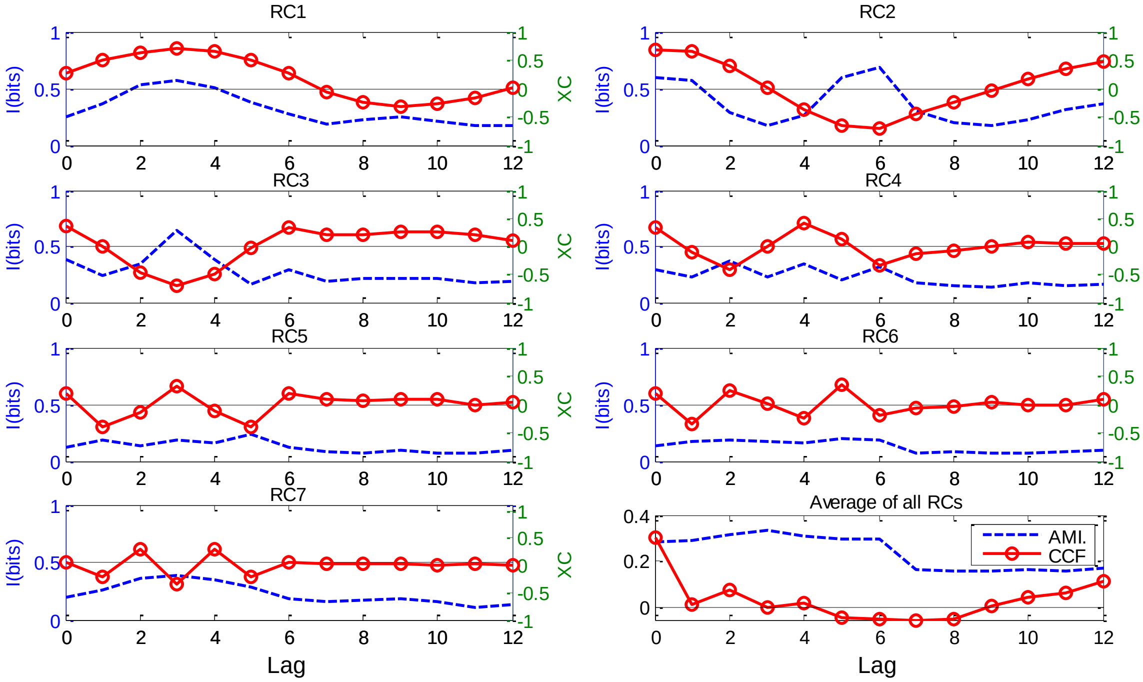


Figure 8. Singular Spectrum for (1) Xiangjiaba, (2) Cuntan (3) Manwan, and (4) Danjiangkou with different τ



1

2 Figure 9. Reconstructed components (RCs) and original streamflow series of Danjiangkou



3

4 Figure 10. AMI and CCF between the original flow series and each RC (Danjiangkou)

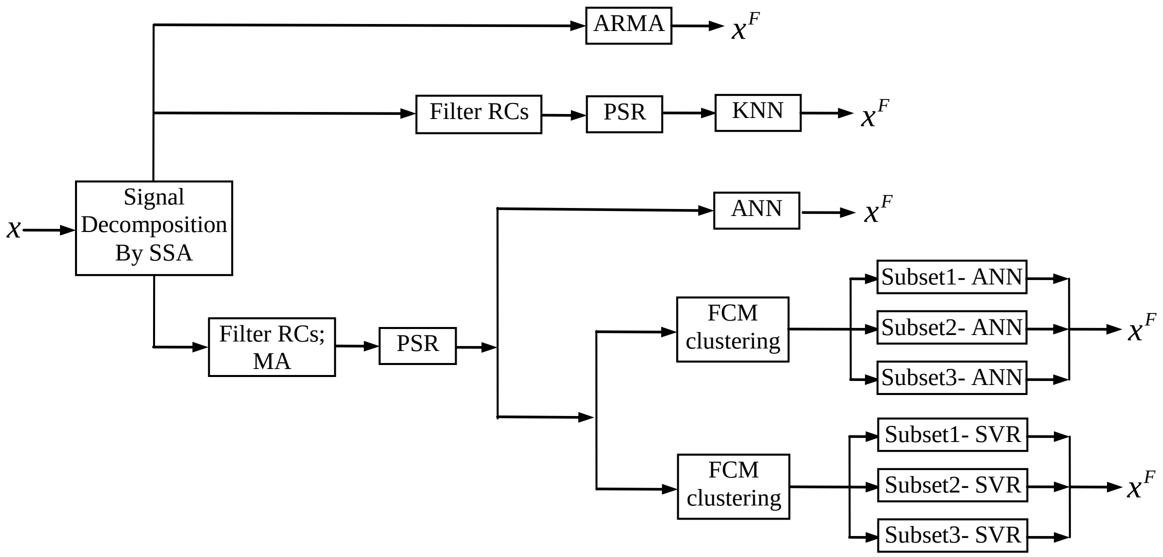


Figure 11. Framework of modelling implementation for monthly streamflow predictions (x^F stands for forecasts)

