Improved higher lead time river flow forecasts using sequential neural network with error updating

Om Prakash[1], K.P. Sudheer[2*], K. Srinivasan[3]

[1] Former Research Scholar, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai – 600036, India. E-mail: omprakashiitm@gmail.com
[2] Professor, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai – 600036, India.
[3] Professor, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai – 600036, India. E-mail: ksrini@iitm.ac.in
[*] Corresponding author. E-mail: sudheer@iitm.ac.in

Abstract: This paper presents a novel framework to use artificial neural network (ANN) for accurate forecasting of river flows at higher lead times. The proposed model, termed as sequential ANN (SANN), is based on the heuristic that a mechanism that provides an accurate representation of physical condition of the basin at the time of forecast, in terms of input information to ANNs at higher lead time, helps improve the forecast accuracy. In SANN, a series of ANNs are connected sequentially to extend the lead time of forecast, each of them taking a forecast value from an immediate preceding network as input. The output of each network is modified by adding an expected value of error so that the residual variance of the forecast series is minimized. The applicability of SANN in hydrological forecasting is illustrated through three case examples: a hypothetical time series, daily river flow forecasting of Kentucky River, USA and hourly river flow forecasting of Kolar River, India. The results demonstrate that SANN is capable of providing accurate forecasts up to 8 steps ahead. A very close fit (>94% efficiency) was obtained between computed and observed flows up to 1 hour in advance for all the cases, and the deterioration in fit was not significant as the forecast lead time increased (92% at 8 steps ahead). The results show that SANN performs much better than traditional ANN models in extending the forecast lead time, suggesting that it can be effectively employed in developing flood management measures.

Keywords: River flow forecasting; Forecast lead time; Error updating; Artificial neural network; Genetic algorithm.

INTRODUCTION

Flood warning and emergency planning are essential for damage reduction since it is impossible to completely eliminate the flood risk. Conventional engineering design against flood damage is based on developing infrastructure to withstand floods resulting from rainfalls with a specific return period. More and more investments are being made to improve the infrastructure which would bring down the flood damage to acceptable levels. Nonetheless, the investments sometimes could fail to keep the current risk levels from increasing. Rapid urbanization causes increased flood runoff due to increased paved areas. Competing investment demands make it difficult for developing economies to commit funds for mitigation measures. Consequently, flood forecasting is considered as one of the effective methods for non-structural flood management. Therefore, early flood warning is extremely important in reducing the flood damage and avoiding loss of life. A flood warning system should address the technical issues that would make it possible to issue an accurate warning with sufficient lead time. Also, improvement in the accuracy of the higher lead time flood forecasts enables better mitigation of flood damage at control points (potential damage centers) through improved flood control operation of the reservoirs.

A significant amount of literature is available in the area of flood forecasting and flood warning. Traditionally, flood forecasting systems have been developed by combining conceptual hydrological models for the land-phase with a suitable hydraulic routing model to simulate flood propagation throughout the drainage network (Arduino et al., 2005). These models are driven by the environmental forcing factors such as meteorological variables like precipitation, temperature and evaporation. This information may be supplemented by radar-born measurements and ground based gauging networks. Conceptual models (based on the system physics) represent the real physical processes and their interactions involved in rainfall-runoff transformation. However, such models are highly complex, and calibration of such models is not trivial (Duan et al., 1992). Generally a lot of simplification and/or assumptions are made while developing such models. This approach hinders the acceptance of traditional conceptual models to represent the real world system dynamics. Owing to the complexities associated with physically based hydrologic and hydrodynamic simulation models, there has been an increasing interest to employ system theoretic techniques to provide flood forecasting (Hsu et al., 1995). While such models do not provide any insight into the actual hydrologic processes and their interactions, they are found to be good at estimating the flow at a specific location in the drainage network (Sudheer et al., 2003). Recently, significant progress in the fields of nonlinear pattern recognition has been made possible through advances in the branch of nonlinear system theoretic modeling called artificial neural networks (ANN). The development of ANN technique has resulted in a plethora of applications in hydrology, most of them being to rainfall-runoff modeling applied to flood forecasting (ASCE Task Committee, 2000a).

