# A hybrid model coupled with singular spectrum analysis for daily rainfall prediction

K. W. Chau and C. L. Wu

## ABSTRACT

A hybrid model integrating artificial neural networks and support vector regression was developed for daily rainfall prediction. In the modeling process, singular spectrum analysis was first adopted to decompose the raw rainfall data. Fuzzy C-means clustering was then used to split the training set into three crisp subsets which may be associated with low-, medium- and high-intensity rainfall. Two local artificial neural network models were involved in training and predicting low- and medium-intensity subsets whereas a local support vector regression model was applied to the high-intensity subset. A conventional artificial neural network model was selected as the benchmark. The artificial neural network with the singular spectrum analysis was developed for the purpose of examining the singular spectrum analysis technique. The models were applied to two daily rainfall series from China at 1-day-, 2-day- and 3-day-ahead forecasting horizons. Results showed that the hybrid support vector regression model performed the best. The singular spectrum analysis model also exhibited considerable accuracy in rainfall forecasting. Also, two methods to filter reconstructed components of singular spectrum analysis, supervised and unsupervised approaches, were compared. The unsupervised method appeared more effective where nonlinear dependence between model inputs and output can be considered.

**Key words** | artificial neural network, daily rainfall prediction, fuzzy C-means, hybrid models, singular spectral analysis, support vector regression

**K. W. Chau** (corresponding author)
**C. L. Wu**
Department of Civil and Structural Engineering,
Hong Kong Polytechnic University,
Hung Hom,
Kowloon,
Hong Kong
E-mail: *cekwchau@polyu.edu.hk*

## NOMENCLATURE

### Acronyms

| | |
|---|---|
| ACF | Auto-correlation function |
| AMI | Average mutual information |
| ANN | Artificial neural networks |
| CE | Coefficient of efficiency |
| FCM | Fuzzy *C*-means |
| FNN | False nearest neighbors |
| LM | Levernberg–Marquardt |
| MLP | Multilayer perceptron |
| MOGA | Multi-objective genetic algorithm |
| NNM | Nearest-neighbor method |
| PACF | Partial auto-correlation function |
| PMI | Partial mutual information |
| RCs | Reconstructed components |
| RMSE | Root mean square error |
| SRM | Structural risk minimization |
| SSA | Singular spectrum analysis |
| SVD | Singular value decomposition |
| SVM | Support vector machine |
| SVR | Support vector regression |

### Symbols

| | |
|---|---|
| $\{x_i\}_{i=1,\ldots N}$ | Raw rainfall time series |
| $N$ | Length size of raw rainfall time series |
| $y$ | Observed (or target) value |
| $\hat{y}$ | Forecasted value |
| $\bar{y}$ | Average of observed value |

doi: 10.2166/hydro.2010.032

| | |
|---|---|
| $\boldsymbol{X}$ | Trajectory matrix |
| $\boldsymbol{S}$ | Lagged-covariance matrix |
| $\boldsymbol{D}$ | Left singular vectors of $\boldsymbol{X}$ |
| $\boldsymbol{E}$ | Right singular vectors of $\boldsymbol{X}$ |
| $\boldsymbol{L}$ | Diagonal matrix of singular values |
| $\lambda$ | Eigenvalues of $\boldsymbol{S}$ |
| $a^k$ | Principal component vector $k$ |
| $e^k$ | Eigenvector $k$ of $\boldsymbol{S}$ |
| $n$ | The length size of principle component vector |
| $\tau$ | Delay (or lagged) time |
| $L$ | The window length (or singular number) |
| $p$ | Number of selected RCs |
| $\boldsymbol{Y}$ | Model input vector consisting of $\{x_i\}_{i=1,\ldots N}$ |
| $m$ | Dimension of input vector $\boldsymbol{Y}$ |
| $h$ | Number of nodes in the hidden layer of ANN |
| $T$ | Prediction lead time |
| $w$ | Weight in ANN |
| $\theta$ | Bias in ANN |
| $\omega$ | Weight in the SVR equation |
| $b$ | Bias in the SVR equation |
| $C$ | Positive constant in the SVR equation |
| $\sigma$ | Standard deviation of the rainfall data |
| $\epsilon$ | Error tolerance |
| $\xi,\xi^*$ | Slack variables |
| $s$ | Number of support vectors |
| $v$ | Cluster center |
| $\boldsymbol{U}$ | Matrix consisting of membership grade |
| $c$ | Number of clusters |

## INTRODUCTION

An accurate and timely rainfall forecast is crucial for reservoir operation and flooding prevention because it can provide an extension of lead-time of the flow forecast. Many studies have employed soft computing methods, including artificial neural networks (ANN), support vector regression (SVR) and fuzzy logic (FL), for quantitative precipitation (or rainfall) forecast (QPF) (Venkatesan *et al.* 1997; Silverman & Dracup 2000; Toth *et al.* 2000; Pongracz *et al.* 2001; Sivapragasam *et al.* 2001; Brath *et al.* 2002; Bray & Han 2004; Hettiarachchi *et al.* 2005; Han *et al.* 2007; Chattopadhyay & Chattopadhyay 2007, 2008; Chen *et al.* 2008; Guhathakurta 2008). For example, Venkatesan *et al.* (1997) employed the ANN to predict the all-India summer monsoon rainfall with different meteorological parameters as model inputs. Toth *et al.* (2000) found that the ANN performed the best for short-term rainfall prediction compared with the auto-regressive moving average and the nearest-neighbors method (NNM). Fuzzy logic theory was applied to monthly rainfall prediction by Pongracz *et al.* (2001). Depending on the rainfall series alone, Chattopadhyay & Chattopadhyay (2008) constructed an ANN model to predict monsoon rainfall in India.

Physical processes in rainfall and/or runoff are generally composed of a number of sub-processes. For example, base flow mainly contributes to low-flow events whereas intense storm rainfall gives rise to high-flow events. Their accurate modeling by the building of a single global model is sometimes not possible (Solomatine & Ostfeld 2008). Modular (or hybrid) models are therefore developed where sub-processes are first of all identified and then separate models (also called local or expert models) are established for each of them. A comprehensive review of modular models can be found in Solomatine & Ostfeld (2008). Several examples of modular models can be mentioned herein. See & Openshaw (2000) built different neural networks based on different types of hydrological events. Zhang & Govindaraju (2000) examined the performance of modular networks in predicting monthly discharges based on the Bayesian concept. Hu *et al.* (2001) developed a range-(threshold-) dependent network which employs a number of MLPNNs to model the river flow in different flow bands of magnitude (e.g. high, medium and low). Their results indicated that the range-dependent network performed better than the conventional global ANN. Solomatine & Xue (2004) applied M5 model trees and neural networks to a flood-forecasting problem. Sivapragasam & Liong (2005) divided the flow range into three regions and employed different SVR models to predict daily flows in high, medium and low regions. Wang *et al.* (2006) used a combination of ANNs for flow prediction where different networks were trained on the data subsets determined by applying either a threshold discharge value

or clustering in the space of inputs. Wu *et al.* (2008) employed a distributed SVR for daily river stage prediction.

Apart from the adoption of the modular model, a suitable signal filter technique, removing noises in the original hydrologic time series, can be effective in improving forecasts. As such important techniques, singular spectrum analysis (SSA) and wavelet analysis were recently introduced to the hydrology field by some researchers (Sivapragasam *et al.* 2001; Marques *et al.* 2006; Partal & Kişi 2007). Sivapragasam *et al.* (2001) established a hybrid model of support vector machine (SVM) and the SSA for predictions of rainfall and runoff. A great improvement was achieved by the hybrid model in comparison with the original SVM model. The application of wavelet analysis to precipitation was conducted by Partal & Kişi (2007). Their results indicated that the wavelet analysis was highly promising.

The hybrid of ANN and SVR is less studied in hydrology. ANN and SVR are mathematically similar when they are used as approximators to map input/output pairs (Vapnik & Chervonenkis 1971; Vapnik 1995; Haykin 1999). Compared with SVR, ANN is very efficient in processing large-sized training samples when a local optimization technique, such as the Levernberg–Marquardt algorithm, is adopted for optimization of weights. In contrast, SVR has a good ability for generalization due to the adoption of structural risk minimization (SRM) for objective functions. As inspired by SVM, the structural risk minimization was also introduced to the training of ANN for the purpose of improving the model's generalization by Giustolisi & Laucelli (2004). Therefore, a hybrid method from ANN and SVR may be expected to provide a pronounced improvement in rainfall forecasting.