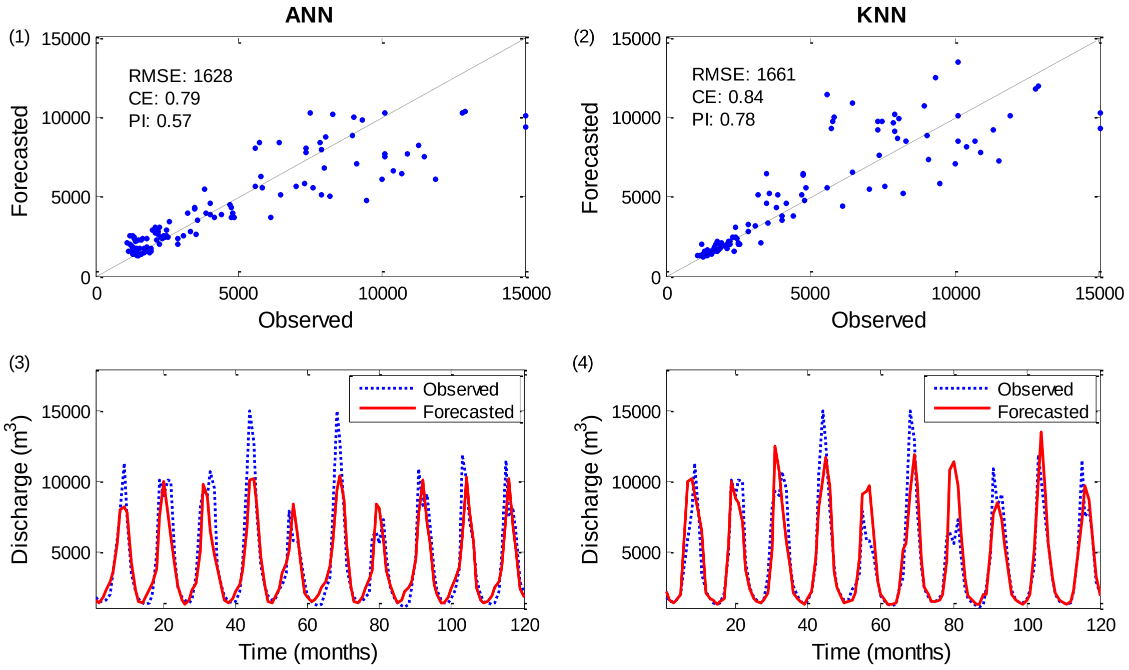
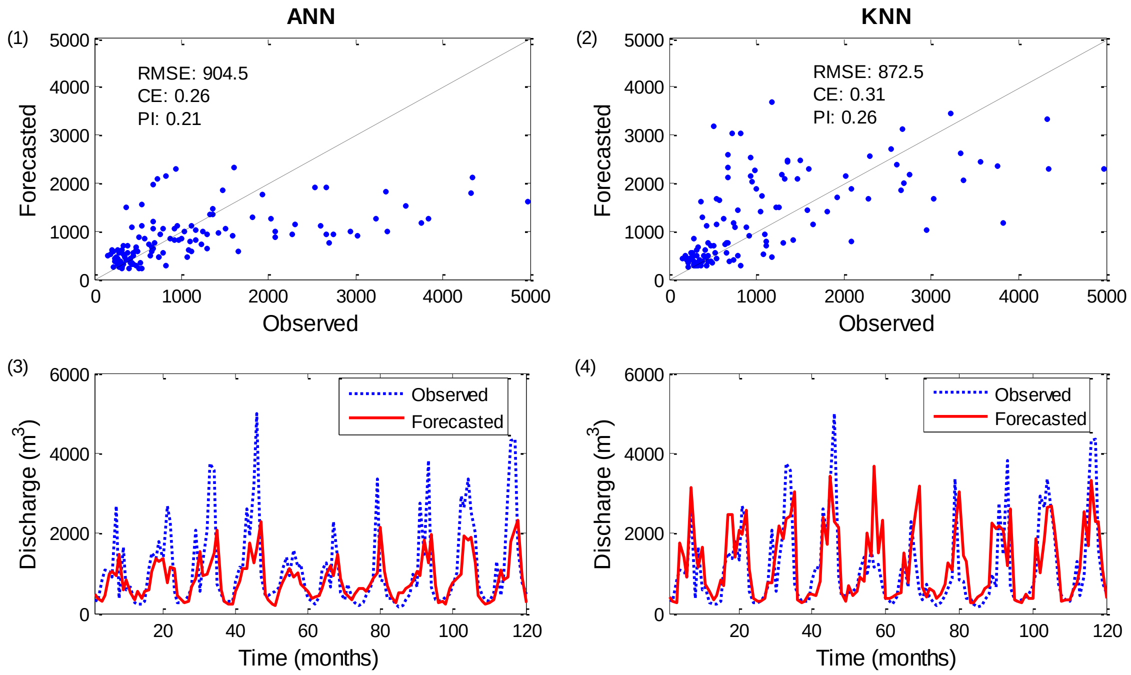


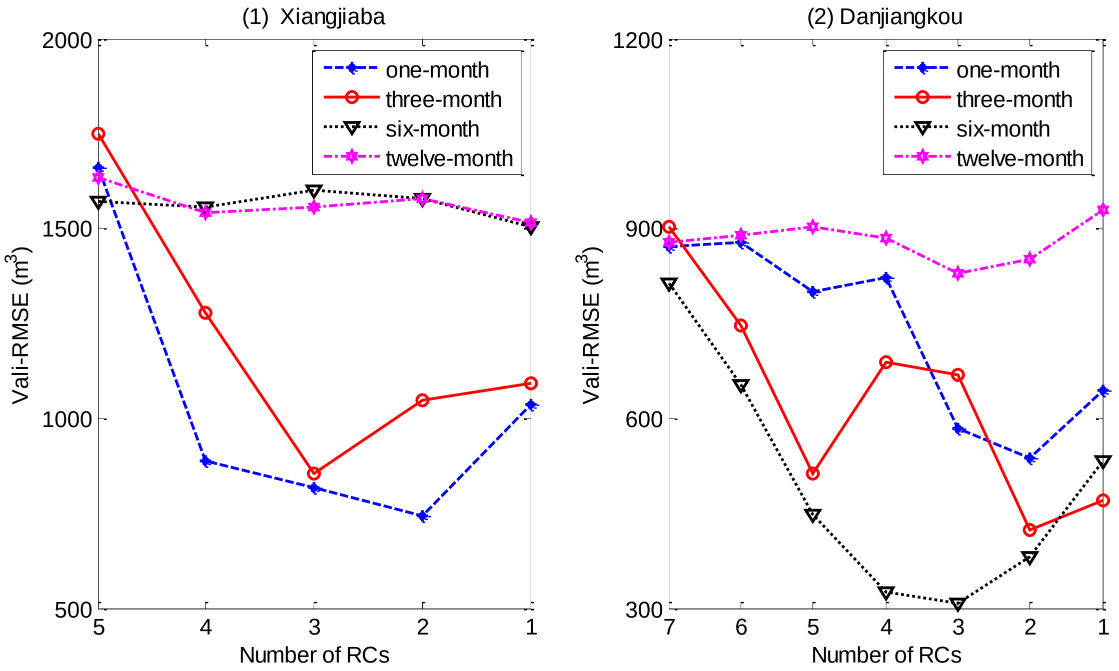
Figure 12. Observed and forecasted Xiangjiaba's discharges for one-month-ahead forecast (ANN modeling results were presented in (1) and (3), and KNN modeling results were presented (2) and (4))