The ANNs have a sound scientific background, and have been reported to have good potential to employ for flood forecasting (Dawson et al., 2006). Comprehensive review of previous applications of ANNs in hydrology can be found in ASCE Abrahart et al., 2010; Dawson and Wilby, 2001; Maier and Dandy, 2000; Maier et al., 2010; Task Committee, 2000a, b. Despite such successful applications, ANNs have not been deployed in operational flood warning systems, except a prototype working example reported by Kneale et al. (2001). This can probably be attributed to various practical disadvantages

that a traditional hydrologist is concerned about: long training times, the potential to overfit the model to a dataset, and a lack of guidance on architecture and parameter settings (Dawson et al., 2006). While these limitations are mostly related to the model building, the limitations of the ANNs to provide accurate flood forecasts at higher lead times is little discussed in literature. In other words, many of the studies that used ANNs to forecast river flow at higher lead time report that the performance of the model is found to deteriorate as the forecast lead time increases. For instance, Dawson et al. (2006) employed ANN to forecast river flow up to 24 hours in advance, and reported that the efficiency (Nash and Sutcliffe, 1970) of the model is only 58% at 24 hour lead time. In an attempt to provide 4 hours ahead forecasts, Bruen and Yang (2005) found that the performance of the ANN is reduced to 76% at 4 hours from 95% at 1 hour ahead. It is to be noted that the performance of ANN models may be good for smaller lead times, but may become worse as the lead time increases. This is plausibly due to inaccurate representation of basin saturation in terms of input information in ANNs at higher lead time. Campolo et. al. (1999) noted that the capacity of a basin to respond to a perturbation is more accurate when recent input information is used. Hence, an improved forecast at higher lead time is possible only when updated information about the basin saturation is provided to the network. Birkundavyi et al. (2002) employed the ANN technique to develop a one-step-ahead forecast for the Mistassibi River, and used it recursively to obtain forecasts up to a lead time of 7 days. That is, the forecasted value of stream flow at a particular time step is used as an input to forecast the stream flow at the next time step and so on. Nevertheless, the performance of the model was poor at higher lead times. This can be attributed to the forecast errors in the current state of the river basin in terms of inputs to the model (exogenous), errors in the model forcing forecasts and errors related to the models itself. In order to improve the estimate of the initial state of the system and to reduce the simulation errors in the forecast period, a data assimilation procedure (real time data updating) is often implemented in the forecast system (Madsen and Skotner, 2005; WMO, 1992).

While major attention has been given to the scientific challenge in refining the rainfall-runoff model structure, notable attention has also been focused on procedures for updating the model forecasts, particularly in real time flood forecasting models (WMO, 1992). The real time river flow updating procedures in operational use differ in details in terms of the variable being updated: the variables can be (i) the model inputs, (ii) the model outputs, and (iii) the model parameters (Moore, 1986). The proper choice of an updating procedure depends on the user requirements, the amount of and the quality of the data available, and the experience and expertise of the decision maker. Updating of output variables, also known as error correction, is the most widely used procedure (e.g. Madsen, 2000; Khu et al., 2001; Refsgaard, 1997). Generally, in an error updating scheme, an error correction forecast model is built based on the observed model residuals, and this model is then superimposed on the simulation model. Such corrections are based on the premise of post-blackening the model forecasts with an amount of residual variance; however, this method neither modifies the model parameters nor the model internal storage contents. While such error updating procedures are implemented in ANNs for flood forecasting (e.g. Shamseldin and O'Connor, 2001), they have not been tested for higher lead time forecasts.