The scope of this study was to examine the joint effect of the hybrid model and the SSA on daily rainfall estimates. A conventional ANN was first of all combined with SSA to remove noises hidden in the original rainfall data. Filtered inputs were then accommodated into the hybrid model which consisted of three local models associated respectively with three crisp subsets (low-, medium- and high-intensity rainfall) clustered by the fuzzy C-means (FCM) method. To ensure wider applications, two different sizes of data from different regions of China were explored.

## METHODOLOGY

### Singular spectrum analysis (SSA)

SSA is able to decompose daily rainfall series into several additive components that typically can be interpreted as 'trend' components (which may not exist), various 'oscillatory' components, and 'noise' components (Golyandina *et al.* 2001). The noise and/or high-frequency oscillatory component may be filtered out when the SSA is used as the signal filter technique. The algorithm of the basic SSA can be referred to Vautard *et al.* (1992) and Golyandina *et al.* (2001). Following the methodology in Vautard *et al.* (1992), four steps are performed for the explanation of the SSA algorithm, depending on a univariate rainfall time series $\{x_1, x_2, \ldots, x_N\}$. 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. The 'trajectory matrix', denoted by $\boldsymbol{X}$, therefore reflects the evolution of the time series with a suitable choice of $\tau$ (delay or lagged time) and the window length $L$ (also called the singular number). The 'trajectory matrix' is denoted by

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

The next step is the singular value decomposition (SVD) of $\boldsymbol{X}$. Let $\boldsymbol{S} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ (called the lagged-covariance matrix). With SVD, $\boldsymbol{X}$ can be written as $\boldsymbol{X} = \boldsymbol{DLE}^{\mathrm{T}}$ where $\boldsymbol{D}$ and $\boldsymbol{E}$ are left and right singular vectors of $\boldsymbol{X}$, and $\boldsymbol{L}$ is a diagonal matrix of singular values. $\boldsymbol{E}$ consists of orthonormal columns and is also called the 'empirical orthonormal functions' (EOFs). Substituting $\boldsymbol{X}$ into the definition of $\boldsymbol{S}$ yields the formula $\boldsymbol{S} = \boldsymbol{EL}^2\boldsymbol{E}^{\mathrm{T}}$. Further, $\boldsymbol{S} = \boldsymbol{E} \wedge \boldsymbol{E}^{\mathrm{T}}$ since $\boldsymbol{L}^2 = \wedge$ where $\wedge$ is a diagonal matrix consisting of ordered values $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \lambda_L$. Therefore, the right singular vectors of $\boldsymbol{X}$ are the eigenvectors of $\boldsymbol{S}$. In other words, the singular vectors $\boldsymbol{E}$ and singular values of $\boldsymbol{X}$ can be

respectively attained by calculating the eigenvectors and the square roots of the eigenvalues of $\boldsymbol{S}$.

The third step is to calculate the principal components ($a_i^k$) by projecting the original time record onto the eigenvectors as follows:

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

where $e_j^k$ represents the $j$th component of the $k$th eigenvector. Each principal component is a filtered process of the original series with length $N - L + 1$.

The last step is to generate reconstruction components (RCs) with the length size being the same as the original series. The generation of each RC depends on a convolution of one principal component with the corresponding singular vector (Vautard *et al.* 1992). Therefore, the $L$ RCs can be achieved if all $L$ principal components and their associated eigenvectors are employed in the process of signal reconstruction. Certainly, the original record can be filtered by choosing $p(<L)$ RCs from all $L$ RCs.

## Hybrid model

### Artificial neural networks

The feed-forward multilayer perceptron (MLP) is by far the most popular among many ANN paradigms, which usually uses the technique of error back-propagation to train the network configuration. The architecture of the ANN consists of a number of hidden layers and the number of neurons in the input layer, hidden layers and output layer. ANNs with one hidden layer are commonly used in hydrologic modeling (Dawson & Wilby 2001; De Vos & Rientjes 2005) since these networks are considered to provide enough complexity to accurately simulate the nonlinear properties of the hydrologic process. A three-layer ANN is chosen for the current study, which comprises the input layer with $m$ nodes (i.e. $m$ past daily rainfall), the hidden layer with $h$ nodes (neurons) and the output layer with one node. The hyperbolic tangent functions are used as transfer functions in the hidden layer and the output layer. In terms of a rainfall time series $\{x_1, x_2, \ldots, x_N\}$, it can be reconstructed into a series of delay vectors of the type $\boldsymbol{Y}_t = \{x_t, x_{t+\tau}, x_{t+2\tau}, \ldots, x_{t+(m-1)\tau}\}$, $t = 1, 2, \ldots, n$ ($n = N - (m-1)\tau$), where $\boldsymbol{Y}_t \in R^m$, and $\tau$ was already mentioned above (the value of 1 was taken in the current study). The model architecture is described by the equation

$$x_{t+T}^F = f(Y_t, w, \theta, m, h) = \theta_0 + \sum_{j=1}^{h} w_j^{\text{out}} \tan h\left(\sum_{i=1}^{m} w_{ji} x_{t-i+1} + \theta_j\right) \tag{3}$$

where $x_{t-i+1}$, $i = 1, \ldots, m$ are the $m$ elements in the input vector $\boldsymbol{Y}_t$; $x_{t+T}^F$ is the single output which stands for the forecasted rainfall at the lead time $T$. $w_{ji}$ are the weights defining the link between the $i$th node of the input layer and the $j$th of the hidden layer; $\theta_j$ are biases associated with the $j$th node of the hidden layer; $w_j^{\text{out}}$ are the weights associated with the connection between the $j$th node of the hidden layer and the node of the output layer and $\theta_0$ is the bias at the output node. The Levernberg–Marquardt (LM) training algorithm is used to adjust the $w$ and $\theta$ because it is faster and less easily trapped in local minima compared with other local optimization methods such as gradient descent (Toth *et al.* 2000).

### Support vector regression (SVR)

SVR performs structural risk minimization (SRM) that aims at minimizing a bound on the generalization error (Gunn 1998; Kecman 2001). It creates a model with good generalization. The SVR can be divided into linear and nonlinear, depending on the kernel function being linear or nonlinear. A nonlinear SVR was used in this study. The underlying function $f(\boldsymbol{Y}_i)$ in the context of the nonlinear SVR is given by

$$x_{t+T}^F = f(\boldsymbol{Y}_t, \omega) = \omega \cdot \phi(\boldsymbol{Y}_t) + b \tag{4}$$

where the input vector $\boldsymbol{Y}_t$ in the input space is mapped to a high-dimensional feature space via a nonlinear mapping function $\phi(\boldsymbol{Y}_t)$. The objective of the SVR is to find optimal $\omega$, $b$ and some parameters in the kernel function $\phi(\boldsymbol{Y}_t)$ so as to construct an approximation function of the $f(\boldsymbol{Y}_i)$.

When introducing Vapnik's $\varepsilon$-insensitivity error (or loss function) and slack variables of $\xi_i$ and $\xi_i^*$ (which are respectively measurements 'above' and 'below' the $\varepsilon$ tube), the nonlinear SVR optimization problem becomes a dual problem and can be solved in a dual space (Kecman 2001).

