Improving reliability of river flow forecasting using neural networks, wavelets and self-organising maps

Mukesh K. Tiwari, Ki-Young Song, Chandranath Chatterjee and Madan M. Gupta

ABSTRACT

Neural network (NN) models have gained much attention for river flow forecasting because of their ability to map complex non-linearities. However, the selection of appropriate length of training datasets is crucial and the uncertainty in predictions of the trained NNs with new datasets is a crucial problem. In this study, self-organising maps (SOM) are used to classify the datasets homogeneously and the performance of four types of NN models developed for daily discharge predictions – namely traditional NN, wavelet-based NN (WNN), bootstrap-based NN (BNN) and wavelet-bootstrap-based NN (WBNN) – is analysed for their applicability cluster-wise. SOM classified the training datasets into three clusters (i.e. cluster I, II and III) and the trained SOM is then used to assign testing datasets into these three clusters. Simulation studies show that the WBNN model performs better for the entire testing dataset as well as for values in clusters I and III; for cluster II the performance of BNN model is better compared with others for a 1-day lead time forecasting. Overall, it is found that the proposed methodology can enhance the accuracy and reliability of river flow forecasting.

Key words | bootstrap, cluster analysis, decomposition, forecasting, Mahanadi river basin, river flow

Mukesh K. Tiwari
Chandranath Chatterjee (corresponding author)
Agricultural and Food Engineering Department,
Indian Institute of Technology,
Kharagpur,
West Bengal 721 302,
India
E-mail: *cchatterjee@agfe.iitkgp.ernet.in*

Ki-Young Song
Madan M. Gupta
Intelligent Systems Research Laboratory,
College of Engineering,
University of Saskatchewan,
Saskatoon,
Saskatchewan S7N 5A9,
Canada

INTRODUCTION

River flow forecasting for short time horizons (hourly, daily) can help to improve water resources management. The task of river flow forecasting is a great challenge. Neural network (NN) models have shown their potential in river flow forecasting and have been successfully applied because of their ability to map complex non-linearities between inputs and outputs (Altunkaynak 2007; Solomatine & Ostfeld 2008; Demirel *et al.* 2009; Tiwari & Chatterjee 2010a, b, c). Substantial literature on NN has been reported in ASCE (2000a, 2000b). NNs are the unstable learning techniques where small changes in training datasets/parameter selection can produce a large change in predicted outputs (Naftaly *et al.* 1997; Carney & Cunningham 2000). NN models are computationally fast and efficient, which makes them a highly suitable tool for river flow forecasting. Disadvantages related to NN models include interpretation of the NN structure ('black box') and their extrapolation capacity (Minns & Hall 1996).

Recently, researchers have been exploring different preprocessing approaches for inclusion of additional hydrological knowledge as input to NN models to improve the hydrological representation and generalization (Corzo & Solomatine 2007a, b). Abrahart (2003) employed the bootstrap technique to continuously sample the input space in the context of rainfall–runoff modelling and reported that it offered marginal improvement in terms of greater accuracies and better global generalizations. Jeong & Kim (2005) used ensemble neural network (ENN) and bootstrap techniques to simulate monthly rainfall–runoff. They concluded that ENN is less sensitive to the input variable selection and the number of hidden nodes than the single neural network (SNN). Jia & Culver (2006) used the bootstrap technique to estimate the generalization errors of neural networks with different structures and to construct the confidence intervals (CIs) for synthetic flow prediction with a small data sample. Han *et al.* (2007) found that the

doi: 10.2166/hydro.2012.130

performance of the NN model for long-term predictions was only probabilistic depending on the arrangement of calibration and testing datasets.

Ensemble forecasting has been suggested to overcome the drawbacks of traditional NN models, and increases the forecasting accuracy by controlling the generalization of predictive model (Jeong & Kim 2005). Boucher *et al.* (2009) found that ensemble forecasts using stacked NNs outperform point forecasts. Ensemble predictions using bootstrap techniques are found to be more accurate and reliable (Sharma & Tiwari 2009; Tiwari & Chatterjee 2010a).

Processing the inputs to NN models using wavelet transformation have further improved the accuracy and consistency of predictions of water resources variables (Adamowski 2008a, b; Kisi 2009; Partal & Cigizoglu 2009; Tiwari & Chatterjee 2010b; Adamowski & Chan 2011; Adamowski *et al.* 2012).

Furthermore, the performance of trained NN models largely depends on the appropriate length of training datasets and their appropriate representation in different river flow profiles such as low, medium, high or extreme (Minns & Hall 2004; Kentel 2009; Tiwari & Chatterjee 2010a). Despite having an appropriate training dataset, there is a probability that an extreme event (i.e. unfamiliar input) beyond the range of recorded datasets might occur in the future and could not be forecasted correctly. Even though it is not straightforward to classify an event as regular or extreme with many input variables, the extreme event can be identified by clustering techniques (Kentel 2009).

In the traditional clustering method (e.g. K-means method), the number of clusters is determined subjectively (Lin & Wang 2006). For similar types of data, different clustering techniques generate different clustering results (Nathan & McMahon 1990). Kentel (2009) tested the fuzzy c-means clustering algorithm and found that it satisfactorily classified input vectors into 'regular event' or 'extreme event' classes. Self-organising maps (SOMs) (Kohonen 2001) are also recognized as a powerful clustering technique and have recently been used in water resources studies (Astel *et al.* 2007; Kalteh *et al.* 2008). SOMs map high-dimensional input datasets into low-dimensional output space, which helps to understand the relationship between complex data.

Moradkhani *et al.* (2004) explored the applicability of a self-organising radial basis (SORB) function to one-step ahead forecasting of daily streamflow. Composed of a Gaussian radial basis function architecture and self-organising feature map (SOFM), SORB was used in data classification. It was found that SORB outperformed the two other NN algorithms. Jain & Srinivasulu (2006) decomposed the flow hydrograph into different segments based on physical concept and modelled different segments using conceptual and NN models. The major finding was that decomposing a flow hydrograph into different segments corresponding to different dynamics based on physical concepts was better than using soft decomposition employed using SOM.

Chang *et al.* (2007) found that SOM NNs can adequately produce streamflow forecasting. They proposed an enforced self-organising map (ESOM) network by recycling the high-flow data to retrain the SOM network. They found that it not only increased the mapping spaces of peak flow in the topological structure of the SOM, but also improved the performance of flood forecasting at high flows. Gopakumar *et al.* (2007) evaluated the performance of NN models for the modelling of daily river flows in a humid tropical river basin with seasonal rainfall pattern. They explored the rainfall and discharge data using SOM and identified the subset of data which had a distinct relationship with the selected hydrologic variables. NN models were developed for each subset and the SOM technique was found to be very helpful for developing logically sound NN models.

Abrahart & See (2000) assessed the performance of NN and ARIMA models for hydrologic forecasting and found that, even though the performance of the models was similar, the NN model performance was improved when the training data were clustered in distinct groups using the SOM technique. Ismail *et al.* (2010) introduced the SOM and least-square support vector machine (LSSVM) model, referred to as SOM-LSSVM model, by combining the capabilities of SOM and LSSVM. The SOM algorithm was used to cluster the training data into several disjointed clusters, and the individual LSSVM models were developed to forecast the river flow. The models were evaluated using flow data from Bernam River located in Selangor, Malaysia. Results showed that the SOM-LSSVM model outperformed other models for forecasting river flow.

In this study, we propose a novel method to improve the predictive capacity of hydrologic forecasting by analysing the performance of different developed NN models

cluster-wise. First, NN models are developed for 1-day lead time forecasting considering the effect of length of training datasets. The effect of input patterns is also investigated using the bootstrap resampling method considering different training algorithms. Thereafter input data are clustered using SOM. Subsequently, the performance of NN models is assessed with cluster-wise model predictions.

In an earlier study, Tiwari & Chatterjee (2010b) developed different NN models such as traditional NN, wavelet-based NN (WNN), bootstrap-based NN (BNN) and wavelet-bootstrap-based NN (WBNN). Firstly, NN models are developed using the significant inputs and optimising the NN structure by trial and error method. Thereafter, wavelets are used to decompose the original time series data at each station into different discrete wavelet components (DWCs). Significant DWCs are then selected and combined to make new time series data at each gauging station. The new time series data are used as input to the NN model to develop WNN models. BNN models are the ensemble of several NN models developed using bootstrap resamples of training dataset, whereas WBNN models are the ensemble of BNN models developed by resampling newly constructed time series using wavelets instead of raw dataset.

In this study, we have used these models to evaluate their performance in simulating clustered datasets using SOMs. Note that we have not developed separate NN models for each identified input cluster, as the number of datasets in some of the clusters was too few. Instead, we have developed NN models using the entire training datasets and then evaluated their performance in forecasting the clustered testing datasets using SOMs. To the best of our knowledge, this is the first such attempt in this direction and will prove to be extremely useful in improving the river flow forecasts.

THEORY

Neural networks (NNs)

Neural networks have been widely used in information processing, systems identification and image processing applications and it has been proved that NNs are capable of model non-linear, multi-input and multi-output systems in an efficient way (Gupta & Rao 1994; Gupta *et al.* 2003). Three-layered feed-forward NNs are widely used in water resources successfully, and consist of an input layer, one hidden layer and an output layer of computational neurons.