In this paper, we propose a sequential ANN (SANN) architecture which has a built-in-error updating scheme that effectively accounts the parameter updating for improved flood forecasts at higher lead times. SANN comprises of a series of ANNs that are connected sequentially to extend the lead time of forecast, each of them taking a forecast value from an immediate preceding network as input. The output of each network is modified by adding an expected value of error so that residual variance of the forecast series is minimized. The heuristic behind SANN is that the model is able to adapt to the physical conditions prevailing at the time of forecast from one forecast to the next, without any need for calibration or user intervention in general, since it employs a feed-forward mechanism followed by error updating. Details of the proposed SANN model architecture are discussed in later sections. The performance of the proposed model framework is illustrated through various case studies in this paper. The paper also intends to evaluate the relative performance of SANN with that of the traditional ANN models of the form of Eq. (1), Eq. (2) and Eq. (3) (discussed in the following section), for the same case studies. The paper is organized as follows. The following section discusses the background of the formulation of the proposed SANN, followed by the details of its architecture and its training. The details of the case examples are discussed further, and subsequently the details about the SANN model building on these example data sets. The results in terms of the performance of SANN and its comparative evaluation with the traditional ANN models are discussed in the subsequent section. The paper ends with conclusions drawn from the research study.

BACKGROUND

The concept of ANN came approximately 50 years ago (McCulloch and Pitts, 1943), inspired by a desire to understand the human brain and emulate its functioning. While the ANN applications remained dormant for long due to the complexity in and/or non-availability of a training algorithm, a tremendous growth in the interest of this computational mechanism occurred since Rumelhart et al. (1986) illustrated a mathematically rigorous training algorithm for neural networks. Subsequently, ANNs have found applications in many areas of science and engineering. In hydrology, majority of the ANN applications are in the field of rainfall-runoff modeling and/or flood forecasting (ASCE Task Committee, 2000b), probably due the inherent nonlinearity and complexity associated with rainfall-runoff process. The basic concepts about the ANN theory and its applications, such as architecture, parallelism, weight parameters, training, learning algorithm, validation, which have been introduced in numerous hydrological papers, are not reproduced in the body of this paper.

Ann for flood forecasting: forms of models

In the context of applications in hydrology, the ANN technique can be considered as a nonlinear regression technique, which can be used typically in four different ways to forecast river flow at larger lead times. For example, consider modeling a river flow time series, where it is required to forecast the value of flow (y_{t+i}) at time $t+i$, where i is the lead-time. The inputs to the ANN are typically chosen as the values of the time series up to time t and the output will be the forecast value. It may be noted that if the aim of the model is prediction, the output vector has to be mapped against available input information only. In other words, any model that forecasts a flow y_{t+i} at time $t+i$ can use input information up to time t only. General-

ly, the modeler uses a set of n "candidate" examples of the form (x, y), and finds out an optimal set of weights vector by training ANN, where x is the input vector to ANN. Consequently, the functional form of ANN model becomes:

$$y_{t+i} = f(y_t, y_{t-1}, ..., y_{t-j}), \qquad (1)$$

where f is the unknown function mapped by ANN, i is an index representing lead time, and j is the maximum number of time steps in the past considered important in modeling y_{t+i}. The network is normally trained and tested on sufficiently large training and testing sets that are extracted from the historical time series.

In a different modeling framework using ANN, one can utilize the values (or forecasts) of other time series (or external variables), in addition to the previous values of the time series as inputs that have a correlated or causal relationship with the series to be forecasted (Sudheer et al., 2002). For a river flow forecasting problem such exogenous time series could be rainfall, evaporation, and/or temperature over the basin. The functional form of this type of ANN is:

$$y_{t+i} = f(y_t, y_{t-1}, ..., y_{t-j}, z_t, z_{t-1}, ..., z_{t-k}), \qquad (2)$$

where z refer to the exogenous input variables considered as input. The appropriate values of j and k can be determined by the procedure suggested by Sudheer et al. (2002).