Therefore, by introducing a dual set of Lagrange multipliers, $\alpha_i$ and $\alpha_i^*$, the objective function in dual form can be represented as (Gunn 1998)

$$\begin{aligned}&\text{maximize } L_d(\alpha, \alpha^*) = -\varepsilon\sum_{i=1}^{n}(\alpha_i^* + \alpha_i) \\ &+\sum_{i=1}^{n}(\alpha_i^* - \alpha_i)y_i - \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(\phi(Y_i)\cdot\phi(Y_j)) \\ &\text{subject to}\begin{cases}\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \\ 0 \le \alpha_i^* \le C, \quad i = 1, \ldots, n \\ 0 \le \alpha_i \le C, \quad i = 1, \ldots, n\end{cases}\end{aligned} \tag{5}$$

where $y_i$ represents the observed value, $\boldsymbol{Y}_i$ is the $\boldsymbol{Y}_t$ for simplicity and $C$ is a positive constant that determines the degree of penalized loss when a training error occurs. By using a 'kernel' function $K(\boldsymbol{Y}_i, \boldsymbol{Y}_j) = (\phi(\boldsymbol{Y}_i)\cdot\phi(\boldsymbol{Y}_j))$ to yield inner products in feature space, the computation in input space can be performed. In the present study, a Gaussian radial basis function (RBF) was adopted in the form of $K(\boldsymbol{Y}_i, \boldsymbol{Y}_j) = \exp(-\|Y_i - Y_j\|^2/2\sigma^2)$. Once parameters $\alpha_i$, $\alpha_i^*$ and $b_0$ are obtained, the final approximation function of the $f(\boldsymbol{Y}_i)$ becomes

$$f(\boldsymbol{Y}_i) = \sum_{i=1}^{n}\left(\alpha_k - \alpha_k^*\right)K(Y_k\cdot Y_i) + b_0, \quad k = 1, \ldots, s \tag{6}$$

where $\boldsymbol{Y}_k$ stands for the support vector, $\alpha_k$ and $\alpha_k^*$ are parameters associated with support vector $\boldsymbol{Y}_k$, and $n$ and $s$ represent the number of training samples and support vectors, respectively. Three parameters ($C$, $\varepsilon$, $\sigma$) need to be optimized in order to identify the optimal $f(\boldsymbol{Y}_i)$ in Equation (5). In the current study, a two-step genetic algorithm method was used for their optimizations (Wu *et al.* 2008).

## Fuzzy C-means (FCM)

The FCM method (Bezdek 1981) partitions a set of $N$ vector $\boldsymbol{Y}_j$, $j = 1, \ldots, n$, 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 $\boldsymbol{U}$ comprising the elements $u_{ij}$ and assume that the summation of the degrees of belonging for a data point is equal to 1, i.e. $\sum_{i=1}^{c} u_{ij} = 1\ \forall j = 1, \ldots, n$. The goal of the FCM algorithm is to find $c$ cluster centers so that the cost function of the dissimilarity measure is minimized. The cost function can be defined by

$$J(\boldsymbol{U}, \nu_1, \ldots, \nu_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^l d_{ij}^2 \tag{7}$$

where $\nu_i$ is the cluster center of the fuzzy subset $i$; $d_{ij} = \|\nu_i - \boldsymbol{Y}_j\|$ is the Euclidean distance between the $i$th cluster center and $j$th data point; and $l \ge 1$ is a weighting exponent, taken as 2 here so as to match the square Euclidean distance. The necessary conditions for Equation (7) to reach its minimum are

$$\nu_i = \frac{\sum_{j=1}^{n} u_{ij}^l Y_j}{\sum_{j=1}^{n} u_{ij}^l} \tag{8}$$

$$u_{ij} = \left[\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{2/(l-1)}\right]^{-1} \tag{9}$$

The FCM algorithm is an iterative procedure that satisfies Equations (8) and (9) to minimize Equation (7). Implementation of the algorithm can be referred to Bezdek (1981) and Wang *et al.* (2006) for details.

The number $c$ of clusters was taken to be 3 in this study. Three crisp sets were therefore obtained when only the biggest weight was taken for each subset, although FCM inherently conducts a fuzzy division. The three subsets may represent three types of rainfalls, i.e. low-intensity (or zero) rainfall, medium-intensity rainfall and high-intensity rainfall (storm events).

## Hybrid model

The hybrid model was based on the crisp data split. A basic idea behind the hybrid model was that the training set was first split into three subsets by the fuzzy $c$-means (FCM) technique, and then each subset was approximated by ANN (or SVR). They were associated with subset 1, subset 2 and subset 3 in Figure 1. The final output of the hybrid model was obtained directly from the output of one of the three local models.

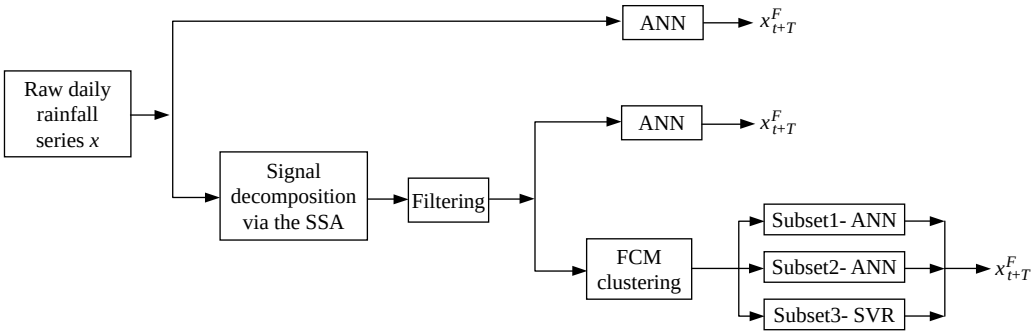


**Figure 1** | Implementation framework of forecasting models.

## Combination with the SSA

The ANN and the hybrid model were coupled with the SSA (depicted in Figure 1). They were hereafter referred to as ANN-SSA and ANN-SVR-SSA. The methodological procedures of this combination were summarized into three steps: SSA decomposition, correlation coefficients sort and reconstructed components filter. The operation in each step is bounded by the dashed box in Figure 2.

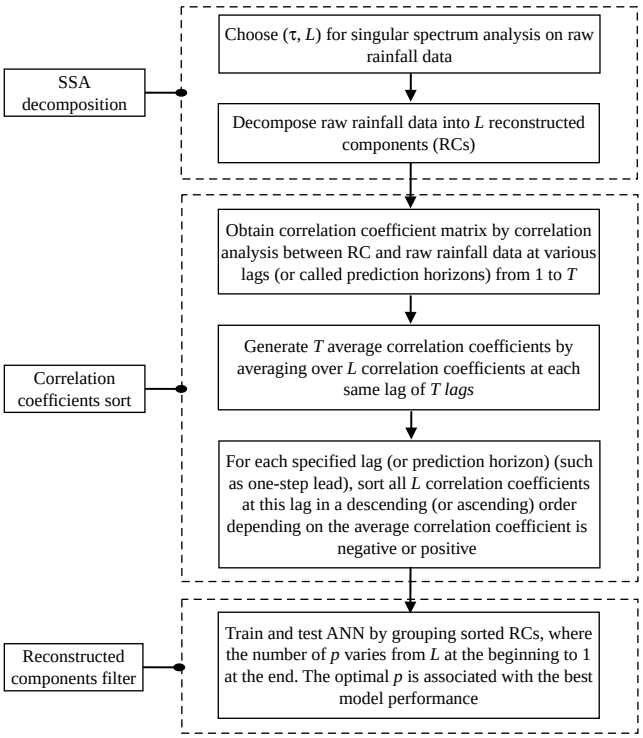


**Figure 2** | Recommended procedure for prediction models with SSA.

## Evaluation of model performances

As suggested by Legates & McCabe (1999), a comprehensive assessment of model performance at least includes absolute error measure and relative error measure. Therefore, the measures of model performance evaluation comprise the Root Mean Square Error (RMSE) and the Nash–Sutcliffe Coefficient of Efficiency (CE). They are respectively formulated as

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{10}$$

$$\mathrm{CE} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{11}$$

where $n$ is the number of observations; $\hat{y}_i$ stands for the forecasted rainfall; $y_i$ is the observed rainfall; $\bar{y}$ denotes the average observed rainfall and $y_{i-T}$ is the rainfall estimate from the persistence model. The value of 1 for CE stands for a perfect fit.

# CASE STUDY

Daily rainfall series of two watersheds, Zhenshui and Da'ninghe, were analyzed in this study.

The Da'ninghe basin, a first-order tributary of the Yangtze River, is located in the northwest of Hubei Province. The daily rainfall data from 1 January 1988 to 31 December 2007 were measured at six rain gauges located at the upstream of the study basin. The upstream part of the Da'ninghe basin is controlled by the Wuxi hydrology station, with a drainage area of around 2,000 km$^2$. The rainfall was spatially averaged with the Thiessen polygon

method (hereafter the averaged rainfall series was referred to as Wuxi).

The Zhenshui basin is located in the north of Guangdong Province and is adjoined by Hunan Province and Jianxi Province. The basin belongs to a second-order tributary of the Pearl River and has an area of 7,554 km$^2$. The daily rainfall time series data of the Zhenwan rain gauge was collected between 1 January 1989 and 31 December 1998 (hereafter the rainfall series was referred to as Zhenwan).