NNs map the input vectors $\boldsymbol{x}$ and target vector $\boldsymbol{y}$ by adjusting the weights through the Levenberg–Marquardt algorithm. The performance function of the neural module is defined (Hou & Gupta 2004):

$$E(W) = \frac{1}{2}e^T e = \frac{1}{2}\sum_{k=1}^{N} e^2(k) \tag{1}$$

where N is the number of training datasets and e represents the error between observed values O_i and predicted values P_i. The gradient of error is calculated:

$$\Delta E(W) = J^T(W)e \tag{2}$$

where $J(W)$ is a Jacobian matrix containing the first derivatives of network errors with respect to the weights and biases. The Hessian is determined by:

$$\nabla^2 J(W) = J^T(W)J(W) + S(W) \tag{3}$$

where

$$S(W) = \sum_{k=1}^{N} e\nabla^2 e(k)$$

is usually small and can be ignored. The Hessian can therefore be written:

$$\nabla^2 J(W) \cong J^T(W)J(W) \tag{4}$$

For detailed information on the different properties of NNs and their use in water resources, see Haykin (1999), Maier & Dandy (2000) and Gupta *et al.* (2003).

Bootstrap-based neural networks (BNNs)

The bootstrap is a computational procedure that uses intensive resampling with replacement to estimate the sampling distribution of a statistic (Efron 1979; Efron & Tibshirani 1993). The estimates of different bootstrap resamples can produce more stable and reliable solutions (Efron & Tibshirani

1993). Assume that the data consist of i.i.d. (independent and identically distributed) sample $\mathbf{T}_n = X_1, X_2, \ldots X_n$ of size n with an unknown cumulative distribution function (CDF) F. Let θ be a parameter to be estimated using a function of the data $\hat{\theta} = \hat{\theta}\ X_1, X_2, \ldots X_n$. We want to know the true sampling distribution of $\hat{\theta}$ to determine its bias as an estimator of the true value θ and its variance to know the average and variability in terms of percentiles. Since F is unknown, true sampling distribution of $\hat{\theta}$ cannot be derived. A bootstrap technique that uses true resampling with replacement can approximate the sampling distribution of $\hat{\theta}$ by replacing unknown CDF F with the empirical distribution function $\hat{F}$ by putting mass $1/n$ on $X_1, X_2, \ldots X_n$ to generate a bootstrap resample $\mathbf{T}^*$. Thus, a set of bootstrap samples B can be generated as $\mathbf{T}_n^* = \mathbf{T}^1, \mathbf{T}^2, \ldots, \mathbf{T}^b, \ldots, \mathbf{T}^B$ where B is the total number of bootstrap samples and generally takes the value $50 - 200$ (Efron 1979). For each $\mathbf{T}^b$, an NN prediction model is developed and the output is represented as $f_{NN}(\mathbf{x}_i, \mathbf{w}_b/\mathbf{T}^b)$, built using all n observations (Jia & Culver 2006), where $\mathbf{x}_i$ is a particular input vector and $\mathbf{w}_b$ is a weight vector.

The performance NN model during training is evaluated using the observation pairs that are not included in a bootstrap sample (i.e. testing dataset). The average performance of these NNs built using 200 bootstrap resamples using $\mathbf{T}_n$ is used as an estimate of the generalization error. The generalization error E_0 of an NN model is defined (Twomey & Smith 1998; Jia & Culver 2006):

$$\hat{E}_0 = \frac{\sum_{b=1}^{B} \sum_{i \in A_b} \left[y_i - f_{\mathrm{ANN}}(x_i, w_b/T^b) \right]^2}{\sum_{b=1}^{B} \#(A_b)} \tag{5}$$

and the BNN estimate $\hat{\theta}(x)$ is given by the average of the outputs of NN models developed using B bootstrapped resamples:

$$\hat{\theta}(x) = \frac{1}{B} \sum_{b=1}^{B} f_{NN}\left(\mathbf{x}_i, \mathbf{w}_b/\mathbf{T}^b\right) \tag{6}$$

Wavelet analysis

Wavelet analysis is multi-resolution analysis and provides time and frequency representation of a signal. The continuous wavelet transform of time series $f(t)$ is defined:

$$\mathrm{CWT}(a, b) = |a|^{-1/2} \int_{-\infty}^{+\infty} f(t) \psi * \left(\frac{t-b}{a} \right) \mathrm{d}t \tag{7}$$

where CWT(a, b) is the wavelet coefficient, a is scale parameter, b is translation parameter and * corresponds to the complex conjugate. Wavelet function $\psi(t)$ is called the mother wavelet and is defined:

$$\int_{-\infty}^{+\infty} \psi(t) \mathrm{d}t = 0$$

The shifted and scaled version of the mother wavelet $\psi_{a,b}(t)$ is obtained as:

$$\psi_{a,b}(t) = |a|^{-1/2} \psi * \left(\frac{t-b}{a} \right) \tag{8}$$

Time series decomposition using wavelet transformation generates a wavelet coefficient contour map CWT(a, b) known as a scalogram, which allows the frequency to be determined at different scales. However, a large number of data is generated by computing the wavelet coefficients at every possible scale or resolution level. To avoid this, logarithmic uniform spacing can be used for the a scale discretization with correspondingly coarser resolution of the b locations. This transform is called the discrete wavelet transform (DWT) and is defined (Mallat 1989):

$$\psi_{m,p}(t) = a_0^{-m/2} \psi * \left(\frac{t - p b_0 a_0^m}{a_0^m} \right) \tag{9}$$

where m and p are integers that control the wavelet scale and translation, respectively; a_0 is a specified fined dilation step greater than 1; and b_0 is a location parameter greater than zero. A general choice for these parameters is: $a_0 = 2$ and $b_0 = 1$. By adjusting the scale and the translation parameters based on the powers of dyadic scales and translation (i.e. $a_0 = 2$ in Equation (9)), the volume of data may be reduced considerably and analysis can be more efficient.

DWT operates high-pass and low-pass filters on the original time series. The time series is decomposed into one

comprising low frequencies and its trend (the approximation) and another comprising the high frequencies and the fast events (the detail).

Self-organising maps (SOMs)

SOMs are a kind of NN which use an unsupervised learning algorithm to map high-dimensional input space into low-dimensional space. The SOM clustering technique is non-linear and it has an ability to preserve the topological structure of the data (ASCE 2000a). Figure 1 displays SOM network architecture consisting of one input layer and one output layer (Kohonen layer). The input layer of neurons is fully connected to the output layer and the dimension of the output space is two-dimensional. The topological structure of SOM can be one-, two- and higher-dimensional, but higher-dimensional SOMs are generally not used. Each input data pattern is represented as a vector:

$$\mathbf{x} = [x_1, x_2, \ldots, x_q]^{\mathrm{T}} \tag{10}$$

where q is the dimension of the input data pattern $\mathbf{x}$. The synaptic weight vectors of neurons of the Kohonen layer are denoted (Kohonen 2001):

$$\mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jq}]^{\mathrm{T}}, \quad j = 1, 2, \ldots, l \tag{11}$$

where $\mathbf{w}_j$ is the synaptic weight vector of neuron j in the Kohonen layer and l is the total number of neurons in the Kohonen layer. The synaptic weights are initialized with small random numbers at the beginning of training process. The SOM algorithm computes the similarity measures in terms of Euclidean distance between random input pattern x and synaptic weight vector $\mathbf{w}_j$ as:

$$i(x) = \arg\min_j \|x - w_j\|, \quad j = 1, 2, \ldots\ l \tag{12}$$

Smaller Euclidean distance indicates higher similarity between the input pattern $\mathbf{x}$ and the synaptic weight vector $\mathbf{w}_j$. The winning neuron is therefore selected using the smallest distance from the current input vector to all neurons of the Kohonen layer. After determining the winning neuron, the neurons in the topological neighbourhood are updated as

$$w_j(t+1) = w_j(t) + \eta(t) h_{j,i(x)}(t)(x - w_j(t)) \tag{13}$$

where $\eta(t)$ is the learning rate at time t and $\mathbf{w}_j(t+1)$ is the synaptic weight vector of neuron j at time $t+1$. Here, $h_{j,i(x)}$ determines the degree of neighbourhood between the winning neuron and a new neuron for an input x. This function is required to be symmetric about the winning neuron and decreasing to zero with growing lateral distance from the winning neuron (Haykin 1999). This topological neighbourhood function is represented mathematically as:

$$h_{j,i(x)}(t) = \exp - \left(\frac{d_{j,i}^2}{2\sigma^2(t)} \right), \quad t = 0, 1, 2, \ldots \tag{14}$$

where $d_{j,i}$ is the Euclidean distance between the winning neuron i and the neighbouring neuron j and $\sigma(t)$ is the effective width of the topological neighbourhood, defined:

$$\sigma(t) = \sigma(0) \exp\left(-\frac{t}{\tau_2} \right) \tag{15}$$

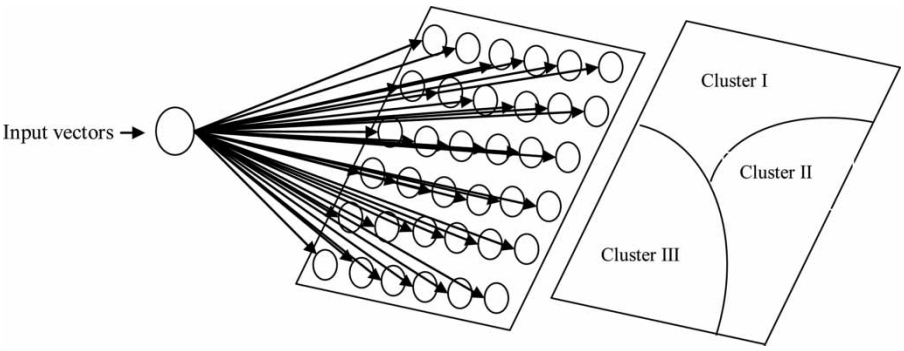

Figure 1 | Architecture of a 2D self-organising feature map.

where $\sigma(0)$ is set to be equal to the radius of the lattice in the output layer of SOM and τ_2 is a constant, defined:

$$\tau_2 = \frac{t_{\max}}{\ln \sigma(0)} \tag{16}$$

Finally, a trained SOM can be obtained. More details regarding the SOM can be found in Kohonen (2001).