1

2 Figure 13. Observed and forecasted Danjiangkou's discharges for one-month-ahead forecast (ANN modeling

3 results were presented in (1) and (3), and KNN modeling results were presented (2) and (4))



4

5 Figure 14. Performance curves of different prediction horizons by KNN model with various RCs as inputs

6

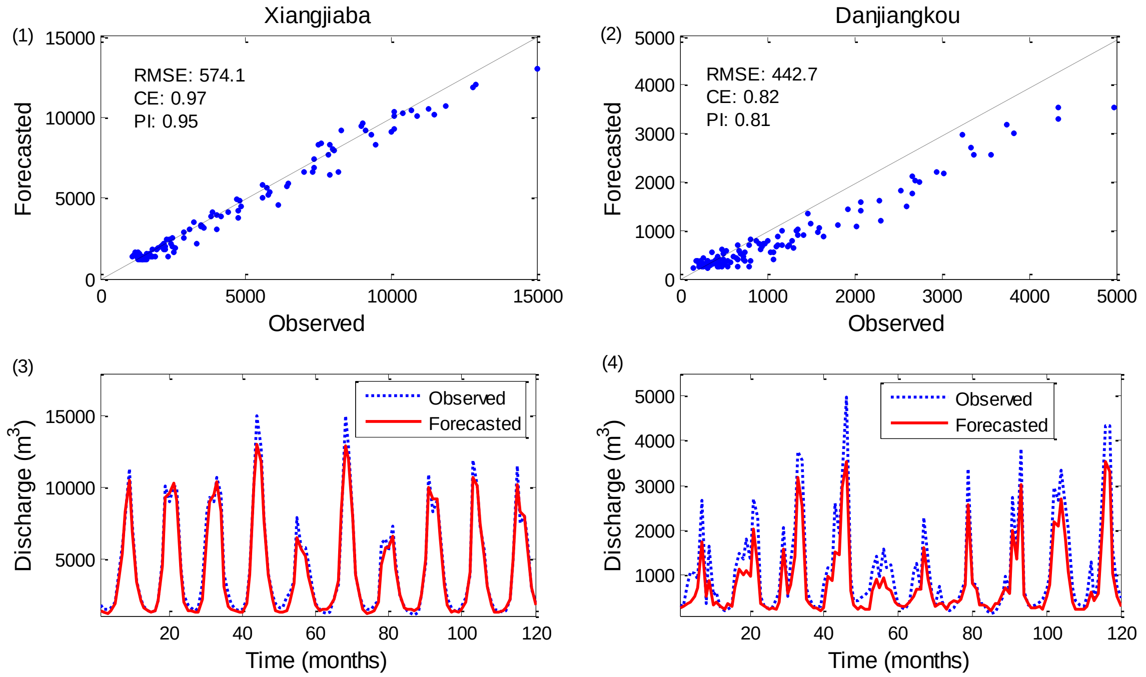


Figure 15. Observed and forecasted discharges for one-month-ahead forecast from ANN model with the data preprocessing of MA and SSA ((1) and (3) for Xiangjiaba; (2) and (4) for Danjiangkou)

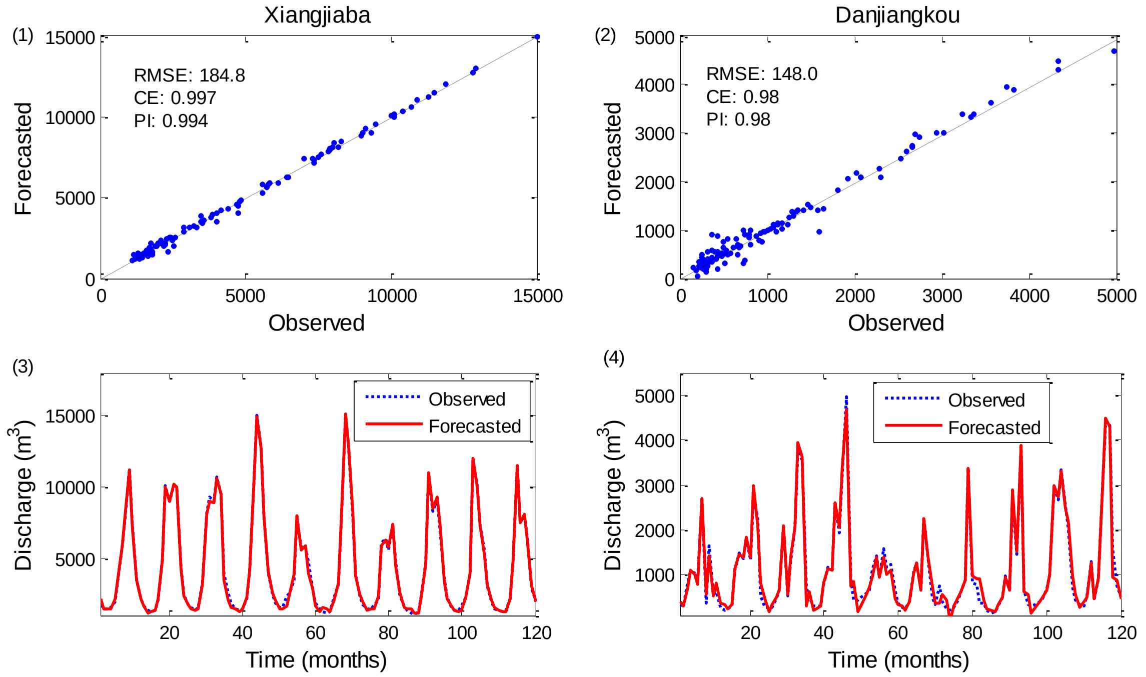
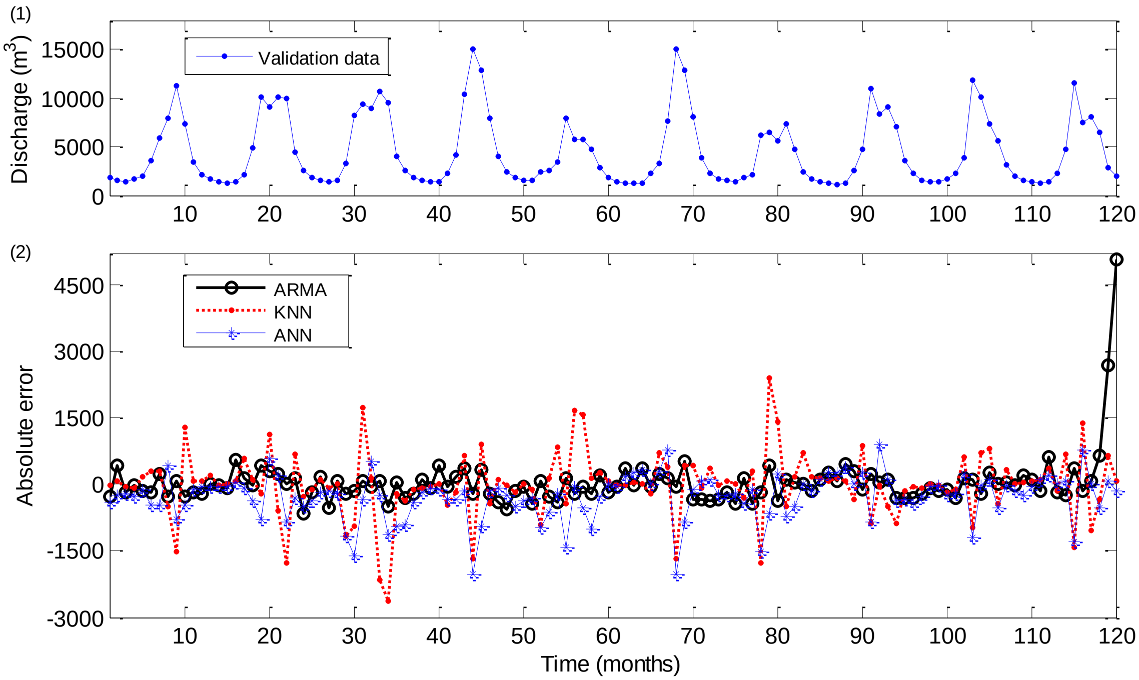