Different models of the form in Eqs. (1) and (2) result when different ANN models are developed for various lead times $i = 1, 2, 3, ...,$ etc. Alternatively, the robustness of ANNs (because of their massively parallel structure) can be exploited by having multiple output neurons in one single ANN model each representing stream flows at different lead times, with the same input vector, resulting in a model of the form:

$$[y_{t+1}, y_{t+2}, ..., y_{t+i}] = $$
$$= f(y_t, y_{t-1}, ..., y_{t-j}, z_t, z_{t-1}, ..., z_{t-k}), \qquad (3)$$

Models of the form given in Eq. (3) make a third way of generating river flow forecasts. A fourth way of deriving forecasts at higher lead times is by using the one step-ahead ANNs recursively (Sudheer, 2009). As the forecasts provided by the ANN models at one step-ahead are proven to be good by many researchers (Campolo et al., 1999; Sudheer et al., 2003, among many others), it can be envisaged that the use of an ANN model for the one-step-ahead forecasts may allow an extension of the lead-time up to which a reliable flood forecast may be issued by providing a quick prediction solely based on forecasted values. That is, the forecasted value of stream flow at a particular time step (say, $\hat{y}_{t+1}$) is used as an input to forecast the streamflow at the next time step ($\hat{y}_{t+2}$) and so on. The functioning of these models is to first develop a model of the form of Eq. 1, and use it recursively to get a forecast of y_{t+i}, say $\hat{y}_{t+i}$. Use $\hat{y}_{t+i}$ and the observed values of other variables as inputs to the same model to forecast y_{t+i+1}. Such a recursive approach could be employed for models of the form of Eq. 2 also. However, models of the form of Eq. 2 to be used in recursive forecasting has a limitation that it requires values of exogenous variables in future, such as z_{t+i}, in order to forecast y_{t+i+1}. This can be addressed by providing a predicted value for the variable z (Coulibaly, 2003; Sudheer, 2009).

ANN for flood forecasting: critique and research need

A brief overview of a few selected applications of ANN in flood forecasting is summarized in Table 1. It can be observed from Table 1 that in most of these applications, the developed models are of the form of Eq. (2). It can be noted from Table 1 that the model forecasts are good only at lower lead times (e.g. one step ahead), and in most of these applications, the accuracy of forecasts (in terms of forecast error) at higher lead time is not really satisfactory (efficiency statistic being less than 80%). This observation is significant and creates skepticism in their practical applications, especially when these forecasts are to be used for flood management. It is noted that many of these studies (Table 1) considered independent models for each lead time. To the best of the authors' knowledge, only one application has been reported that used the recursive way to extend the forecast lead time (Birkundavyi et al., 2002) which resulted in an efficiency of 79% at 7 days ahead forecasts. Recently, Chang et al. (2007) reported a serial-propagated neural network for multi-step-ahead flood forecasting, in which the forecasted flow at a lower step-ahead is used as an input to forecast the flow at subsequent steps. This way of extending the lead time of forecast is different from the recursive approach presented by Birkundavyi et al. (2002) in the sense that they (Chang et al., 2007) use independent model for each lead time. However, preventing the prediction errors at lower time steps getting propagated to higher time steps is the major concern in a serial-propagated neural network. This concern would lead to a constraint in extending the lead time of prediction, which is evidently reported by Chang et al. (2007) that the performance of the model is found to be stable only up to 3 steps ahead and gets deteriorated thereafter. While there are a plethora of flood forecasting applications of ANN that have been reported recently, most of them fall in any three of the categories reported in Table 1 (Abrahart et al, 2010; Maier et al., 2010; Sudheer, 2009).