STUDY AREA AND DATA USED

Mahanadi River basin, India, flows to the Bay of Bengal in east-central India with a drainage area of 141 589 km^2 and length of 851 km. Mahanadi River basin lies at longitude 80°30′–86°50′ E and latitude 19°21′–23°35′ N. Some important tributaries of the Mahanadi River are Seonath, Jonk, Hasdeo, Mand, Ib, Tel and Ong. Several dams, irrigation projects and barrages are present in the Mahanadi River basin, the most prominent being Hirakud reservoir. Mahanadi River flows during the monsoon season (June–September). The location Naraj, situated at the mouth of the Delta region, was selected for daily discharge forecasting. The location map of different gauging stations is shown in Figure 2.

The data used for the study consist of daily discharge data from six gauging sites from the middle reaches of Mahanadi River basin and the daily release data from Hirakud reservoir during the period 2000–2006. Seven years of daily discharge data for the monsoon period (13 July–25 September) during 2000–2006 yield 525 data patterns. Some of the statistical properties of the discharge data for different years and for different gauging stations are listed in Table 1. The dataset for the years 2000–2004 (375 data patterns) is used for training the models and the dataset for the year 2006 (75 data patterns) is used for testing the performance of the models. The dataset for the year 2005 (75 data patterns) is used as cross-validation that helps to implement an early stopping approach in order to avoid over-fitting of the model during training.

METHOD

Input determination

One of the most important steps in the development of an NN hydrologic model is determining the significant input

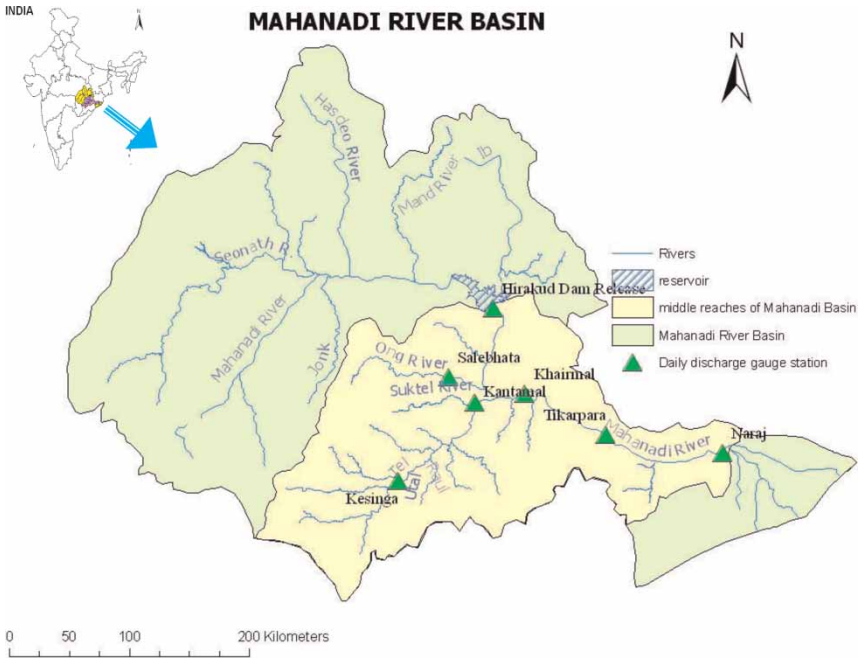

Figure 2 | Index map of the middle reaches of Mahanadi River basin showing location of different gauging stations (after Tiwari & Chatterjee 2010b).

Table 1 | Statistics of the datasets for daily river flow forecasting (13 July–25 September for the years 2000–2006)

Year	Statistics ($m^3 s^{-1}$)	Khairmal	Hirakud Release	Salebhata	Kesinga	Kantamal	Tikarpara	Naraj
2000	Mean	1,758.8	764.4	27.1	356.4	567.6	1,985.6	2,260.2
	Standard deviation	885.5	689.8	22.3	357.7	628.7	839.6	650.4
	Maximum	5,720.0	4,601.3	114.0	2,452.0	3,661.0	4,774.8	5,050.0
	Minimum	509.7	132.9	4.3	113.5	105.0	600.0	901.5
2001	Mean	7,506.9	4,442.9	305.7	1,023.8	1,622.5	6,387.1	10,352.3
	Standard deviation	6,654.9	4,588.5	493.6	921.4	1,159.7	5,704.6	8,962.9
	Maximum	30,468.9	21,567.3	3,225.0	5,293.0	6,500.0	26,700.0	36,714.7
	Minimum	894.8	474.5	8.6	116.5	336.7	1,369.6	1,842.1
2002	Mean	2,128.9	1,131.7	133.9	241.8	448.3	2,119.4	2,967.4
	Standard deviation	2,757.9	1,925.5	239.2	321.7	571.0	2,374.8	3,539.1
	Maximum	14,724.8	10,329.4	1,545.4	1,990.9	2,900.0	12,305.6	16,630.3
	Minimum	305.8	43.2	1.1	10.0	20.2	436.8	215.1
2003	Mean	8,621.4	4,652.4	630.7	1,207.1	1,575.4	7,173.0	11,203.9
	Standard deviation	8,284.7	5,255.9	1,131.5	1,308.1	2,027.1	6,452.6	9,966.4
	Maximum	34,150.1	24,267.2	7,916.0	8,908.4	12,915.1	25,062.0	35,253.2
	Minimum	1,076.0	471.8	1.2	280.0	208.6	687.0	2,111.4
2004	Mean	4,448.0	2,518.3	191.5	718.8	928.7	4,094.9	5,347.9
	Standard deviation	4,531.0	3,178.7	231.2	618.6	673.8	3,771.4	5,349.5
	Maximum	20,331.5	14,952.0	1,004.0	3,240.6	4,014.0	17,744.4	22,465.4
	Minimum	1,121.3	457.5	13.6	182.6	345.5	1,139.1	1,320.7
2005	Mean	5,417.1	3,265.6	144.9	645.5	960.7	4,802.2	7,908.8
	Standard deviation	5,456.3	3,624.2	278.3	1,180.0	1,742.7	4,081.8	6,949.1
	Maximum	26,249.7	13,196.6	1,924.0	8,120.8	11,030.1	19,000.0	24,399.1
	Minimum	849.5	393.9	1.0	113.4	131.8	1,130.0	1,274.9
2006	Mean	7,255.2	3,175.7	543.4	1,217.3	2,035.9	6,789.0	11,399.5
	Standard deviation	5,785.4	3,038.5	824.0	1,294.4	2,935.5	5,491.4	8,847.4
	Maximum	27,099.2	11,870.3	4,681.4	6,483.1	15,276.2	29,000.0	33,979.2
	Minimum	815.5	120.0	2.0	167.9	203.9	600.0	696.7

variables. The current study used a statistical approach suggested by Sudheer *et al*. (2002) to identify the appropriate input vectors. The method is based on the heuristic that the potential influencing variables corresponding to different time lags can be identified through statistical analysis of the data series that uses cross-correlation functions (CCF), autocorrelation functions (ACF) and partial autocorrelation functions (PACF) between the variables. This process was also applied by Tiwari & Chatterjee (2010b) to select significant inputs from the seven discharge gauging stations for daily river flow forecasting.

The cross-correlation statistics for daily discharge are presented in Figure 3. The CCF between the discharge at Kantamal and discharge at Naraj shows a significant correlation for 1–2 days lag as shown in Figure 3(a). For Hirakud dam release and Naraj, the significant correlation is for 1–3 days (Figure 3(b)). To select significant input variables from the Naraj time series itself, the ACF and PACF are used. The ACF (Figure 3(c)) shows a significant correlation at the 95% confidence level for 0 to >20 days, whereas the PACF (Figure 3(d)) shows a significant correlation for up to 2 days lag. The decaying pattern of the PACF confirms the dominance of the autoregressive process, relative to the moving-average process.