Figure 16. Observed and forecasted discharges for one-month-ahead forecast from CDSVR model with the data preprocessing of MA and SSA ((1) and (3) for Xiangjiaba; (2) and (4) for Danjiangkou)

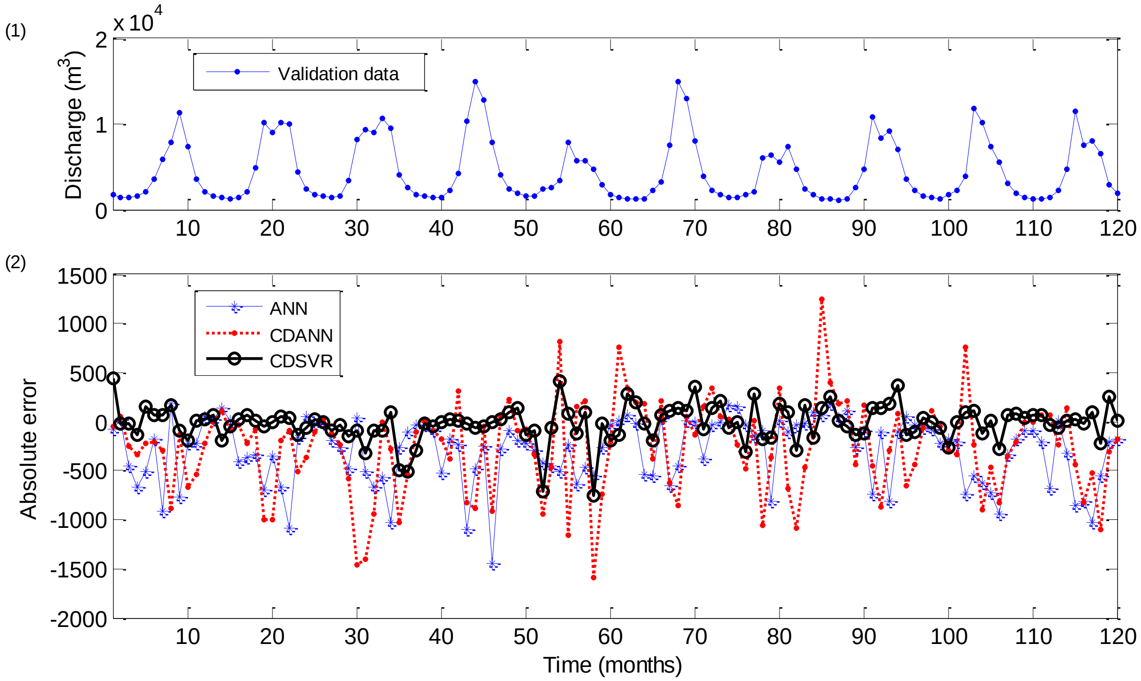
1



2

3 Figure 17a. Forecasting results with MA and SSA for Xiangjiaba (1) validation data (2) absolute error (=

4 forecast - target) of forecasts from ARMA, KNN and ANN



5

Figure 17b. Forecasting results with MA and SSA for Xiangjiaba (1) validation data (2) absolute error (= forecast - target) of forecasts from ANN, CDANN and CDSVR