Figure 3 shows the original rainfall processes of Wuxi and Zhenwan. Each of them was partitioned into three parts as a training set, cross-validation set and testing set. The training set served for model training and the testing set was used to evaluate the performances of models. The cross-validation set functioned for dual purposes: one was to implement an early stopping approach in order to avoid overfitting of the training data and another was to select some best predictions from the many ANN runs. In the present study, the 10 best predictions were selected from a total of 20 ANN runs. The same data partition was adopted in two rainfall series: the first half of the entire data was the training set and the first half of the remaining data was cross-validation set with the other half as the testing set.

Table 1 presents related information about watersheds and some descriptive statistics of the original data and three data subsets, including mean ($\mu$), standard deviation ($S_x$), coefficient of variation ($C_v$), skewness coefficient ($C_s$), minimum ($X_{\min}$) and maximum ($X_{\max}$). As shown in Table 1, the training set cannot fully include the cross-validation for Wuxi. Due to the weak extrapolation ability of ANN, it is suggested that all data are scaled to the interval $[-0.9, 0.9]$ when ANN employs hyperbolic tangent functions as transfer functions in the hidden layer and output layer.

## APPLICATIONS

### Decomposition of rainfall data

The decomposition of the daily average rainfall series required identifying the window length $L$ (or the singular number). In principle, the value of $L$ should be able to clearly resolve different oscillations hidden in the original signal. The present application did not need to precisely resolve potential oscillations in the raw rainfall signal. A rough resolution can be adequate for the separation of oscillations and noises. Therefore, a small $L$ was

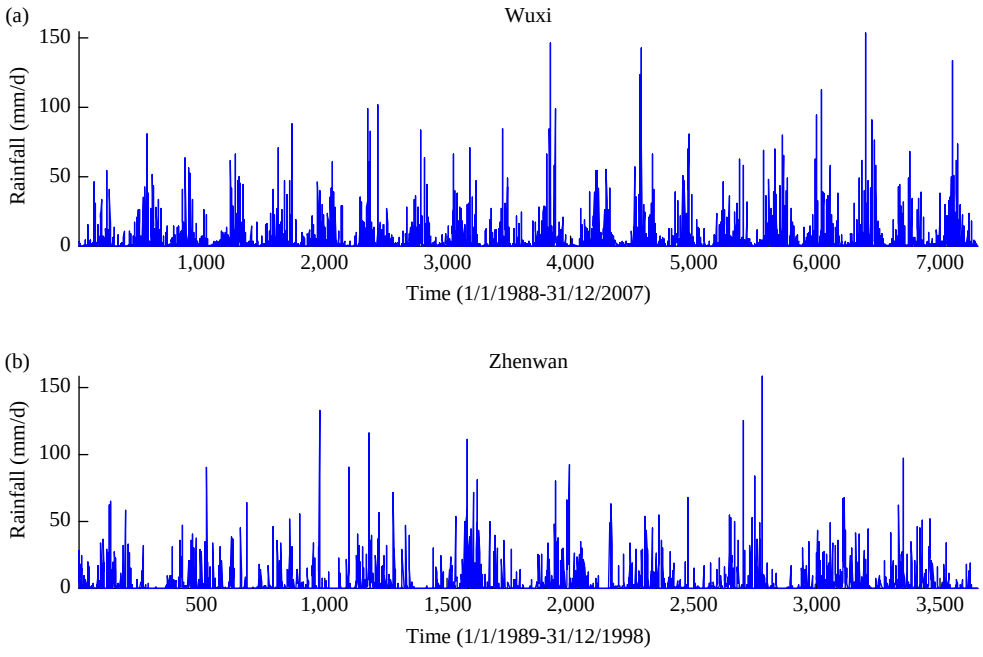


**Figure 3** | Daily rainfall processes: (a) Wuxi and (b) Zhenwan.

**Table 1** | Related information for two watersheds and the rainfall data

| | Statistical parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| **Watershed and datasets** | **$\mu$ (mm)** | **$S_x$ (mm)** | **$C_v$** | **$C_s$** | **$X_{min}$ (mm)** | **$X_{max}$ (mm)** | **Watershed area and data period** |
| Wuxi | | | | | | | |
| Original data | 3.67 | 10.15 | 0.36 | 5.68 | 0.00 | 154 | Area: 2,000 km$^2$ |
| Training | 3.81 | 10.94 | 0.35 | 6.27 | 0.00 | 147 | |
| Cross-validation | 3.42 | 8.87 | 0.39 | 4.96 | 0.00 | 102 | Data period: Jan., 1988–Dec., 2007 |
| Testing | 4.03 | 11.60 | 0.35 | 5.46 | 0.00 | 154 | |
| Zhenwan | | | | | | | |
| Original data | 4.3 | 11.0 | 0.39 | 4.94 | 0.0 | 159 | Area: 7,554 km$^2$ |
| Training | 4.3 | 11.2 | 0.38 | 5.60 | 0.0 | 159 | |
| Cross-validation | 4.7 | 11.2 | 0.42 | 4.22 | 0.0 | 125 | Data period: Jan., 1989–Dec., 1998 |
| Testing | 4.0 | 10.9 | 0.37 | 4.97 | 0.0 | 133 | |

empirically chosen for the two rainfall series. Figure 4 displays the singular spectrum as a function of lag using various window lengths $L$ for Zhenwan and Wuxi. Results showed that the $L$ value for Zhenwan was more sensitive than Wuxi because the small singular values in the former became indistinguishable when $L$ was larger than 7. Therefore, $L$ was set to a value of 7 for Zhenwan. For the convenience of the filtering operation, $L$ was chosen as a small value of 5 for Wuxi. Finally, 5 RCs and 7 RCs were respectively employed for Wuxi and Zhenwan in the SSA.

## Identification of the ANN architecture

The architecture identification included determining model inputs and the number of nodes (or neurons) in the hidden layer when there was one model output. The statistical approach of examining auto- and partial-auto-correlation of the observed time series was recognized as a good and parsimonious method in identifying model inputs (Sudheer *et al.* 2002; Kişi 2008). The model input in this method is mainly determined by the plot of the partial-auto-correlation function (PACF). The essence of this method is to examine the linear dependence between the

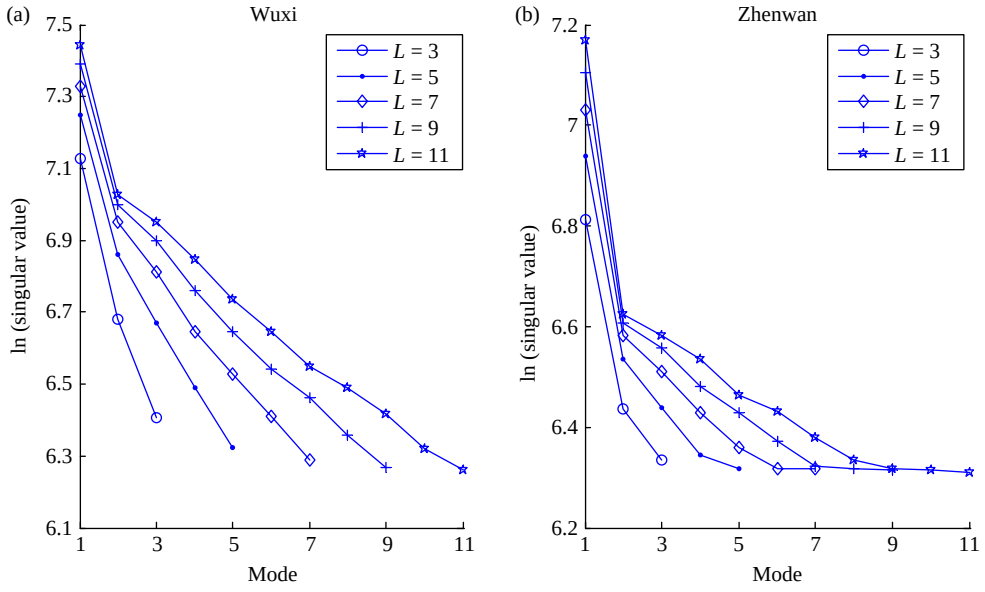


**Figure 4** | Singular spectrum for (a) Wuxi and (b) Zhenwan as a function of lag using various window lengths $L$.

input and output data series. According to this method, the model inputs were originally considered to take the previous five daily rainfalls for Wuxi and the previous seven daily rainfalls for Zhenwan because the PACF value decayed within the confidence band around at lag 5 for Wuxi and lag 7 for Wuxi (see Figure 5).