A similar procedure is applied to select significant inputs from the discharge data of six gauging stations and Hirakud dam release for daily discharge forecasting. The total input vectors identified are 17 for daily discharge forecasting models, listed in Table 2. These 17 input variables are used to forecast 1-day lead time discharge at Naraj gauging station. The river flow forecast at 1-day lead time (Q_{t+1}) at Naraj gauging station is therefore a function of 17 input variables as identified in Table 2.

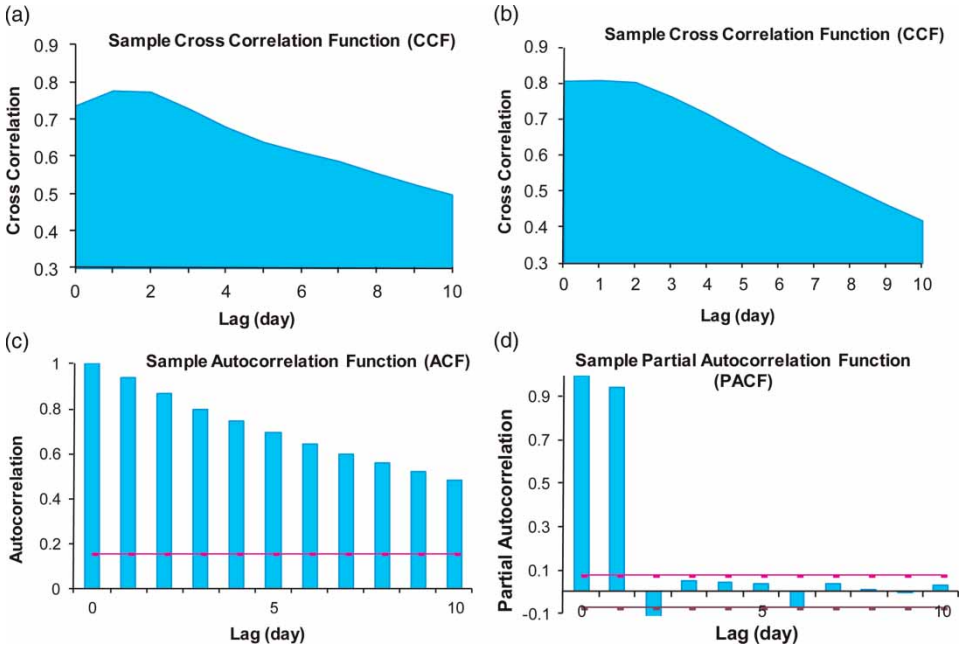

Figure 3 | Correlation statistics for input vector identification for daily discharge forecasting. (a) CCF between discharge at Kantamal and discharge at Naraj; (b) CCF between discharge at Hirakud Dam and discharge at Naraj; (c) ACF of discharge at Naraj; and (d) PACF of discharge at Naraj.

Table 2 | Most significant input variables selected using cross-correlation statistics (CCF, ACF and PACF) for daily discharge forecasting (Q_{t-1} represents discharge at 1-day lag time)

Gauge stations	Input variables
Naraj	Q_t, Q_{t-1}
Tikarpara	Q_t, Q_{t-1}
Khairmal	Q_t, Q_{t-1}, Q_{t-2}
Kantamal	Q_{t-1}, Q_{t-2}
Hirakud Release	Q_t, Q_{t-1}, Q_{t-2}
Salebhata	Q_t, Q_{t-1}, Q_{t-2}
Kesinga	Q_{t-1}, Q_{t-2}

Optimal NN model structure identification

The identification of the optimal network geometry is one of the major tasks in developing NN models. As discussed earlier, the performance of an NN model during training is evaluated using the generalization error which estimates the average error on the testing dataset by a trained NN model built using 200 bootstrap resamples of training dataset. The optimum number of hidden neurons is calculated using the generalization error of various NN structures tested for one to six hidden neurons. The NN structure with four hidden neurons is selected as optimum, for which the generalization error is minimum.

Selection of training algorithms and length of training datasets

In this study, 200 bootstrap resamples are generated to create various data scenarios for each length of training datasets (i.e. 1-, 3- and 5-year training datasets). We have tested three popular training algorithms – namely Levenberg–Marquardt (LM), Bayesian Regularization (BR) and Conjugate Gradient (CGB) – to study the performance of the models for different data scenarios using a 1-year (i.e. 2000), 3-year (i.e. 2000–2002) and 5-year (i.e. 2000–2004) training dataset to train NN models. In all scenarios, the 2005 dataset is used for cross-validation and the 2006 dataset is used to test the performance of the developed NN models.

NN model development

Once an appropriate length of training dataset and an appropriate training algorithm is selected, an NN model is

established. Four NN models (NN, WNN, BNN and WBNN), described in detail in Tiwari & Chatterjee (2010b), are then developed. A brief explanation of these models is provided here.

First, NN models are developed as described in the previous section using the most significant inputs. In the next step, all the time series data are decomposed using DWT and the effective discrete wavelet components (DWCs) are determined using the correlation coefficients between each wavelet component and the observed discharge at Naraj. Correlation between the periodic component and the original discharge data is a measure of the effectiveness of the different wavelet components (i.e. sub-time series). Significant wavelet components of a particular gauging station are added to constitute the new time series. These DWCs data are taken as inputs for the NN model, instead of raw data, to develop the WNN model. By taking the ensemble of 200 NN and WNN models, BNN and WBNN models are developed, respectively. The WBNN model has both bootstrap resampling and wavelet transformation capabilities. Bootstrap resamples of training datasets for BNN and WBNN models are generated using an Excel Add-In named *Bootstrap.xla* (Barreto & Howland 2006). All four models use the same number of training datasets to maintain consistency and all models are tested with one to six hidden neurons and four hidden neurons; the selected optimal structure is that for which the generalization error is minimum.

Cluster analysis

A SOM is trained using the training datasets (375 input patterns from 17 input variables) and the trained SOM is used to cluster the training and testing datasets (75 data patterns from 17 variables). One of the major advantages of SOM over the conventional clustering methods is that the number of clusters and the members of each cluster can be determined objectively without prior assignment of cluster numbers. The SOM classification identifies similar classes, which can further be joined to form homogeneous groups or clusters of dataset. Input data patterns belonging to similar classes activate adjoining nodes on the output layer or Kohonen layer. In this way, closer nodes may be considered representative of similar classes.

The steps to develop a SOM-based clustering method are as follows. Firstly, after SOM training is complete as discussed in the 'Theory' section, all the input patterns are fed into the well-trained SOM. If a neuron in the output layer responds to a specific input pattern, the neuron becomes the winner and is called the image of the specific input pattern. A feature map is then obtained by labelling all winning neurons in the output layer with the identities of corresponding input patterns. The feature map is two-dimensional and composed of grids that represent neurons in the Kohonen layer. The density of winning neurons presents the statistical distribution of input data patterns. If input data patterns are similar in the input space, their images will be located in a certain place of the feature map (Zhang & Li 1993). Accordingly, the SOM feature map can be used to objectively cluster the input data patterns.

The density map can be constructed using feature map. First, the number of data patterns in each grid/node of the feature map is counted. The number of data patterns representing the frequency of grids in the output layer is 'imaged' by specific input data patterns. By visually inspecting the feature and the density maps, the number of clusters and the members of each cluster can be objectively determined.

Performance indices

The Nash–Sutcliffe coefficient E, root mean square error (RMSE) and mean absolute error (MAE) performance indices are used to evaluate the model performance. These performance indices are defined:

$$E = 1 - \frac{\sum_{i=1}^{n} (O_i - P_i)^2}{\sum_{i=1}^{n} (O_i - \overline{O_i})^2} \tag{17}$$

where O_i and P_i are observed and predicted discharge; $\overline{O_i}$ is mean of the observed discharge; and n is the number of data points:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (O_i - P_i)^2} \tag{18}$$