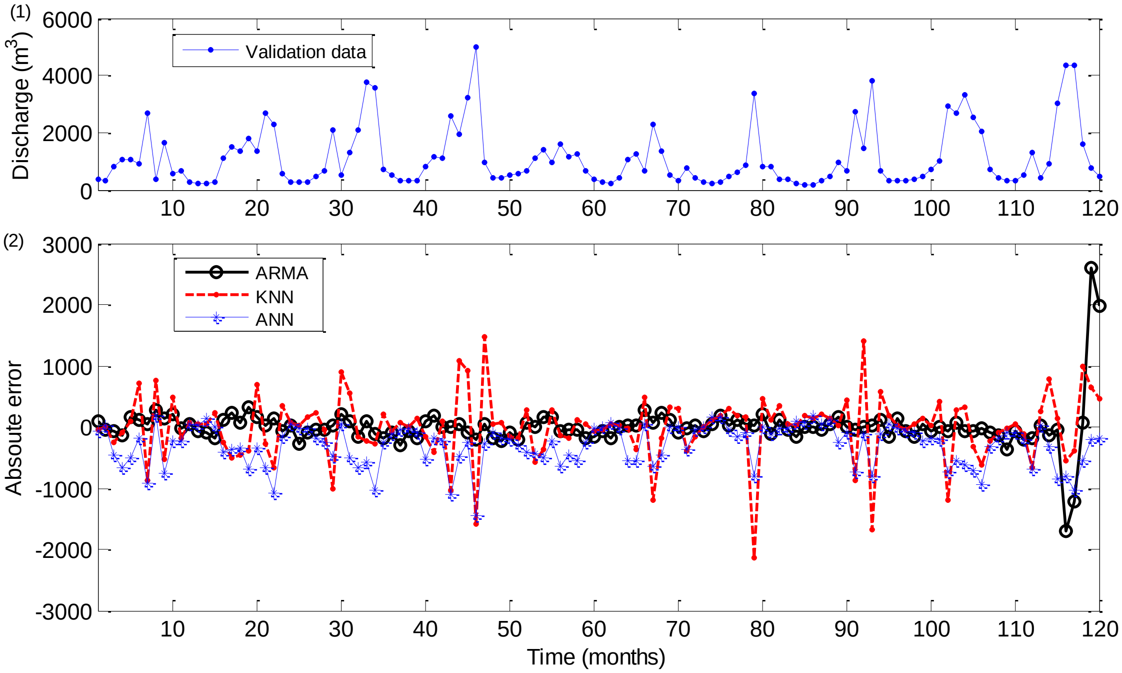


Figure 18a. Forecasting results with MA and SSA for Danjiangkou (1) validation data (2) absolute error (= forecast - target) of forecasts from ARMA, KNN and ANN

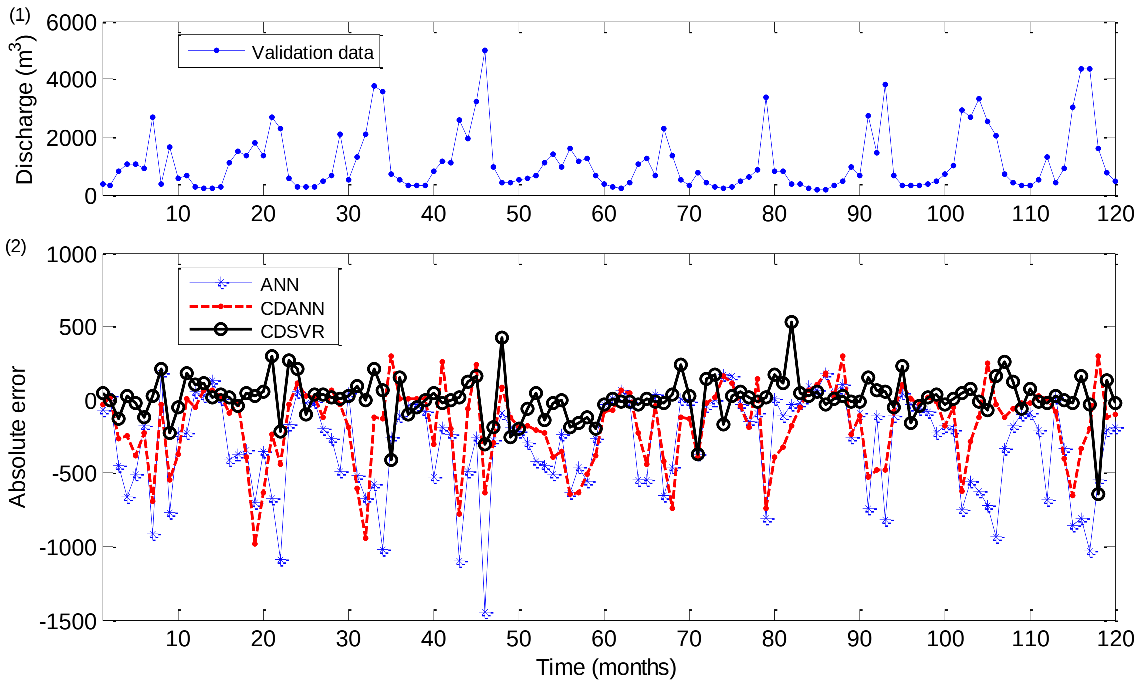


Figure 18b. Forecasting results with MA and SSA for Danjiankou (1) validation data (2) absolute error (= forecast - target) of forecasts from ANN, CDANN and CDSVR