The ensuing task was to optimize the size of the hidden layer with the determined inputs and one output. The optimal size $h$ of the hidden layer was found by systematically increasing the number of hidden neurons from 1 to 10, where the training data was rescaled to $[-0.9, 0.9]$ due to the use of a hyperbolic tangent function as the transfer function. The identified ANN architecture was 5-4-1 for Wuxi and 7-4-1 for Zhenwan.

It is worthwhile mentioning that the standardization of the training data is a very crucial factor in the improvement of model performance. Two common standardization methods of rescaling and normalization can be found in the literature (Dawson & Wilby 2001; Cannas *et al.* 2002; Rajurkar *et al.* 2002; Campolo *et al.* 2003; Wang *et al.* 2006). The rescaling method, as adopted above, is to rescale the training data to $[-1, 1]$ or $[0, 1]$ or even more narrow interval, depending on what kinds of transfer functions are employed in ANN. The normalization method is to rescale the training data to a Gaussian function with a mean of 0 and unit standard deviation, which is by subtracting the mean and dividing by the standard deviation. When the normalization approach was adopted, we used the linear transfer function (e.g. purelin) instead of the hyperbolic tangent function in the output layer. In addition, some studies have indicated that considerations of statistical principles may improve ANN model performance (Cheng & Titterington 1994; Sarle 1994). For example, the training data was recommended to be normally distributed (Fortin *et al.* 1997). Sudheer *et al.* (2002) suggested that the issue of stationarity should be considered in the ANN development because the ANN cannot account for trends and heteroscedasticity in the data. Their results showed that data transformation to reduce the skewness of data can significantly improve the model performance. For the purpose of obtaining better model performance, four data-transformed schemes were examined:

- Rescaling the raw data (referred to as Resc_raw);
- Normalizing the raw data (referred to as Norm_raw);
- Rescaling the $n$th root transformed data (referred to as Resc_nth_root);
- Normalizing the $n$th root transformed data (referred to as Norm_nth_root).

Table 2 compares the ANN model performance in terms of RMSE and CE among the four schemes. The normalization to the raw data scheme was more effective than the

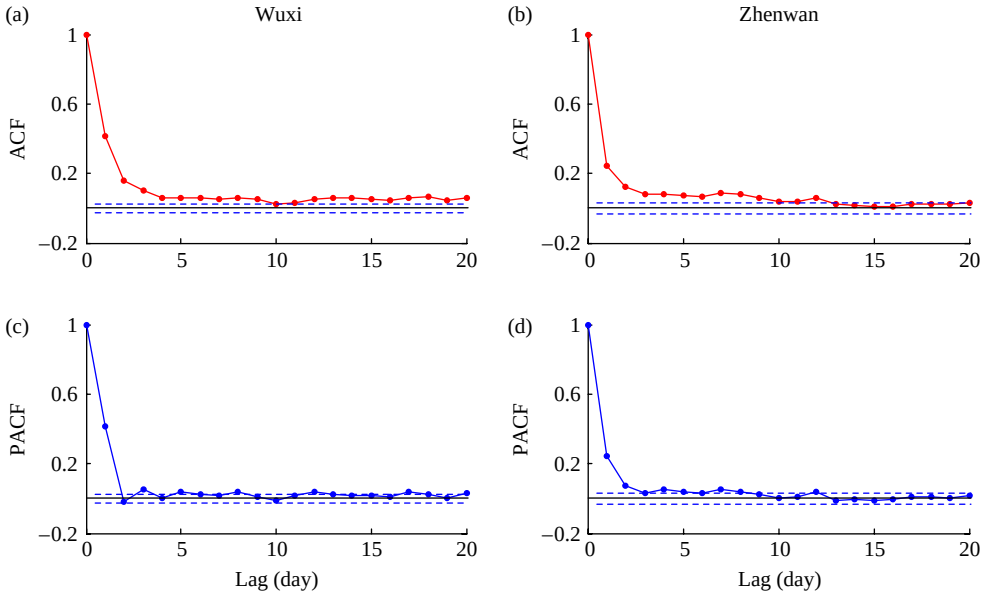


**Figure 5** | Plots of ACF and PACF of the rainfall series with the 95% confidence bounds (dashed lines), (a) and (c) for Wuxi, and (b) and (d) for Zhenwan.

**Table 2** | Performance comparison of the ANN model with different data-transformed methods

| | | RMSE | | | CE | | |
|---|---|---|---|---|---|---|---|
| **Watershed** | **Data transformation** | **1** | **2** | **3*** | **1** | **2** | **3** |
| Wuxi | | | | | | | |
| | Resc_raw | 10.77 | 11.54 | 11.62† | 0.14 | 0.02 | 0.00 |
| | Norm_raw | 10.57 | 11.49 | 11.59 | 0.17 | 0.01 | 0.00 |
| | Resc_nth_root | 11.00 | 12.02 | 12.10 | 0.10 | −0.07 | −0.09 |
| | Norm_nth_root | 11.15 | 12.01 | 12.09 | 0.08 | −0.07 | −0.09 |
| Zhenwan | | | | | | | |
| | Resc_raw | 11.03 | 11.11 | 11.16 | 0.03 | 0.02 | 0.01 |
| | Norm_raw | 10.72 | 11.06 | 11.14 | 0.09 | 0.03 | 0.02 |
| | Resc_nth_root | 11.25 | 11.68 | 11.75 | −0.01 | −0.09 | −0.10 |
| | Norm_nth_root | 11.34 | 11.70 | 11.74 | −0.02 | −0.09 | −0.09 |

*Numbers '1, 2, and 3' denote one-, two-, and three-day-ahead forecasts.
†Result is average over 10 best runs from total 20 runs.

rescaling method. It can also be seen that using the root of *n*th degree (taking 3 by trial and error) as data transformation was ineffective for the improvement of performance. Therefore, the Norm_raw scheme was adopted for the later rainfall prediction in the present study.

## Main results

Figure 6 shows the results of filtering the RCs at the one-, two-, and three-day-ahead prediction horizons using the conventional ANN model. The RMSE associated with the maximal $p$ showed the performance of the conventional ANN. The numbers of chosen optimal $p$ RCs in three forecasting horizons were respectively 3, 2 and 1 for Wuxi and 4, 3 and 2 for Zhenwan. In terms of the RMSE, the SSA substantially improved the conventional ANN performance. A more detailed description is shown in Table 3.

Table 3 demonstrates forecasting results from three models at various prediction horizons in terms of RMSE and CE. It can be seen that the benchmark model ANN

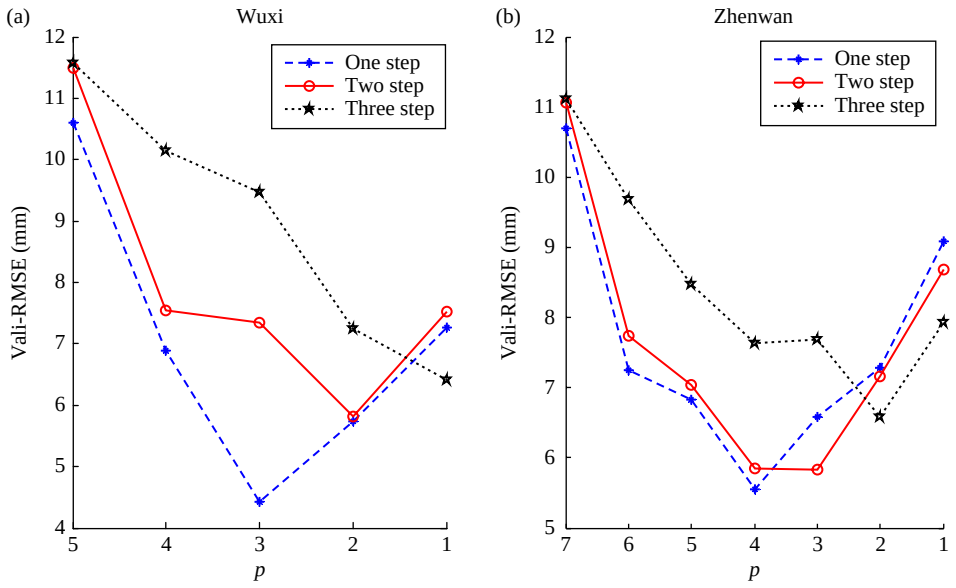


**Figure 6** | Performance of ANN-SSA in terms of RMSE as a function of $p$ ($\leq L$) selected RCs at various prediction horizons: (a) Wuxi and (b) Zhenwan.