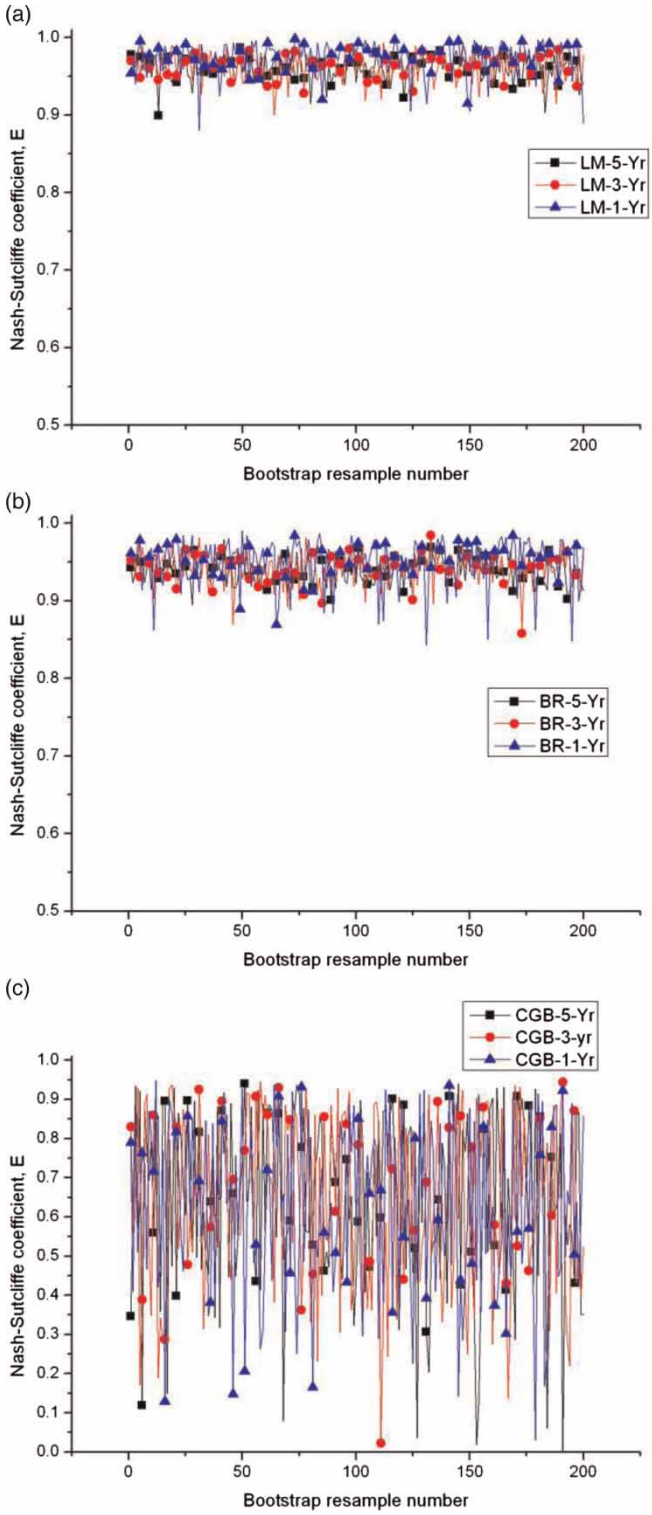

Figure 4 | Performance of NN models during training using (a) LM, (b) BR and (c) CGB training algorithms for 200 bootstrap resamples generated using 1-, 3- and 5-year training datasets.

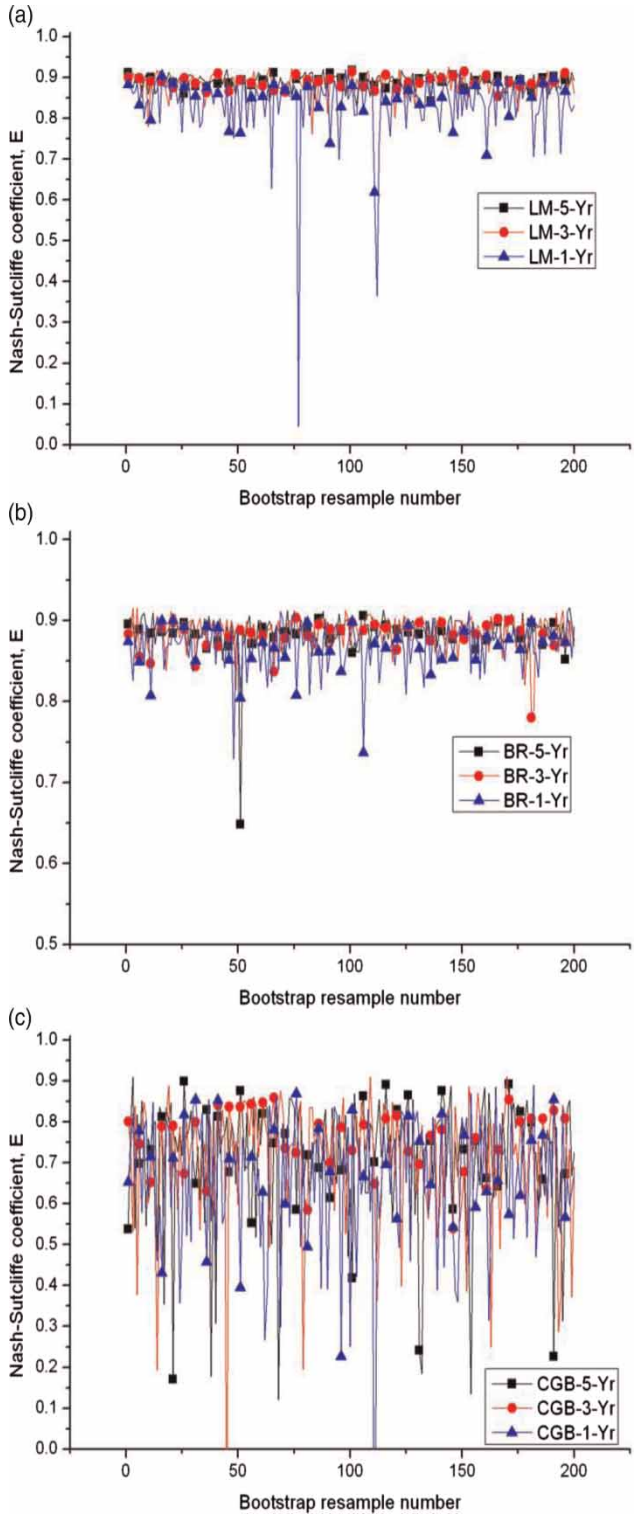

Figure 5 | Performance of NN models during testing using (a) LM, (b) BR and (c) CGB training algorithms for 200 bootstrap resamples generated using 1-, 3- and 5-year training datasets.

and

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|O_i - P_i| \tag{19}$$

SIMULATION RESULTS AND DISCUSSION

NN models are trained with all the 200 bootstrap resampled datasets generated from the 1- (i.e. 2000), 3- (i.e. 2000–2002) and 5- (i.e. 2000–2004) year training datasets, and LM, BR and CGB training algorithms are used to investigate the performance of NN models. Figure 4 depicts the performance of the NN model during training and Figure 5 depicts the performance of NN model during testing. Table 3 lists the performance comparison of NN models with the three algorithms and three different lengths of datasets for training and testing.

It is observed that with 1-, 3- and 5-year training datasets, the training performances of LM and BR are much better than that of CGB. It is also observed that the LM algorithm is more effective than the BR and CGB algorithms to train NN models. It is observed that testing performance using the 3- and 5-year training dataset is much better than the 1-year training dataset for NN models. With a 1-year training dataset, the BR algorithm performed better compared with LM and CGB algorithms for NN models. It can therefore be concluded that the CGB algorithm is less effective for NN models; it is therefore not considered for further analysis.

CDFs are developed to assess the performance and stability of NN models with randomly selected training datasets. Figures 6 and 7 confirm our deduction mentioned above. The increasing steepness of the plotted curve indicates reduced sensitivity of model performance (Yapo *et al.* 1996). An NN model therefore performs better with BR training algorithm compared with LM training algorithm with short-length training datasets. Finally, with the 5-year training dataset, the optimal performance of the NN model is ensured with the LM algorithm; this configuration has therefore been used for cluster analysis.

In this study SOM is used to classify each data pattern shown in Table 2. Each data pattern was taken successively as input vector **x** to the input node of SOM. SOM was initially applied to the training dataset (375 input patterns). These input patterns were iteratively used as input to the SOM

Table 3 | Performance comparison of NN models with three training algorithms and three different lengths of training datasets for training and testing

Training/ Testing	Training algorithm	Training length of data (years)	Mean E	Std. deviation	Skewness	Std. error of skewness	Kurtosis	Std. error of kurtosis	Minimum	Maximum	Percentiles 25	Percentiles 75
Training	LM	5	0.96	0.02	–1.0	0.2	1.7	0.3	0.90	0.99	0.95	0.97
		3	0.96	0.02	–0.8	0.2	0.5	0.3	0.90	0.99	0.95	0.98
		1	0.97	0.02	–1.8	0.2	4.3	0.3	0.88	1.00	0.97	0.99
	BR	5	0.94	0.02	–0.2	0.2	–0.6	0.3	0.90	0.98	0.93	0.95
		3	0.94	0.02	–0.9	0.2	1.4	0.3	0.86	0.98	0.93	0.96
		1	0.95	0.03	–1.6	0.2	2.6	0.3	0.84	0.99	0.94	0.97
	CGB	5	0.66	0.22	–0.7	0.2	0.1	0.3	–0.02	0.94	0.51	0.86
		3	0.64	0.22	–0.4	0.2	–0.9	0.3	0.02	0.95	0.48	0.86
		1	0.63	0.21	–0.4	0.2	–0.5	0.3	0.03	0.95	0.48	0.82
Testing	LM	5	0.89	0.02	–0.9	0.2	0.5	0.3	0.83	0.92	0.88	0.90
		3	0.89	0.02	–1.8	0.2	5.8	0.3	0.76	0.93	0.87	0.90
		1	0.84	0.08	–5.9	0.2	49.2	0.3	0.05	0.92	0.82	0.88
	BR	5	0.89	0.02	–6.9	0.2	73.5	0.3	0.65	0.92	0.88	0.90
		3	0.89	0.02	–1.9	0.2	6.4	0.3	0.78	0.92	0.88	0.90
		1	0.87	0.03	–1.8	0.2	5.9	0.3	0.73	0.91	0.86	0.89
	CGB	5	0.71	0.15	–1.8	0.2	3.9	0.3	0.12	0.91	0.65	0.82
		3	0.71	0.15	–2.4	0.2	9.7	0.3	–0.24	0.91	0.65	0.81
		1	0.66	0.16	–2.2	0.2	11.1	0.3	–0.46	0.89	0.59	0.78