The data presented in Table 1 suggest the requirement for improving the forecast accuracy at higher lead times, if ANNs are to be deployed in operational flood management systems. Improving the forecast accuracy essentially implies reducing the error between the simulated and the observed hydrographs. This error in general falls into three categories: amplitude error, phase error and shape error. The amplitude error represent the over-estimation (or under-estimation) of the flow hydrograph ordinate and may be due to noise in the input data or due to deficiencies in the model structure (Shamseldin and O'Connor, 2001). The phase error indicates the shift in timing of the flood hydrograph and this error is critical in flood forecasting applications. The shape error represents the inaccuracy in preserving the shape (indicating the rate of rise and rate of fall in the flow hydrograph) of the hydrograph. The phase and shape errors are generally induced by inaccuracies in the flood routing module of the forecasting models. While an improvement of the structure of the forecasting model is expected to result in increased forecasting accuracy, many researchers have tried implementing an error updating procedure in order to enhance the forecasting accuracy of the model (e.g. Khu et al., 2001; Madsen et al., 2000; Refsgaard, 1997). The principle behind such updating models is that the simulation model would be coupled with a corresponding error forecasting procedure. Traditionally, the error forecast models are built as a separate add-on module, and are used to partially compensate for the inevitable errors between the flow-forecasts of the simulation model and the corresponding observed discharges. Shamseldin and O'Connor (2001) designed and evaluated a non-linear Auto-Regressive

Exogenous-input model (NARXM) for error updating, which employed ANN for flow forecasting. However, this updating procedure, similar to any other model-output updating proce-

dures, does not intervene in the operation of the base model (simulation model) in the sense that it modifies neither the model parameters nor the model internal storage structure.

Table 1. Summary of selected publications on flood forecasting that use ANN.

Sl. No.	Author(s) and year of publication	M.F.L.T.[1]	Model form	CE[#]	RMSE	Remarks
1.	French et al. (1992)	1 hours	Eq. 1	0.382[*]	0.5246 (mm/h)	[*]CC
2.	Hsu et al. (1995)	1 days	Eq. 2	0.950[*]	16.4900 (m³/s)	[*]CC
3.	Shamseldin (1997)	1 day	Eq. 2	0.903	2.2700[*] (mm/day)	[@]MSE
4.	Zealand et al. (1999)	4 weeks	Eq. 2	0.892	89.2000 (m³/s)	-
5.	Sajikumar and Thandaveswara (1999)	1 month	Eq. 2	0.799	42.8000 (mm)	-
6.	Thirumalaiah and Deo (2000)	3 hours	Eq. 2	0.751	-	-
7.	Coulibaly et al. (2000)	7 days	Eq. 2	0.860	96.6700 (m³/s)	-
8.	Coulibaly et al. (2001)	7 days	Eq. 2	0.925	83.0000 (m³/s)	-
9.	Kneal et al. (2001)	6 hours	Eq. 2	0.716[$]	1.1278 (m)	[$]E₁
10.	Birikundavyi et al. (2002)	7 days	Recursive	0.797	95.8900 (m³/s)	-
11.	Nazemi et al. (2003)	1 day	Eq. 2	0.846[*]	33.5924 (m³/s)	[*]CC
12	Pan and Wang (2004)	3 hours	Eq. 2	0.902	-	-
13.	Jain and Srinivasulu (2004)	1 day	Eq. 2	0.941	0.3870[$]	[&]NRMSE
14.	de Vos and Rientjes (2005)	6 hours	Eq. 2	0.790	0.3570 (m³/s)	-
15.	Bruen and Yang (2005)	4 hours	Eq. 2	0.760	-	-
16.	Nayak et al. (2005)	6 hours	Eq. 2	0.710	77.5200 (m³/s)	-
17.	Corani and Guariso (2005)	5 hours	Eq. 2	0.880	0.0130 (m³/s)	-
18.	Coulibaly et al. (2005)	4 days	Eq. 2	0.710	19.7100 (cm/s)	-
19.	Wang et al. (2006)	10 days	Eq. 1	0.769	211.0000 (m³/s)	-
20.	Dawson et al. (2006)	24 hours	Eq. 2	0.710	0.5836 (m)	-
21.	Parasuraman and Elshorbagy (2007)	1 month	Eq. 1	0.840[*]	3.4000 (m³/s)	[*]CC
22	Chang et al. (2007)	4 hours	Serial propagated	0.43	66.7000 (m³/s)	

[1]Maximum Forecast Lead Time; [#]CE = Nash-Sutcliffe Efficiency;
[*]CC = Correlation coefficients; [@]MSE = Mean of the sum of squares of errors;
[$]E₁ = Coefficient of efficiency with single power to the error term;
[&]NRMSE = Normalized root mean square error.