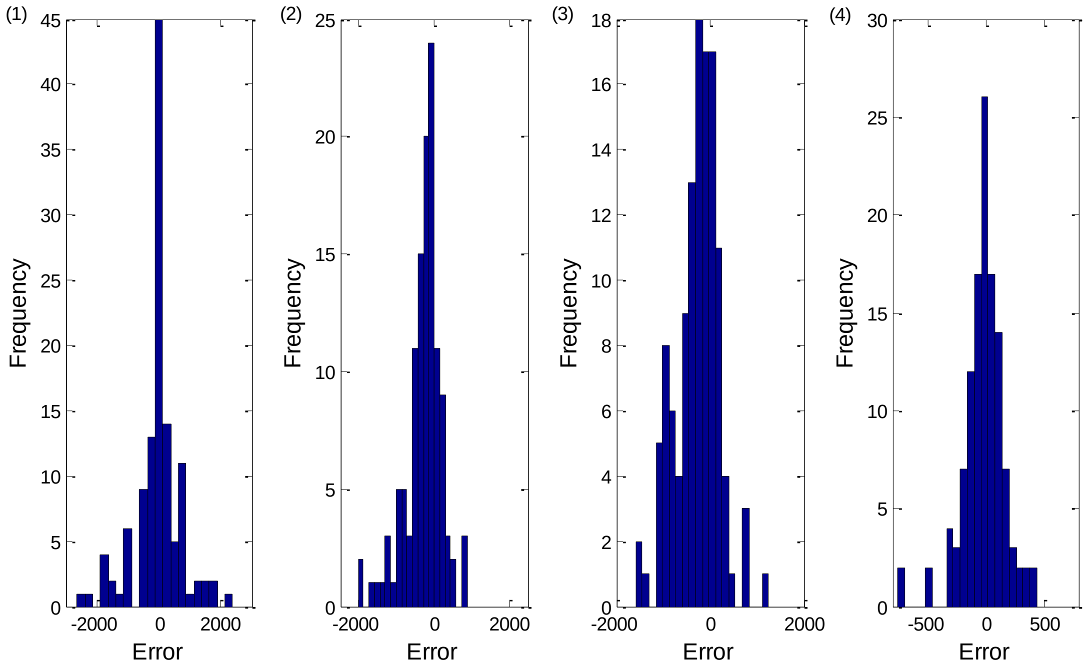


Figure 19. Histograms of absolute prediction errors of (1) KNN, (2) ANN, (3) CDANN, and (4) CDSVR for Danjiangkou

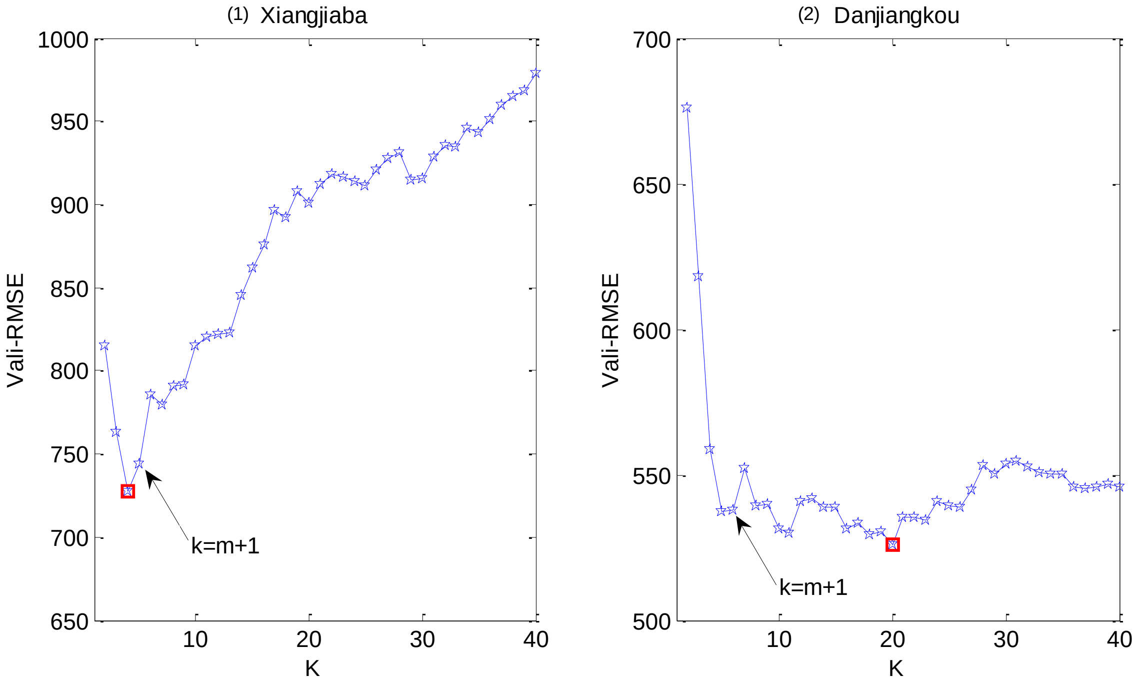


Figure 20. The check of robustness of k in KNN method for (1) Xiangjiaba and (2) Danjiangkou

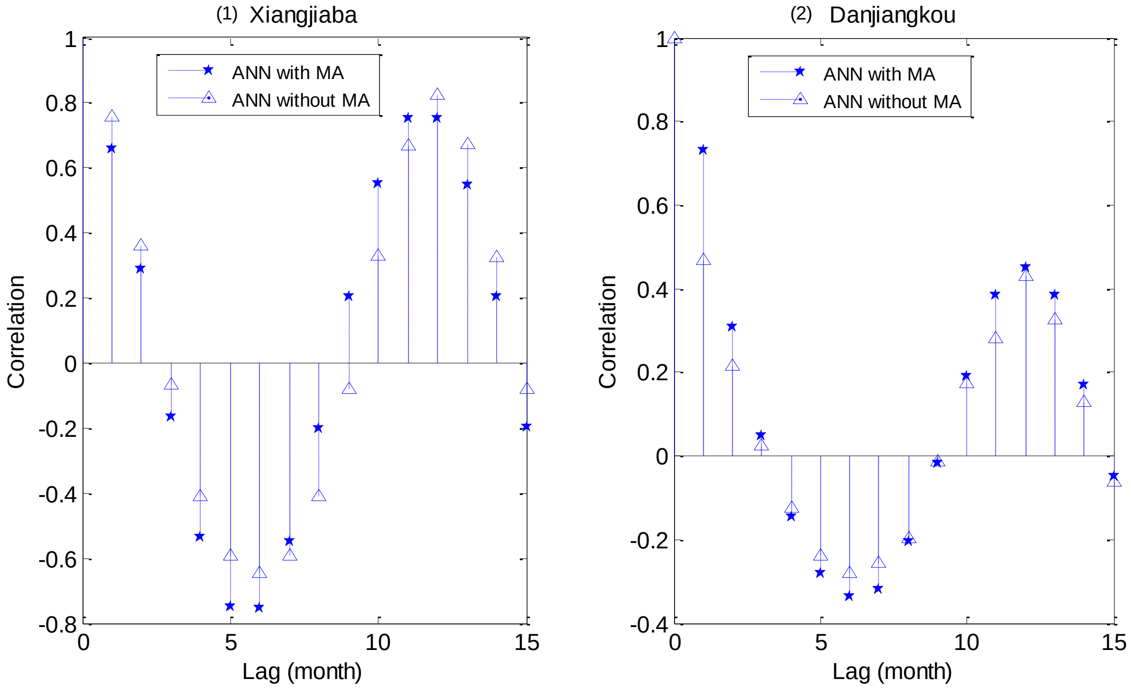


Figure 21. CCFs between inputs and output from ANN with/without MA for one-month-ahead forecast: (1) Cuntan and (2) Danjiangkou.