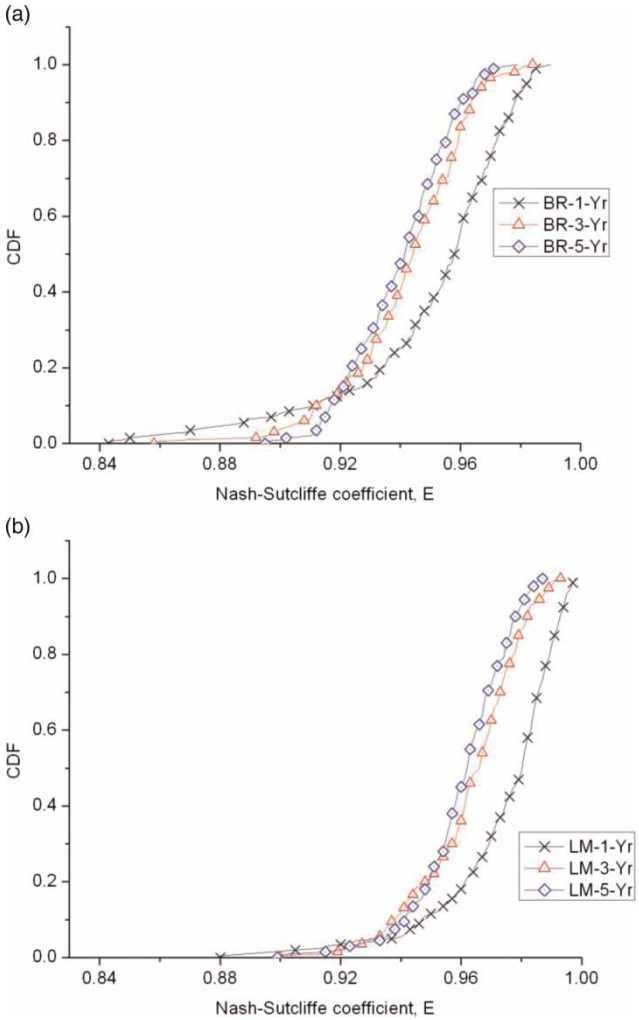

Figure 6 | Empirical CDFs of Nash–Sutcliffe coefficient *E* for training datasets of 5, 3 and 1 year(s) length using (a) BR and (b) LM training algorithms.

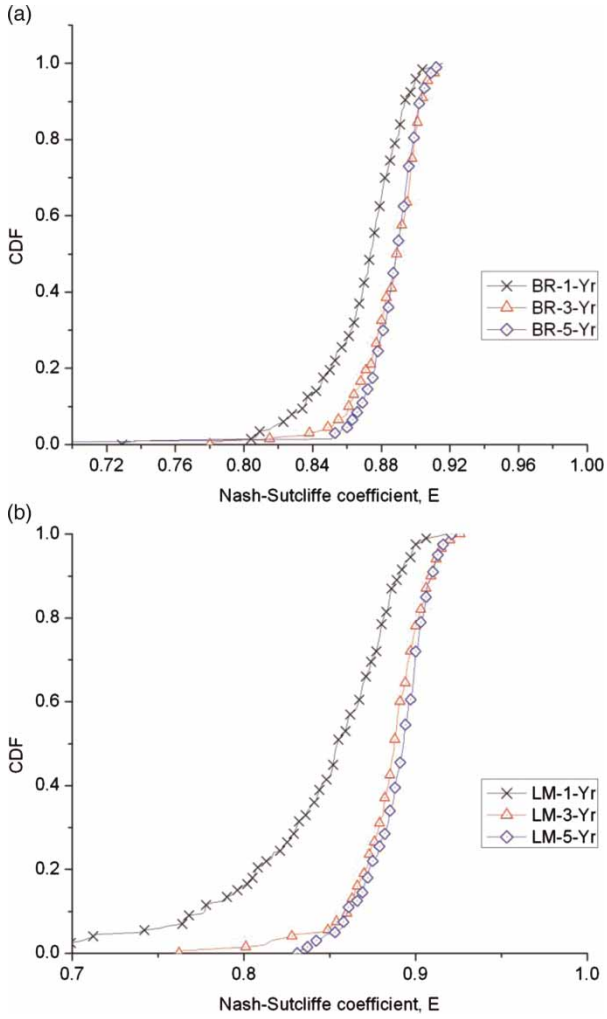

Figure 7 | Empirical CDFs of Nash–Sutcliffe coefficient *E* for testing datasets of 5, 3 and 1 year(s) length using (a) BR and (b) LM training algorithms.

and, after training, all data patterns were classified as either nodes of output layer or Kohonen layer. The SOM technique is used to classify the training and testing dataset objectively by considering three groups which can be representative of low-, medium- and high-discharge values. Several output layer dimensions of SOM were selected (e.g. 4×4, 5×5, 6×6, 7×7 and 8×8) and all dimensions were tested using visual inspection. The 6×6 dimension yielded comparatively better distinction among the desired three clusters, and was therefore selected as the optimal dimension.

A trained SOM or SOM feature map (Figure 8) shows topological relationships among input data patterns where datasets with similar characteristics are closely neighboured, whereas datasets with significant differences are located at a distance from each other. The SOM feature map is used to group the dataset into three groups by visual inspection (two lines in Figure 8); the distance or dissimilarity between the nodes increases as the connection between the nodes changes from bright to dark.

A SOM density map yields the number of clusters for input patterns objectively. The number of training and testing data patterns assigned in different clusters for (a) training datasets and (b) testing datasets are shown in Figure 9. As discussed earlier, SOM is trained using the training datasets and the trained SOM is used to cluster the training and testing datasets. Generally, empty nodes are used to separate different clusters (Lin & Wu 2009), but instead in our study we used SOM neighbour distances

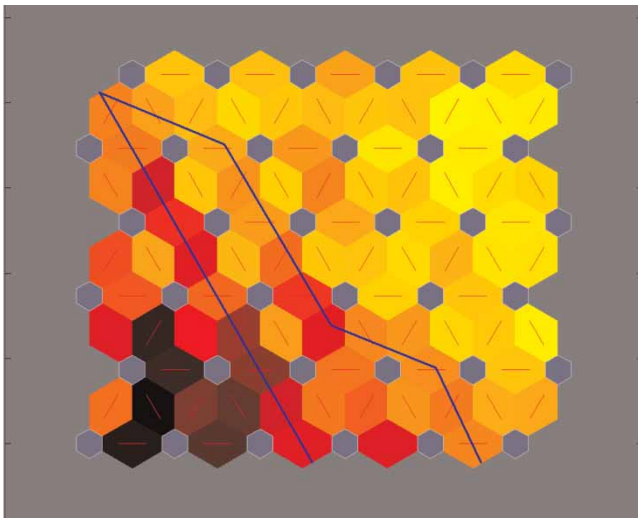

Figure 8 | SOM neighbour distances of the feature map derived from the SOM of 6 × 6 dimension.

for the similarity/dissimilarity between the nodes since no empty nodes were found. To verify the classification by SOM, SOM neighbour distances which show the similarity measures between the nodes are applied by visually inspecting groups which can be demarcated comparatively.

Other authors (e.g. Jain & Srinivasulu 2004; Nayak *et al.* 2005; Tiwari & Chatterjee 2010a, b, c) have evaluated the weakness and strength of different models in predicting different profiles of flow magnitude by dividing the time series data into low-, medium- and high-flow values using simple statistics. In this study the capability of SOM is used to classify the time series data into three clusters (e.g. low-, medium- and high-discharge profiles) and the performance of different NN models are evaluated for different discharge profiles.

The trained SOM neural network is used to cluster the testing dataset. Clustering of different data patterns in the testing dataset is based on 17 input variables (Table 2). However, for the sake of simplicity, clustering of the testing dataset at Naraj gauging station is shown in Figure 10. The data are grouped into clusters I, II and III using the SOM feature map. It is observed that cluster I represents low-discharge values, cluster II represents medium-discharge values and cluster III represents high-discharge values. From Figure 10, it appears that a few data within the medium-discharge cluster are located in the high-discharge cluster; this is because the classification is based on the