**Table 3** | Comparison of three models' performances at various forecasting horizons

| | | RMSE | | | CE | | | *p* from *L* RCs | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Watershed** | **Model** | **1** | **2** | **3** | **1** | **2** | **3** | **1** | **2** | **3** |
| Wuxi | | | | | | | | | | |
| | ANN | 10.59 | 11.50 | 11.59 | 0.17 | 0.02 | 0.00 | | | |
| | ANN-SSA | 4.66 | 5.41 | 6.35 | 0.84 | 0.78 | 0.70 | 3/5* | 2/5 | 1/5 |
| | ANN-SVR-SSA | 4.18 | 3.48 | 4.14 | 0.87 | 0.91 | 0.87 | 3/5 | 2/5 | 1/5 |
| Zhenwan | | | | | | | | | | |
| | ANN | 10.68 | 11.05 | 11.12 | 0.09 | 0.03 | 0.02 | | | |
| | ANN-SSA | 4.94 | 5.60 | 5.89 | 0.81 | 0.75 | 0.71 | 4/7 | 3/7 | 2/7 |
| | ANN-SVR-SSA | 3.18 | 3.20 | 3.31 | 0.92 | 0.92 | 0.91 | 4/7 | 3/7 | 2/7 |

*Denominator stands for all RCs and numerator represents the number of selected RCs.

presented very poor performances for each watershed. The performance indices from ANN-SSA and ANN-SVR-SSA indicate that the SSA gave rise to a considerable improvement in the accuracy of the rainfall forecasting. The hybrid model among all models performed best for each rainfall series. It is worthwhile to note that the values of selected *p* (the number of effective RCs) were different at the three forecasting horizons.

One-day-ahead estimates of ANN, ANN-SSA and ANN-SVR-SSA models are shown in Figures 7 (Wuxi) and 8 (Zhenwan) in the form of hyetographs and scatter plots (the former was plotted in a selected range for better

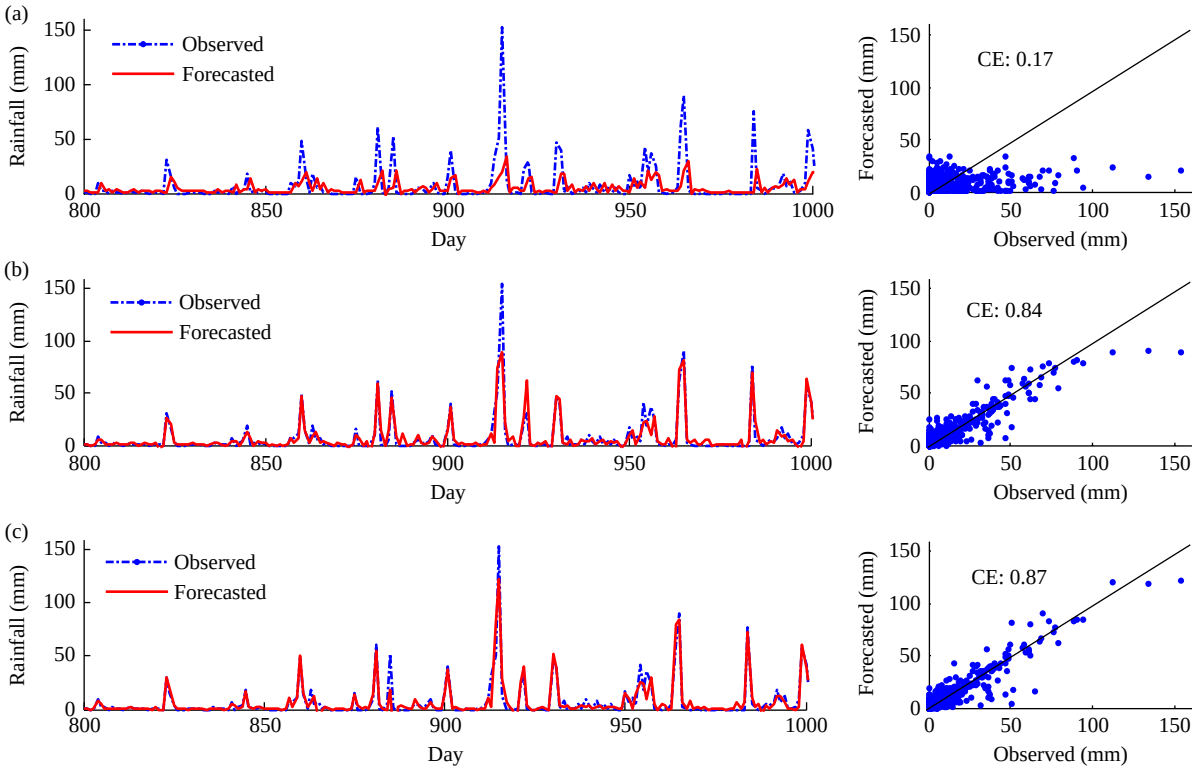


**Figure 7** | One-day-ahead estimates of the Wuxi rainfall using models of (a) ANN, (b) ANN-SSA and (c) ANN-SVR-SSA.

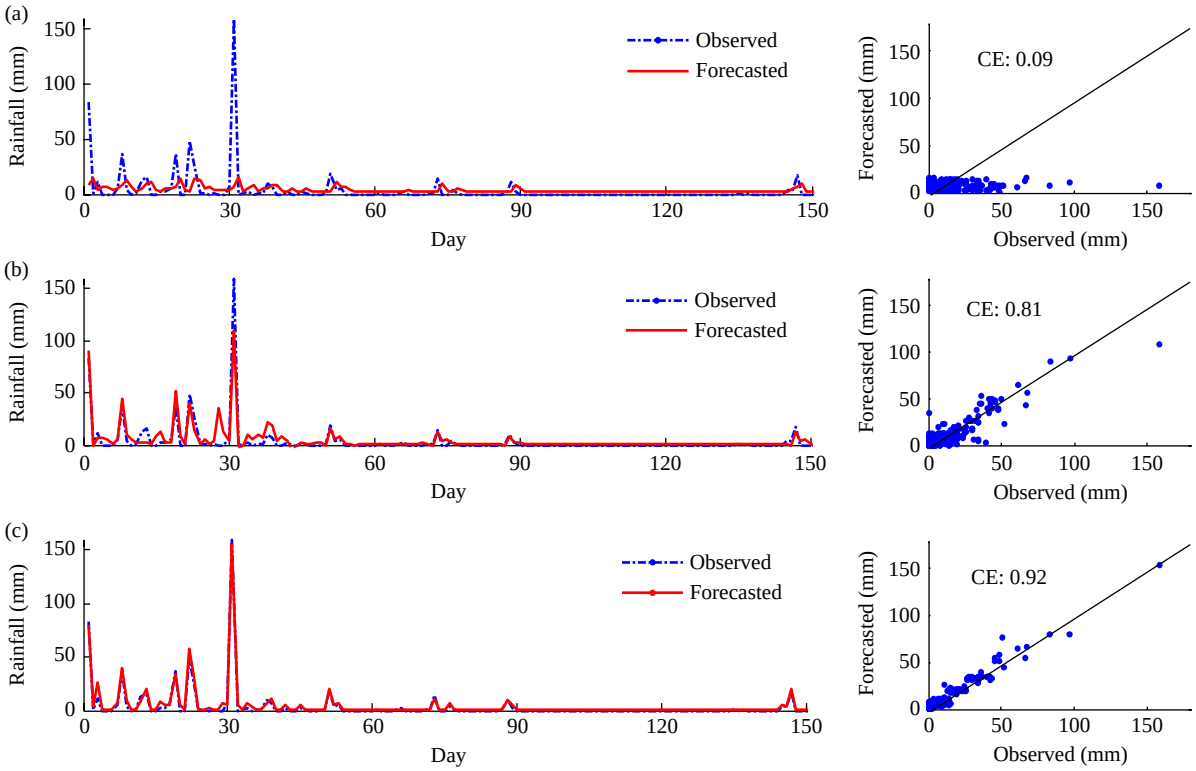


**Figure 8** | One-day-ahead estimates of the Zhenwan rainfall using models of (a) ANN, (b) ANN-SSA and (c) ANN-SVR-SSA.

visual inspection). As seen from the hyetograph graphs, the ANN-SVR-SSA reproduced the corresponding observed rainfall data better than the other models. It can also be seen from the scatter plots that the ANN-SVR-SSA predictions were much closer to the exact fitting line than those of the ANN and ANN-SSA models. The scatter plots' comparison between ANN-SVR-SSA and ANN-SSA models indicates that the three local models are able to better approximate different rainfall characteristics than a global model because the high intensity rainfall was well simulated by the ANN-SVR-SSA. Compared with the ANN, the ANN-SSA also exhibited an acceptable ability to forecast daily rainfall.