In this paper, a framework that integrates the simulation model (ANN in this case) and an error updating procedure is presented. A series of ANN models are integrated together, each taking input from a previous ANN, for extending the lead time of forecasts. The error forecast model supports a general linear and non-linear model formulation. Since the functioning of ANNs in this framework is sequential, we termed the proposed model as 'sequential ANN (SANN)'. The details about the structure and training of the sequential ANN are discussed in the following section.

SEQUENTIAL ANN

The architecture of the proposed sequential ANN is presented in Fig. 1. As can be observed from Fig. 1, SANN considers various ANN structures, each of them integrated with a feed-forward mechanism to the next ANN structure in order to increase the lead time. The first structure is similar to a traditional ANN model (for one step-ahead forecast), whose network output is modified with an error model to produce the corrected forecast at one step ahead ($\overset{\backprime}{Q}_{t+1}$). Thus,

$$\overset{\backprime}{Q}'_{t+1} = g_1\left(\sum_{j=1}^{j} f_1\left(\sum_{l=1}^{n} w_{l,j}\, x_l\right) w_{j,1}\right),$$

(4)

$$\overset{\backprime}{Q}_{t+1} = \overset{\backprime}{Q}'_{t+1} + h_1(\varepsilon_t),$$

(5)

where $\overset{\backprime}{Q}'_{t+1}$ is the network output and $\overset{\backprime}{Q}_{t+1}$ is the error corrected forecast from the model at one step ahead; g_1, f_1 are the transfer functions at the output and the hidden neurons respectively; x is the input vector having l elements; j is the number of hidden neurons; w is the connection weights, h_1 is the error function employed for one step ahead forecasts, and ε_t is the residual at time step t.

The structure of the ANN for subsequent lead times is defined by providing an additional input neuron which takes the values from the previous network structure, in addition to the same inputs provided to the previous structure. The output of this structure is also modified by an appropriate procedure to produce the error corrected forecast at second lead time. This computational scheme is continued until the desired lead time forecast is computed. In general, the SANN computations for any lead time $t+i$ can be mathematically represented as:

$$\overset{\backprime}{Q}_{t+i} = g_i\left(\sum_{j=1}^{j} f_i\left(\sum_{l=1}^{n+i-1} w_{l,j} \times x_l\right) \times w_{j,k}\right) + h_i(\varepsilon_{t+i-1}),$$

(6)

where the variables are as defined earlier. Note that the number of hidden neurons in each structure may vary (i.e, j need not be a constant for all i, but have to be optimized). Further, one has the option of varying the transfer functions as well as the error function in every structure of SANN.

Error updating model in SANN

The error forecast model can be defined as a general linear or non-linear model with a one step-ahead prediction (ε_{t+1}) that depends on the previous innovations (autoregressive).

$$\varepsilon_{t+1} = h(\varepsilon_t, \varepsilon_{t-1}, ..., \varepsilon_{t-j}),$$

(7)

The Eq. (7) can be used to get a reasonable estimate of the associated error for any lead time forecast. However, the parameters of this model also need to be calibrated during the training of SANN. In order to ensure fast and accurate flow forecasts, a fully automatic parameter calibration technique has been implemented in SANN. Note that the error forecast models may have different parameters (or may be different models) for different lead times.