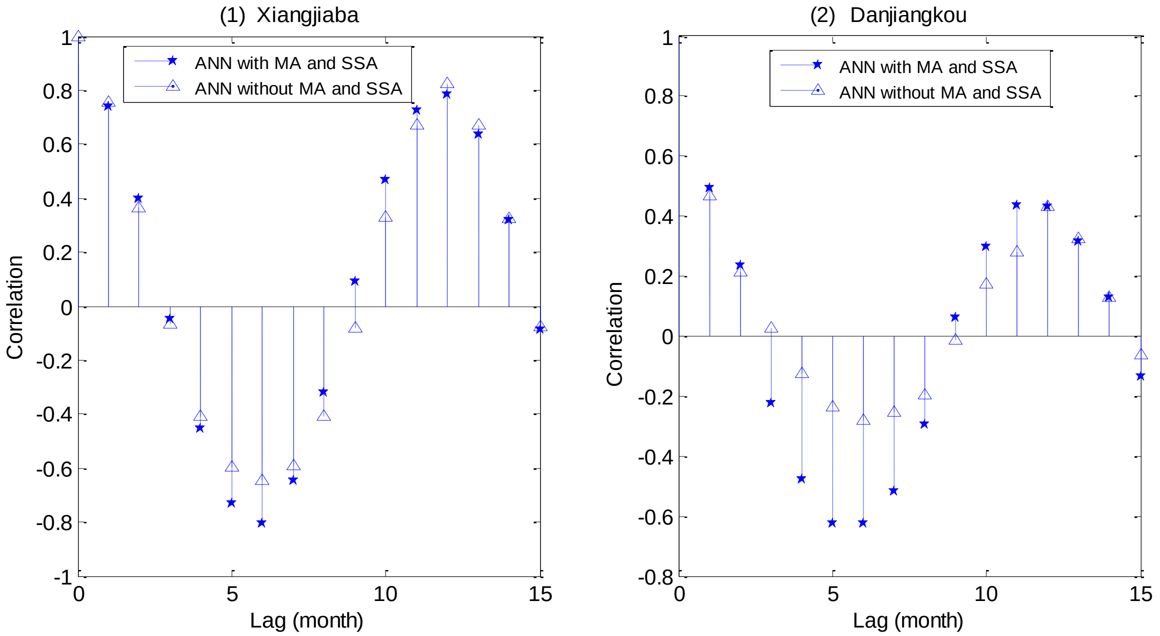


Figure 22. CCFs between inputs and outputs from ANN with/without MA and SSA for six-month-ahead forecast: (1) Cuntan and (2) Danjiangkou.

Table 1 One-month-ahead forecasting results using various models without data preprocessing of MA and/or SSA

Watershed	Model	RMSE	CE	PI	Model parameters			
					(P,Q)	(τ , m , k)	(I-h-O)*[a]	(C,ε,σ)
Xiangjiaba								
	ARMA	1293.4	0.87	0.73	(12,11)			
	KNN	1661.2	0.78	0.55		(3,4,5)		
	ANN	1627.8	0.79	0.57			(4-8-1)	
	CDANN	1427.1	0.84	0.70			(4-8/8/8-1)	
	CDSVR	1407.1	0.84	0.68				3*[b]
Danjiangkou								
	ARMA	762.3	0.47	0.44	(11,10)			
	KNN	872.5	0.31	0.26		(3,5,6)		
	ANN	904.5	0.26	0.21			(5-5-1)	
	CDANN	895.3	0.27	0.42			(5-5/5/5-1)	
	CDSVR	690.3	0.57	0.54				3*[c]

*[a] (I-h-O) stands for the number of neurons in the input layer, hidden layer and output layer;

3*[b] three triplet parameters (C,ε,σ) for Xiangjiaba are (46.96, 0.0056, 0.3278), (737.66, 0.0062, 0.8065), and (0.66128, 0.0248, 0.4205);

3*[c] three triplet parameters (C,ε,σ) for Danjiangkou are (773.95, 0.0002, 0.8264), (3.75, 0.0092, 0.1124), and (3.99, 0.0068, 0.5342).

Table 2 One-month-ahead forecasting results using various models with data preprocessing of MA and/or SSA

Watershed	Model	RMSE	CE	PI	Model parameters				Data-preprocessing parameters	
					(P_i,Q_i)	(τ,m,k)	(I-h-O)	(C,ε,σ)	Memory length of MA (months)	p from m RCs of SSA
Xiangjiaba										
	ARMA	586.2	0.97	0.94	*[a]				1	5/5
	KNN	744.4	0.96	0.91		(3,4,5)			1	2/5
	ANN	574.1	0.97	0.95			(4-8-1)		6	5/5
	CDANN	540.8	0.98	0.95			(4-8/8/8-1)		6	5/5
	CDSVR	184.8	1.00	0.99				3*[c]	6	5/5
Danjiangkou										
	ARMA	376.6	0.87	0.86	*[b]				1	7/7
	KNN	538.1	0.74	0.72		(3,5,6)			1	2/7
	ANN	442.7	0.82	0.81			(5-5-1)		3	7/7
	CDANN	298.7	0.92	0.82			(5-5/5/5-1)		3	7/7
	CDSVR	148.0	0.98	0.98				3*[d]	3	7/7

*[a] (P_i,Q_i) denotes optimal parameter pairs for each RC, $i=1,\cdots,5$ for Xiangjiaba, and P_i and Q_i are respectively (10,9,10,11,9) and (12,12,12,12,12);

*[b] (P_i,Q_i) denotes optimal parameter pairs for each RC, $i=1,\cdots,7$ for Danjiangkou, and P_i and Q_i are respectively (5,5,10,8,7,10,5) and (10,10,11,10,12,10,12);

3*[c] three triplet parameters (C, ε, σ) for Xiangjiaba are (454.51, 0.0031, 0.5911), (53.26, 0.0082, 0.3299), and (731.14, 0.0003, 0.7788);

3*[d] three triplet parameters (C, ε, σ) for Danjiangkou are (485.64, 0.0023, 0.4321), (0.38, 0.0007, 0.2839), and (805.86, 0.0006, 8.4053).