The poor forecasts from the ANN imply that the ANN fed by the original rainfall data is not viable, at least in the present cases. Actually, the ANN mainly captured the zero or low-intensity rainfall patterns (dry periods) because the type of pattern was dominant when using the original rainfall series to construct model input/output pairs. The SSA filter on the raw rainfall series eliminated those patterns so that the ANN was able to capture medium or large rainfall patterns.

## DISCUSSION

Some discussions about ANN architecture and the singular spectrum analysis are made as follows.

### Architecture of ANN

As is known, the development of the ANN fully depends on the data employed. Subjective factors are unavoidably involved in almost every aspect of the ANN identification including determination of model inputs, selection of network type and so on. In terms of the network type, the most widely-used network in hydrologic modeling is the feed-forward three-layer perceptron since this type of network is considered to provide enough complexity to accurately simulate the nonlinear properties of the

hydrologic process (Dawson & Wilby 2001; De Vos & Rientjes 2005). As mentioned earlier, the identification of a three-layer ANN requires choosing both the model inputs and the number of nodes in the hidden layer. Optimal network geometries have traditionally been found by trial and error. Other usual approaches include regularization, pruning, constructive and stopped training (Anders & Korn 1999). Recently, genetic algorithms have been used as automatic techniques to determine optimal network architecture (Abrahart *et al.* 1999; Giustolisi & Laucelli 2005; Giustolisi & Simeone 2006). To justify the identified ANN in this study, five trial-and-error methods and one multi-objective genetic algorithm (referred to as MOGA-ANN by Giustolisi & Simeone (2006)) were examined. Each of the five trial-and-error methods consists of two steps: the first step is to identify model inputs and then to find the optimal number of hidden neurons. As a matter of fact, the five methods denote different techniques to determine model inputs. They are correlation analysis (as used in this study), false nearest neighbors (FNN) (Kennel *et al.* 1992; Abarbanel *et al.* 1993), stepwise linear regression, average mutual information (AMI) (De Vos & Rientjes 2007) and partial mutual information (PMI) (May *et al.* 2008). In contrast, the MOGA-ANN is able to automatically obtain the optimal network geometry by using genetic algorithms to search appropriate inputs and the number of hidden neurons simultaneously. For the purpose of comparison, the maximum number of inputs and nodes of the hidden layer were respectively 15 and 10 for each ANN model. Table 4 demonstrates performances of one-step-ahead predictions in terms of RMSE and CE. There are no significant differences among the performances of all models. Regarding the five trail-and-error methods, it can be found that the simple correlation analysis performed best but marginally. The method of MOGA was slightly better than the correlation analysis technique in Zhenwan but worse in Wuxi. Therefore, the correlation analysis method seems tenable for the current two rainfall series.

**Table 4** | Comparison of methods to identify ANN architecture

| Watershed | Methods | Performance | | Structure of ANN |
|---|---|---|---|---|
| | | RMSE | CE | |
| Wuxi | | | | |
| | Correlation analysis | 10.58 | 0.17 | (5-4-1) |
| | FNN | 10.62 | 0.16 | (14-3-1) |
| | Stepwise | 10.64 | 0.16 | (6-3-1) |
| | AMI | 10.73 | 0.15 | (10-3-1) |
| | PMI | 10.74 | 0.14 | (7-8-1) |
| | MOGA | 10.63 | 0.16 | (2-6-1) |
| Zhenwan | | | | |
| | Correlation analysis | 10.70 | 0.09 | (7-4-1) |
| | FNN | 10.76 | 0.08 | (14-3-1) |
| | Stepwise | 10.75 | 0.08 | (6-3-1) |
| | AMI | 10.77 | 0.08 | (7-5-1) |
| | PMI | 10.80 | 0.07 | (10-4-1) |
| | MOGA | 10.43 | 0.14 | (5-8-1) |

## Parameter p in SSA

Filtering is a key step when the raw rainfall series was decomposed by SSA. In principle, each effective reconstructed component should be characterized by significant correlation with the raw data, which underlies the procedure in Figure 2. For convenience of comparison, the filtering method in Figure 2 was called supervised filtering. Potentially, there can be some drawbacks in supervised filtering. The salient point is that only linear correlation analysis is considered, which disregards the existence of nonlinearity in hydrologic processes. In the meantime, random combinations among all reconstructed components were not considered. To overcome these drawbacks, an unsupervised filtering method (also called enumeration) was recommended where all input combinations were examined. As is known, ANN tends to generate unstable outputs due to the randomization of initial weights. The nearest-neighbor method (NNM) (Yakowitz 1987) was therefore employed as the baseline model instead of ANN. Effective RCs from the supervised filtering and unsupervised filtering methods are described in Table 5. The baseline model's performance in terms of RMSE was somewhat improved with the adoption of unsupervised filtering. Moreover, the effective RCs from the two filtering methods are also different. It is suggested that the enumeration method may more suitable in the search of effective RCs of SSA.

**Table 5** | Comparison of two filtering methods to obtain $p$ effective RCs

| | | Supervised filtering | | | Unsupervised filtering | | |
|---|---|---|---|---|---|---|---|
| **Watershed** | **Prediction horizon** | **RMSE** | **Effective RCs** | ***p*** | **RMSE** | **Effective RCs** | ***p*** |
| Wuxi | | | | | | | |
| | 1 | 8.02 | RC1* | 1 | 7.17 | RC2, RC3 | 2 |
| | 2 | 8.41 | RC1 | 1 | 8.03 | RC2, RC4 | 2 |
| | 3 | 9.99 | RC1 | 1 | 9.69 | RC2 | 1 |
| Zhenwan | | | | | | | |
| | 1 | 9.84 | RC1, RC2, RC3, RC4, RC5 | 5 | 7.64 | RC3, RC6 | 2 |
| | 2 | 9.72 | RC1 | 1 | 8.95 | RC3, RC6 | 2 |
| | 3 | 10.33 | RC1 | 1 | 10.24 | RC2, RC5 | 2 |

*RC1 stands for the first reconstructed component which is associated with the maximum singular value.

## CONCLUSIONS

The purpose of this study was to investigate the joint effect of a hybrid ANN-SVR model and the singular spectrum analysis on improving the accuracy of daily rainfall forecasting. Two daily rainfall records from different locations in the People's Republic of China were used as testing cases. A conventional ANN was used as the benchmark.

The poor forecasts from the ANN implied that the ANN fed by the original rainfall data is not viable in the present cases. Actually, the ANN mainly captured the zero or low-intensity rainfall patterns (dry periods) because the type of pattern was dominant when using the original rainfall series to construct model input/output pairs. Comparison between ANN and ANN-SSA showed that the SSA filtering technique considerably improved the ANN performance, which mainly was because the SSA eliminated the zero values in model inputs to strengthen the mapping relationship between inputs and output. When the SSA was coupled with the hybrid ANN-SVR model, the values of the peak of rainfall can better be estimated than those in the ANN-SSA. The result indicates that the three local models can better approximate different rainfall characteristics than a single global model.

In addition, two filtering techniques for determining effective reconstructed components in SSA, supervised (correlation analysis) and unsupervised (enumeration), were evaluated. It was found that enumeration tends to be more effective than correlation analysis. As a matter of fact, the former can consider the nonlinear dependence between model inputs and output whereas the latter only analyzes the linear dependence between them.

In the present study, the selection of the window length $L$ is empirical, which unavoidably introduces uncertainty to rainfall estimates. A future work should, therefore, pursue more rigorous techniques for the determination of the window length in SSA.

## REFERENCES

Abarbanel, H. D. I., Brown, R., Sidorowich, J. J. & Tsimring, L. S. 1993 The analysis of observed chaotic data in physical systems. *Rev. Mod. Phys.* **65** (4), 1331–1392.