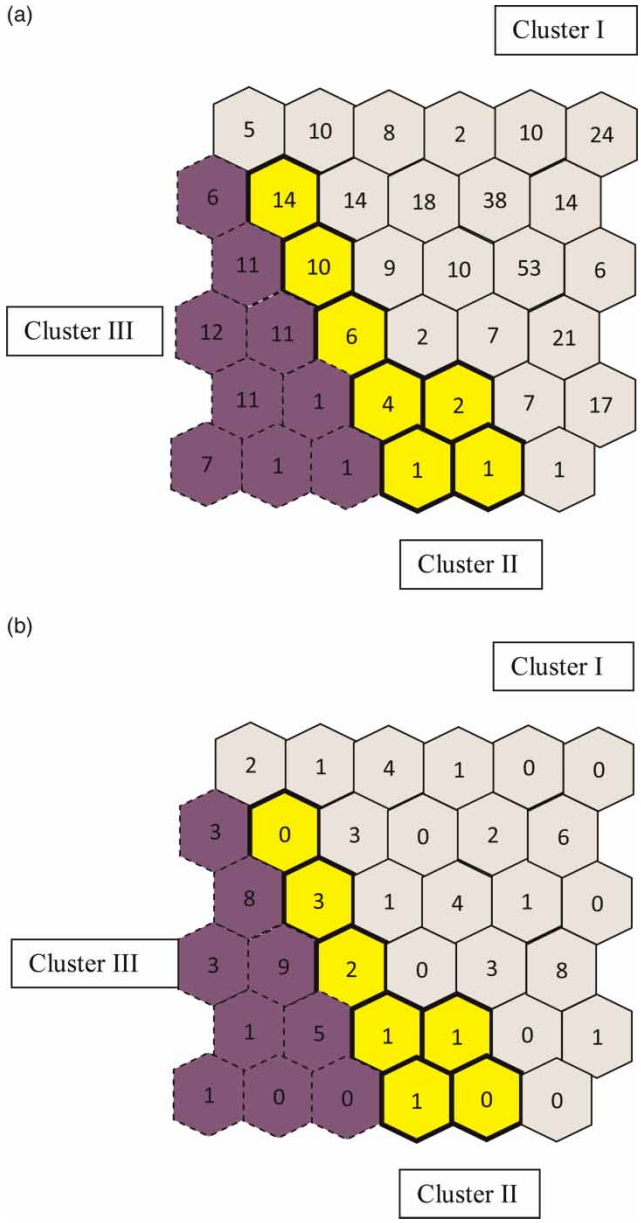

Figure 9 | The density map derived from the SOM of 6 × 6 dimension for (a) training datasets and (b) testing datasets.

non-linear relationship among the 17 input variables. Figure 10 shows a simple representation of clustered discharge values based on a single variable, i.e. for the time series data at Naraj gauging station.

Table 4 shows the number of data patterns in different clusters for training and testing datasets. SOM is first trained using training datasets, three clusters are developed, and the trained SOM is then used to assign corresponding cluster for

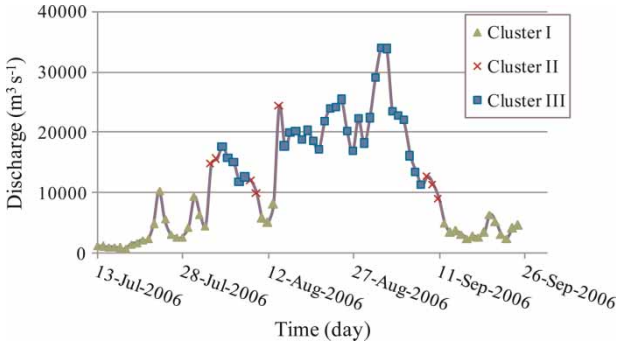

Figure 10 | SOM clustering results for 2006 testing dataset for Naraj gauging station.

Table 4 | Cluster assigned for datasets at Naraj gauging station

Category	Training Number of points	Percentage of the total data	Testing Number of points	Percentage of the total data
Cluster I	276	74	37	49
Cluster II	38	10	8	11
Cluster III	61	16	30	40
Total	375	100	75	100

new datasets (i.e. testing datasets). It is observed that cluster I has the highest percentage of low-discharge values for training and testing datasets and cluster III has the highest percentage of high-discharge values for testing datasets. However, cluster II features the lowest percentage of training datasets and shows unclear grouping as shown in Figure 10. It is observed from Table 4 that the number of data points for training dataset in Clusters II and III are too few to develop separate NN models for these individual clusters. The NN models are therefore developed for the entire training dataset. The performance of these models is then tested for the entire testing dataset and then for different clusters of the testing dataset.

Table 5 shows the performance of NN models for the entire testing dataset for 1-day lead time forecasting. It is observed from Table 5 that the performance of different NN models for the entire testing dataset can be described as WBNN > (is better than) WNN > BNN > NN according to E and RMSE and as WBNN > BNN > WNN > NN according to MAE.

Table 6 shows the performance of NN models for different clusters of testing datasets for 1-day lead time forecasting. It is observed from Table 6 that the performance

Table 5 | Performance of NN models for the entire testing dataset (year 2006) for one day lead time forecasting

Model	E	RMSE ($m^3 s^{-1}$)	MAE ($m^3 s^{-1}$)
NN	0.825	3,675.0	2,428.9
WNN	0.923	2,440.2	1,885.5
BNN	0.915	2,568.8	1,694.8
WBNN	0.936	2,228.8	1,671.7

Table 6 | Performance of NN models for different clusters of testing datasets (year 2006) for 1-day lead time forecasting

Category	E	RMSE ($m^3 s^{-1}$)	MAE ($m^3 s^{-1}$)
NN			
Cluster I	0.481	1,651.4	1,154.5
Cluster II	–0.514	5,576.3	3,638.8
Cluster III	0.268	4,702.0	3,678.2
WNN			
Cluster I	0.601	1,602.1	1,233.2
Cluster II	0.546	3,051.7	2,340.8
Cluster III	0.689	3,065.8	2,566.9
BNN			
Cluster I	0.529	1,573.3	1,085.4
Cluster II	0.763	2,205.8	1,965.8
Cluster III	0.598	3,485.2	2,374.1
Combined	0.915	2,568.8	1,694.8
WBNN			
Cluster I	0.569	1,504.2	929.1
Cluster II	0.539	3,077.1	2,329.6
Cluster III	0.774	2,615.5	2,219.3
Entire testing dataset (with best models)	0.942	2,115.8	1,632.9

of NN models for high-discharge values (Cluster III) can be described as WBNN > WNN > BNN > NN according to E and RMSE. For medium-discharge values (Cluster II), the BNN model performed much better than the other three models. For low-discharge values in cluster I which have a high percentage of training datasets, performance of all four NN models is very similar, with the WBNN model performing the best.

It is also seen in Table 6 that the variation in the Nash–Sutcliffe coefficient for different clustered values is very high compared with that in Table 5 for the entire testing dataset,

which could be misleading. Jain & Sudheer (2008) found that Nash–Sutcliffe coefficient alone is not sufficient to assess the performance of a model and that some other statistical measures also need to be analysed before reaching any conclusions. The WBNN model is therefore selected as the best model for cluster I dataset, based on RMSE and MAE. Further, the best model predictions are selected for clusters I, II and III (i.e. WBNN model predictions for clusters I and III and BNN model predictions for cluster II); E, RMSE and MAE are evaluated as 0.942, 2,115.8 and 1,632.9 $m^3 s^{-1}$, respectively, for the entire testing dataset as shown in the last row of Table 6. This performance is better than that of the best-performing NN model (i.e. WBNN model) for the entire testing dataset, as shown in the last row of Table 5.

This study shows that the selection of a particular model for a particular cluster can play a significant role when the representation of data in a particular cluster is not sufficient. Similar results – that an NN model with a small length of training datasets and appropriate representation can perform as well as an NN model with a large length of training datasets – were reported by Tiwari & Chatterjee (2010a). The existence of cluster II also proved its importance as the values in cluster II were well simulated using the BNN model compared with other models; for other clusters as well as for the entire testing dataset, the WBNN model performs better compared with other models. It can therefore be said that the entire training dataset may be used to develop the NN models, instead of developing individual NN models for different clusters. Further, selection of appropriate NN models for different clusters in the testing dataset leads to an improvement in the prediction performance. However, more rigorous studies may be carried out in future for higher lead times and different basins to confirm this finding.

SUMMARY AND CONCLUSIONS

Improving the model efficiency in river flow forecasting is of utmost importance in water resources planning and management. NN models have emerged as powerful tools in hydrologic forecasting. In this study, we investigated the effect of the length of training datasets as well as input patterns on the performance of NN models. Further, to improve the reliability of NN models with new datasets, SOMs are used to classify the input patterns. In this study it is illustrated that short-length training datasets for traditional NN models are not sufficient for hydrologic forecasting. In order to overcome the limitation of conventional NN models for hydrologic forecasting, training algorithms such as Levenberg–Marquardt (LM), Bayesian Regularization (BR) and Conjugate Gradient (CGB) were embedded in the traditional NN models to improve the performance when using a short-length training dataset.

The simulation studies show that with a 1-year training dataset, the BR training algorithm enhances the performance of NN models more than LM training algorithm. However, with a large amount of training dataset such as a 3- and 5-year dataset, NN models showed very similar performances. Subsequently, NN models (i.e. NN, WNN, BNN and WBNN) are developed using the entire training dataset and it was found that the WBNN model has the best performance during testing when the entire testing dataset is considered.

Further, the SOM clustering technique was used in this study to classify the training and testing datasets into three clusters (I, II and III representing low-, medium- and high-discharge values, respectively) to evaluate the performance of developed NN models (i.e. NN, WNN, BNN and WBNN). The performance of WBNN model is found to be the best compared with other models for clusters I and III, whereas the BNN model provides better performance compared with other models for cluster II for 1-day lead time forecast. Overall, it is found that a selection of appropriate NN models for different clusters classified using SOM leads to an improvement in the prediction performance.