SANN training

The parameters of SANN that are to be calibrated include all the connection weights of all the ANN structures and the parameters of the error updating models. Hence standard algorithms for the ANN training (e.g. back propagation) cannot be easily applied to training SANN. However, the parameter optimization of SANN being a nonlinear optimization problem, any nonlinear optimization algorithm can be employed for SANN model identification. Consequently, application of a proper search mechanism is another decisive factor in the successful implementation of the SANN model, in addition to data quantity and quality, and proper selection of a performance measure (i.e. objective function). If the objective function is not continuously differentiable throughout the domain, techniques such as gradient descent may fail to converge. Hence global search algorithms such as Genetic Algorithm (GA) are preferable for this purpose. Genetic algorithms are globally oriented in searching and thus potentially useful in solving optimization problems in which the objective function responses contain multiple optima and other irregularities. The objective function of the parameter estimation is to minimize the total forecasting error defined in terms of root mean square error (RMSE):

in which N is the total number of flows being forecasted, Q_{t+i}^{o} is the observed value of the flow at time $t+i$, and all other variables as defined earlier. Genetic Algorithms (GA), introduced and developed by Holland (1975), are based on Darwinian natural selection and the mechanisms of population genetics. The advantage of using GA in this case is the simultaneous optimization of all the parameters of SANN. Hence, the error updating function parameters also become a part of the GA optimization framework with an objective to reduce the total error from SANN. We employed the Genetic Algorithm Toolbox available with the MATLAB for parameter estimation (Mathworks, 2004).

This scheme of feed-forward mechanism has the advantages of the regular ANN. In addition, every structure in SANN can adapt to the physical conditions prevailing at the time of forecast from one forecast to the next since feed-forward mechanism is followed by error updating in SANN, which eliminates any need for calibration or user intervention in general. It is to be noted that the error updating procedure in SANN help dampen down the propagation of forecast error (at lower time steps) to higher time steps, unlike the serial-propagated neural network proposed by Chang et al. (2007).
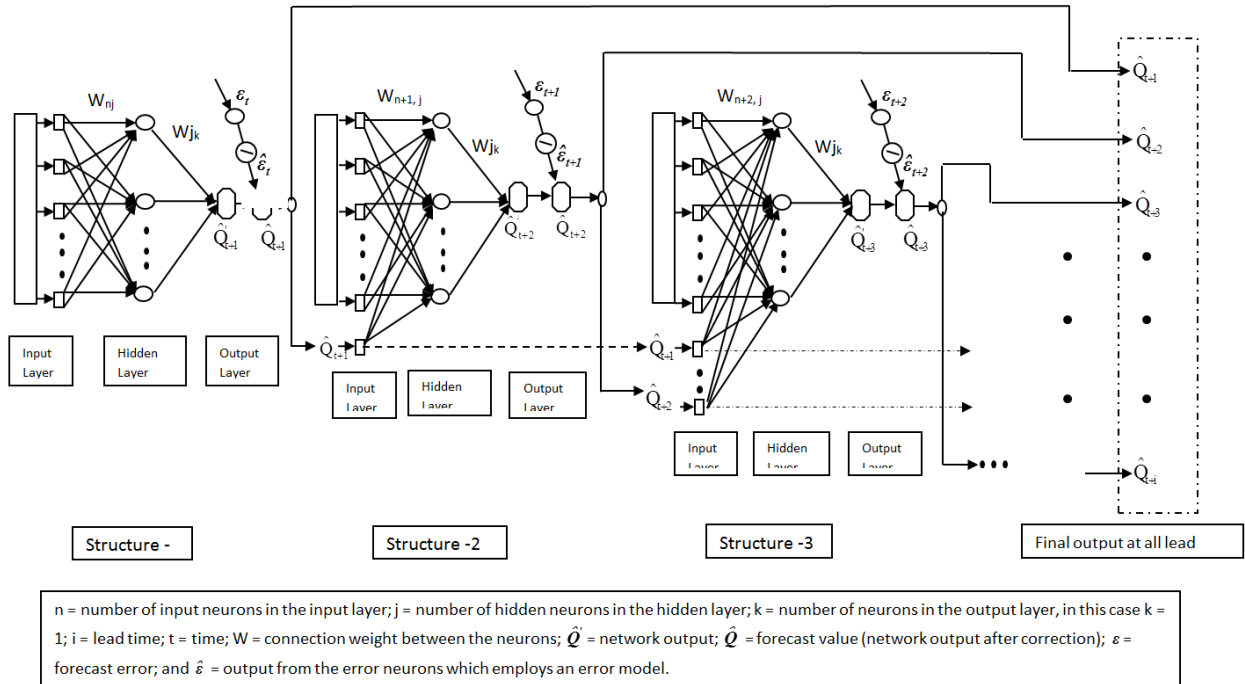
$$RMSE=\sqrt{\frac{\sum_{i=1}^{i}\left[\sum_{t=1}^{N}\left(\hat{Q}_{t+i}-Q_{t+i}^{o}\right)^{2}\right]}{N\,i}}.,\qquad(8)$$