Abrahart, R. J., See, L. & Kneale, P. E. 1999 Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model. *J. Hydroinformatics* **1** (2), 103–114.

Anders, U. & Korn, O. 1999 Model selection in neural networks. *Neural Networks* **12**, 309–323.

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

Brath, A., Montanari, A. & Toth, E. 2002 Neural networks and non-parametric methods for improving real time flood forecasting through conceptual hydrological models. *Hydrol. Earth Syst. Sci.* **6** (4), 627–640.

Bray, M. & Han, D. 2004 Identification of support vector machines for runoff modeling. *J. Hydroinf.* **6** (4), 265–280.

Campolo, M., Andreussi, P. & Soldati, A. 2003 Artificial neural network approach to flood forecasting in the river Arno. *Hydrol. Sci. J.* **48** (3), 381–398.

Cannas, B., Fanni, A., Pintus, M. & Sechi, G. M. 2002 Neural network models to forecast hydrological risk. *Proceedings of the International Joint Conference on Neural Networks* 1, pp. 423–426.

Chattopadhyay, S. & Chattopadhyay, G. 2007 Identification of the best hidden layer size for three layered neural net in predicting monsoon rainfall in India. *J. Hydroinformatics* **10** (2), 181–188.

Chattopadhyay, S. & Chattopadhyay, G. 2008 Identification of the best hidden layer size for three-layered neural net in predicting monsoon rainfall in India. *J. Hydroinformatics* **10** (2), 181–188.

Chen, J. C., Shu, C. S., Ning, S. K. & Chen, H. W. 2008 Flooding probability of urban area estimated by decision tree and artificial neural networks. *J. Hydroinformatics* **10** (1), 57–67.

Cheng, B. & Titterington, D. M. 1994 Neural networks: a review from a statistical perspective. *Stat. Sci.* **9** (1), 2–54.

Dawson, C. W. & Wilby, R. L. 2001 Hydrological modeling using artificial neural networks. *Prog. Phys. Geogr.* **25** (1), 80–108.

De Vos, N. J. & Rientjes, T. H. M. 2005 Constraints of artificial neural networks for rainfall-runoff modeling: trade-offs in hydrological state representation and model evaluation. *Hydrol. Earth Syst. Sci.* **9**, 111–126.

De Vos, N. J. & Rientjes, T. H. M. 2007 Multi-objective performance comparison of an artificial neural network and a conceptual rainfall-runoff model. *Hydrol. Sci. J.-Journal Des Sciences Hydrologiques* **52** (3), 397–413.

Fortin, V., Quarda, T. B. M. J. & Bobee, B. 1997 Comments on 'The use of artificial neural networks for the prediction of water quality parameters'. by Maier, H.R. and Dandy, G.C. *Water Resour. Res.* **33** (10), 2423–2424.

Giustolisi, O. & Laucelli, D. 2004 A new method to train MLPs as SVMs. In: *Proc. VI International Conference on Hydroinformatics HIC 2004, Singapore, 21–24 June* World Scientific, Singapore, Vol. 2, pp. 1605–1612.

Giustolisi, O. & Laucelli, D. 2005 Improving generalization of artificial neural networks in rainfall-runoff modeling. *Hydrol. Sci. J.* **50** (3), 439–457.

Giustolisi, O. & Simeone, V. 2006 Optimal design of artificial neural networks by multi-objective strategy: groundwater level predictions. *Hydrol. Sci. J.* **51** (3), 502–523.

Golyandina, N., Nekrutkin, V. & Zhigljavsky, A. 2001 *Analysis of Time Series Structure: SSA and the Related Techniques*. Chapman & Hall/CRC, Boca Raton, FL.

Guhathakurta, P. 2008 Long lead monsoon rainfall prediction for meteorological sub-divisions of India using deterministic artificial neural network model. *Meteorol. Atmos. Phys.* **101** (1–2), 93–108.

Gunn, S. R. 1998 *Support Vector Machines for Classification and Regression*. Image, Speech and Intelligent Systems Tech. Rep. University of Southampton.

Han, D., Chan, L. & Zhu, N. 2007 Flood forecasting using support vector machines. *J. Hydroinformatics* **9** (4), 267–276.

Haykin, S. 1999 *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice-Hall, Englewood Cliffs, NJ.

Hettiarachchi, P., Hall, M. J. & Minns, A. W. 2005 The extrapolation of artificial neural networks for the modelling of rainfall-runoff relationships. *J. Hydroinformatics.* **7** (4), 291–296.

Hu, T. S., Lam, K. C. & Thomas, N. G. 2001 River flow time series prediction with range-dependent neural network. *Hydrol. Sci. J.* **46** (5), 729–745.

Kecman, V. 2001 *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. MIT Press, Cambridge, MA.

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

Kişi, O. 2008 Constructing neural network sediment estimation models using a data-driven algorithm. *Math. Comput. Simul.* **79**, 94–103.

Legates, D. R. & McCabe, G. J., Jr. 1999 Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water Resour. Res.* **35** (1), 233–241.

Marques, C. A. F., Ferreira, J., Rocha, A., Castanheira, J., Gonçalves, P., Vaz, N. & Dias, J. M. 2006 Singular spectrum analysis and forecasting of hydrological time series. *Phys. Chem. Earth* **31**, 1172–1179.

May, R. J., Maier, H. R., Dandy, G. C. & Fernando, T. M. K. 2008 Non-linear variable selection for artificial neural networks using partial mutual information. *Environ. Model. Softw.* **23**, 1312–1328.

Partal, T. & Kişi, O. 2007 Wavelet and neuro-fuzzy conjunction model for precipitation forecasting. *J. Hydrol.* **342** (2), 199–212.

Pongracz, R., Bartholy, J. & Bogardi, I. 2001 Fuzzy rule-based prediction of monthly precipitation. *Phys. Chem. Earth Part B-Hydrol. Oceans Atmos.* **26** (9), 663–667.

Rajurkar, M. P., Kothyari, U. C. & Chaube, U. C. 2002 Artificial neural network for daily rainfall-runoff modeling. *Hydrol. Sci. J.* **47** (6), 865–877.

Sarle, W. S. 1994 Neural networks and statistical models. In: *Proc. of the Nineteenth Annual SAS Users Group International Conference*. SAS Institute, Cary, NC, pp. 1538–1550.

See, L. & Openshaw, S. 2000 A hybrid multi-model approach to river level forecasting. *Hydrol. Sci. J.* **45** (3), 523–536.

Silverman, D. & Dracup, J. A. 2000 Artificial neural networks and long-range precipitation prediction in California. *J. Appl. Meteorol.* **39** (1), 57–66.

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

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

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

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

Sudheer, K. P., Gosain, A. K. & Ramasastri, K. S. 2002 A data-driven algorithm for constructing artificial neural network rainfall-runoff models. *Hydrol. Process.* **16**, 1325–1330.

Toth, E., Brath, A. & Montanari, A. 2000 Comparison of short-term rainfall prediction models for real-time flood forecasting. *J. Hydrol.* **239** (1–4), 132–147.

Vapnik, V. N. 1995 *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

Vapnik, V. N. & Chervonenkis, A. Ya. 1971 On uniform convergence of relative frequencies of events to their probabilities. *Theor. Prob. Appl.* **17**, 264–280.

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

Venkatesan, C., Raskar, S. D., Tambe, S. S., Kulkarni, B. D. & Keshavamurty, R. N. 1997 Prediction of all India summer monsoon rainfall using error-back-propagation neural networks. *Meteorol. Atmos. Phys.* **62** (3–4), 225–240.

Wang, W., van Gelder, P. H. A. J. M., Vrijling, J. K. & Ma, J. 2006 Forecasting daily streamflow using hybrid ANN models. *J. Hydrol.* **324**, 383–399.

Wu, C. L., Chau, K. W. & Li, Y. S. 2008 River stage prediction based on a distributed support vector regression. *J. Hydrol.* **358**, 96–111.

Yakowitz, S. 1987 Nearest-neighbour methods for time series analysis. *J. Time Series Anal.* **8** (2), 235–247.

Zhang, B. & Govindaraju, R. S. 2000 Prediction of watershed runoff using Bayesian concepts and modular neural networks. *Water Resour. Res.* **36** (3), 753–762.

First received 19 May 2009; accepted in revised form 20 August 2009. Available online 2 April 2010