This study has shown the effectiveness of SOM to cluster the data into different groups and to apply a selected model. This approach has the potential to be applied in different fields of engineering and medical sciences. In this study, NN architecture with 17 input variables is selected based on cross-correlations, autocorrelations and partial autocorrelation functions between the variables. However, in future, efforts may be made to examine alternative combinations of inputs for the NNs. Other hydrological data such as precipitation and temperature may also be considered to improve the river flow forecasts.

REFERENCES

Abrahart, R. J. 2003 Neural network rainfall-runoff forecasting based on continuous resampling. *Journal of Hydroinformatics* **5** (1), 51–61.

Abrahart, R. J. & See, L. 2000 Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes* **14** (11–12), 2157–2172.

Adamowski, J. F. 2008a River flow forecasting using wavelet and cross-wavelet transform models. *Hydrological Processes* **22** (25), 4877–4891.

Adamowski, J. F. 2008b Development of a short-term river flood forecasting method for snowmelt driven floods based on wavelet and cross-wavelet analysis. *Journal of Hydrology* **353** (3–4), 247–266.

Adamowski, J. & Chan, H. F. 2011 A wavelet neural network conjunction model for groundwater level forecasting. *Journal of Hydrology* **407**, 28–40.

Adamowski, J., Chan, H. F., Prasher, S. O., Ozga-Zielinski, B. & Sliusarieva, A. 2012 Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada. *Water Resources Research* **48** (1), W01528.

Altunkaynak, A. 2007 Forecasting surface water level fluctuations of Lake Van by artificial neural networks. *Water Resources Management* **21** (2), 399–408.

ASCE 2000a Artificial neural networks in hydrology. I. Preliminary Concepts. *Journal of Hydrological Engineering* **5** (2), 115–123.

ASCE 2000b Artificial neural networks in hydrology. II. Hydrologic applications. *Journal of Hydrological Engineering* **5** (2), 124–137.

Astel, A., Tsakovski, S., Barbieri, P. & Simeonov, V. 2007 Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets. *Water Research* **41** (19), 4566–4578.

Barreto, H. & Howland, F. M. 2006 *Introductory Econometrics: Using Monte Carlo Simulation with Microsoft Excel*. Cambridge University Press, New York.

Boucher, M. A., Perreault, L. & Anctil, F. 2009 Tools for the assessment of hydrological ensemble forecasts obtained by neural networks. *Journal of Hydroinformatics* **11** (3–4), 297–307.

Carney, J. & Cunningham, P. 2000 Tuning diversity in bagged ensembles. *International Journal of Neural Systems* **10** (4), 267–279.

Chang, F.-J., Chang, L.-C. & Wang, Y.-S. 2007 Enforced self-organizing map neural networks for river flood forecasting. *Hydrological Processes* **21**, 741–749.

Corzo, G. & Solomatine, D. P. 2007a Knowledge-based modularization and global optimization of artificial neural network models in hydrological forecasting. *Neural Networks* **20**, 528–536.

Corzo, G. A. & Solomatine, D. P. 2007b Baseflow separation techniques for modular artificial neural networks modelling in flow forecasting. *Hydrological Sciences Journal* **52**, 491–507.

Demirel, M. C., Venancio, A. & Kahya, E. 2009 Flow forecast by SWAT model and ANN in Pracana basin, Portugal. *Journal of Hydrology* **40**, 467–473.

Efron, B. 1979 Bootstrap methods: another look at the jackknife. *Annals of Statistics* **7**, 1–26.

Efron, B. & Tibshirani, R. J. 1993 *An Introduction to the Bootstrap*. Chapman and Hall, London, UK.

Gopakumar, R., Takara, K. & James, E. J. 2007 Hydrologic data exploration and river flow forecasting of a humid tropical river basin using Artificial Neural Networks. *Water Resources Management* **21**, 1915–1940.

Gupta, M. M. & Rao, D. H. 1994 *Neuro-Control Systems: Theory and Applications*. IEEE Press, Piscataway, NJ.

Gupta, M. M., Jin, L. & Homma, N. 2003 *Static and Dynamic Neural Network: From Fundamentals to Advanced Theory*. John Wiley & Sons, Hoboken, NJ.

Han, D., Kwong, T. & Li, S. 2007 Uncertainties in real-time flood forecasting with neural networks. *Hydrological Processes* **21** (2), 223–228.

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

Hou, Z.-G. & Gupta, M. M. 2004 A rail damage detection and measurement system using neural networks. In: *Proceedings of CIMSA 2004-IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, Boston, MA.

Ismail, S., Samsudin, R. & Shabri, A. 2010 River flow forecasting: a hybrid model of self organizing maps and least square support vector machine. *Hydrology and Earth Systems Science D* **7**, 8179–8212.

Jain, A. & Srinivasulu, S. 2004 Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques. *Water Resources Research* **40** (4), W04302.

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

Jain, S. K. & Sudheer, K. P. 2008 Fitting of hydrologic models: a close look at the Nash-Sutcliffe index. *Journal of Hydrological Engineering* **13** (10), 981–986.

Jeong, D. & Kim, Y. O. 2005 Rainfall-runoff models using artificial neural networks for ensemble streamflow prediction. *Hydrological Processes* **19** (19), 3819–3835.

Jia, Y. & Culver, T. B. 2006 Bootstrapped artificial neural networks for synthetic flow generation with a small data sample. *Journal of Hydrology* **331**, 580–590.

Kalteh, A. M., Hjorth, P. & Berndtsson, R. 2008 Review of the self organizing map (SOM) approach in water resources: analysis, modelling and application. *Environmental Modelling Software* **23**, 835–845.

Kentel, E. 2009 Estimation of river flow by artificial neural networks and identification of input vectors susceptible to producing unreliable flow estimates. *Journal of Hydrology* **375**, 481–488.

Kisi, O. 2009 Neural networks and wavelet conjunction model for intermittent streamflow forecasting. *Journal of Hydrological Engineering* **14** (8), 773–782.

Kohonen, T. 2001 *Self-Organizing Maps*. Springer, New York.

Lin, G. F. & Wang, C. M. 2006 Performing cluster analysis and discrimination analysis of hydrological factors in one step. *Advances in Water Resources* **29** (11), 1573–1585.

Lin, G.-F. & Wu, M.-C. 2009 A hybrid neural network model for typhoon-rainfall forecasting. *Journal of Hydrology* **375**, 450–458.

Maier, H. R. & Dandy, G. C. 2000 Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling Software* **15**, 101–124.

Mallat, S. G. 1989 A theory for multi resolution signal decomposition: the wavelet representation. *IEEE Transaction on Pattern Analysis and Machine Intelligences* **11** (7), 674–693.

Minns, A. W. & Hall, M. J. 1996 Artificial neural networks as rainfall-runoff models. *Journal of Hydrological Sciences* **41** (3), 399–417.

Minns, A. W. & Hall, M. J. 2004 Rainfall–runoff modelling. In: *Neural Networks for Hydrological Modelling* (R. J. Hart, P. E. Kneale & L. M. See, eds). Balkema Publishers, Leiden, The Netherlands.

Moradkhani, H., Hsu, K., Gupta, H. V. & Sorooshian, S. 2004 Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. *Journal of Hydrology* **295**, 246–262.

Naftaly, U., Intrator, N. & Horn, D. 1997 Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems* **8**, 283–296.

Nathan, R. J. & McMahon, T. A. 1990 Identification of homogeneous regions for the purposes of regionalization. *Journal of Hydrology* **121**, 217–238.

Nayak, P. C., Sudheer, K. P., Rangan, D. M. & Ramasastri, K. S. 2005 Short-term flood forecasting with a neurofuzzy model. *Water Resources Research* **41**, W04004.

Partal, T. & Cigizoglu, H. K. 2009 Prediction of daily precipitation using wavelet-neural networks. *Hydrological Sciences Journal* **54** (2), 234–246.

Sharma, S. K. & Tiwari, K. N. 2009 Bootstrap based artificial neural network (BANN) analysis for hierarchical prediction of monthly runoff in Upper Damodar Valley Catchment. *Journal of Hydrology* **374** (3–4), 209–222.

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

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

Tiwari, M. K. & Chatterjee, C. 2010a Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks (BANNs). *Journal of Hydrology* **382**, 20–33.

Tiwari, M. K. & Chatterjee, C. 2010b A new wavelet-bootstrap-ANN hybrid model for daily discharge forecasting. *Journal of Hydroinformatics* **13** (3), 500–519.

Tiwari, M. K. & Chatterjee, C. 2010c Development of an accurate and reliable hourly flood forecasting model using wavelet-bootstrap-ANN hybrid approach. *Journal of Hydrology* **394**, 458–470.

Twomey, J. M. & Smith, A. E. 1998 Bias and variance of validation methods for function approximation neural networks under conditions of sparse data. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews* **28** (3), 417–430.

Yapo, P. O., Gupta, H. V. & Sorooshian, S. 1996 Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data. *Journal of Hydrology* **181**, 23–48.

Zhang, X. & Li, Y. 1993 Self-organizing map as a new method for clustering and data analysis. In: *Proceedings of International Joint Conference on Neural Networks*, Nogoya, pp. 2448–2451.

First received 30 September 2011; accepted in revised form 9 August 2012. Available online 16 November 2012