Table 3 Three-month-ahead forecasting results using various models with data preprocessing of MA and/or SSA

					Model parameters				Data-preprocessing parameters	
Watershed	Model	RMSE	CE	PI	(P_i,Q_i)	(τ, m, k)	(I-h-O)	(C, ε, σ)	Memory length of MA (months)	p from m RCs of SSA
Xiangjiaba										
	ARMA	606.01	0.97	0.99	*[a]				1	5/5
	KNN	853.87	0.94	0.97		(3,4,5)			1	3/5
	ANN	795.83	0.95	0.98			(4-8-1)		8	5/5
	CDANN	601.85	0.97	0.95			(4-8/8/8-1)		8	5/5
	CDSVR	435.05	0.98	0.97				3*[c]	8	5/5
Danjiangkou										
	ARMA	450.13	0.82	0.91	*[b]				1	7/7
	KNN	423.42	0.84	0.92		(3,5,6)			1	2/7
	ANN	533.49	0.74	0.87			(5-5-1)		5	4/7
	CDANN	407.85	0.85	0.69			(5-5/3/2-1)		5	4/7
	CDSVR	305.64	0.92	0.91				6*[d]	5	4/7

*[a] (P_i,Q_i) are the same as one-month-ahead forecast;

*[b] (P_i,Q_i) are the same as one-month-ahead forecast;

3*[c] three triplet parameters (C, ε, σ) for Xiangjiaba are (437.77, 0.0006, 0.3696), (10.01, 0.0050, 0.32593), and (36.25, 0.0053, 0.3683);

6*[d] six triplet parameters (C, ε, σ) for Danjiangkou are (916.01, 0.0025, 2.579), (34.01, 0.0003, 8.5518), (221.38, 0.0004, 4.1752), (0.56, 0.0006, 0.2186), (64.41, 0.0005, 4.9333), and (149.37, 0.0024, 0.3234).

Table 4 Six-month-ahead forecasting results using various models with data preprocessing of MA and/or SSA

Watershed	Model	RMSE	CE	PI	Model parameters				Data-preprocessing parameters	
					(P_i,Q_i)	(τ, m, k)	(I-h-O)	(C, ε, σ)	Memory length of MA (months)	p from m RCs of SSA
Xiangjiaba										
	ARMA	1519.30	0.82	0.94	*[a]				1	5/5
	KNN	1503.00	0.82	0.94		(3,4,5)			1	1/5
	ANN	1535.70	0.81	0.94			(4-8-1)		6	4/5
	CDANN	1193.60	0.89	0.81			(4-8/8/6-1)		6	4/5
	CDSVR	1137.50	0.90	0.79				3*[c]	6	4/5
Danjiangkou										
	ARMA	326.04	0.90	0.96	*[b]				1	7/7
	KNN	308.82	0.91	0.97		(3,5,6)			1	3/7
	ANN	495.14	0.78	0.91			(5-5-1)		6	4/7
	CDANN	414.10	0.84	0.75			(5-3/6/2-1)		6	4/7
	CDSVR	413.68	0.84	0.83				6*[d]	6	4/7

*[a] (P_i,Q_i) are the same as one-month-ahead forecast;

*[b] (P_i,Q_i) are the same as one-month-ahead forecast;

3*[c] three triplet parameters (C, ε, σ) for Xiangjiaba are (437.77, 0.0006, 0.3696), (10.01, 0.0050, 0.32593), and (36.25, 0.0053, 0.3683);

6*[d] six triplet parameters (C, ε, σ) for Danjiangkou are (916.01, 0.0025, 2.579), (34.01, 0.0003, 8.5518), (221.38, 0.0004, 4.1752), (0.56, 0.0006, 0.2186), (64.41, 0.0005, 4.9333), and (149.37, 0.0024, 0.3234).

Table 5 Twelve-month-ahead forecasting results using various models with data preprocessing of MA and/or SSA

Watershed	Model	RMSE	CE	PI	Model parameters				Data-preprocessing parameters	
					(P_i,Q_i)	(τ, m, k)	(I-h-O)	(C, ε, σ)	Memory length of MA (months)	p from m RCs of SSA
Xiangjiaba										
	ARMA	2157.10	0.63	0.09	*[a]				1	5/5
	KNN	1516.20	0.82	0.55		(3,4,5)			1	1/5
	ANN	1657.20	0.78	0.46			(4-8-1)		1	5/5
	CDANN	1440.60	0.83	0.86			(4-8/8/8-1)		2	5/5
	CDSVR	1258.70	0.87	0.74				6*[c]	4	5/5
Danjiangkou										
	ARMA	848.35	0.35	-0.13	*[b]				1	7/7
	KNN	851.94	0.34	-0.14		(3,5,6)			1	3/7
	ANN	817.83	0.39	-0.05			(5-7-1)		1	4/7
	CDANN	634.35	0.63	0.64			(5-5/3/2-1)		1	4/7
	CDSVR	627.10	0.64	0.62				6*[d]	4	4/7

*[a] (P_i,Q_i) are the same as one-month-ahead forecast;

*[b] (P_i,Q_i) are the same as one-month-ahead forecast;

6*[c] six triplet parameters (C,ε,σ) for Xiangjiaba are (0.67, 0.0064, 0.0685), (39.71, 0.0062, 1.3351), (473.02, 0.0028, 0.9397), (42.88, 0.0294, 0.4906), (7.24, 0.0004, 0.5580), and (0.67, 0.0056, 0.0632).

6*[d] six triplet parameters (C,ε,σ) for Danjiangkou are (342.6, 0.0068, 2.1952), (5.30, 0.0363, 3.7493), (167.4, 0.0837, 7.060), (0.9507, 0.0008, 0.0837), (0.61, 0.0059, 0.2515), and (29.35, 0.0042, 0.9685).