n = number of input neurons in the input layer; j = number of hidden neurons in the hidden layer; k = number of neurons in the output layer, in this case k = 1; i = lead time; t = time; W = connection weight between the neurons; $\hat{Q}$ = network output; $\hat{Q}$ = forecast value (network output after correction); ε = forecast error; and $\hat{\varepsilon}$ = output from the error neurons which employs an error model.

Fig. 1. Sequential Artificial Neural Network architecture.

CASE EXAMPLES

In order to illustrate the potential of the proposed SANN model, the model has been applied to three different case studies: (i) a synthetic (hypothetical) time series that represents a complex flood hydrograph from a hypothetical watershed, (ii) rainfall and runoff data for a specific watershed for predicting the river flow on a daily time step up to 8 days, and (iii) rainfall-runoff data for a different watershed in order to predict the river flow up to 8 hours in advance. The latter two case studies are real world applications: for the daily forecasting, data pertaining to Kentucky River basin, USA have been employed. In the case of hourly forecasting, the data have been obtained for Kolar River basin, India. The characteristics of these example data sets are presented below.

Hypothetical time series

The synthetic generation of the hypothetical time series data has been performed by giving manual perturbation to a daily hydrograph of an unknown catchment. This is done in order to preserve the properties of a typical hydrograph in the synthetic series. The generated data consisted of a total of 7 events (Fig. 2), out of which first 4 have been employed for calibration and the rest for validation.
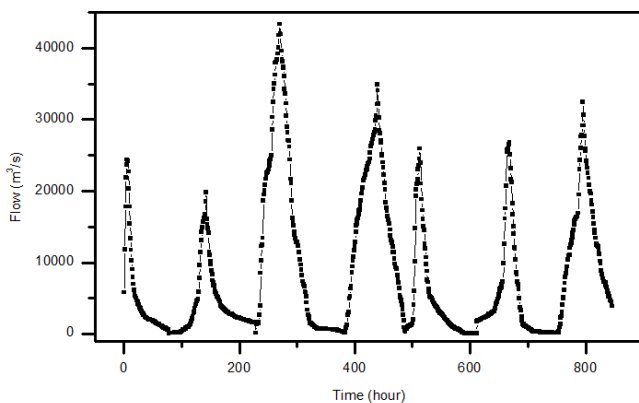

Fig. 2. Hypothetical runoff series containing seven flood events.

Kentucky River basin, USA

The Kentucky River basin (Fig. 3) encompasses over 4.4 million acres (17,820 km^2) of the state of Kentucky. Forty separate counties lie either completely or partially within the boundaries of the river basin. The Kentucky River is the sole water supply source for several water supply companies of the state. There is a series of fourteen Locks and Dams on the Kentucky River, which are owned and operated by U.S. Army Corps of Engineers. The drainage area of the Kentucky River at Lock and Dam 10 (LD10) near Winchester, Kentucky is approximately 10,240 km^2. The data used in this study include average daily streamflow (m^3/s) (US West Optical Company, 1989a), and daily total rainfall (mm) from five rain gauges (Manchester, Hyden, Jackson, Heidelberg, and Lexington Airport) scattered throughout the Kentucky River basin (US West Optical Company, 1989b). The total length of the rainfall-runoff data used is for 26 years (1960–1989 with data in some years missing). The data were divided into two sets: a training data set consisting of daily rainfall and flow data for 13 years (1960–1972), and a testing data set of 13 years (1977–1989).
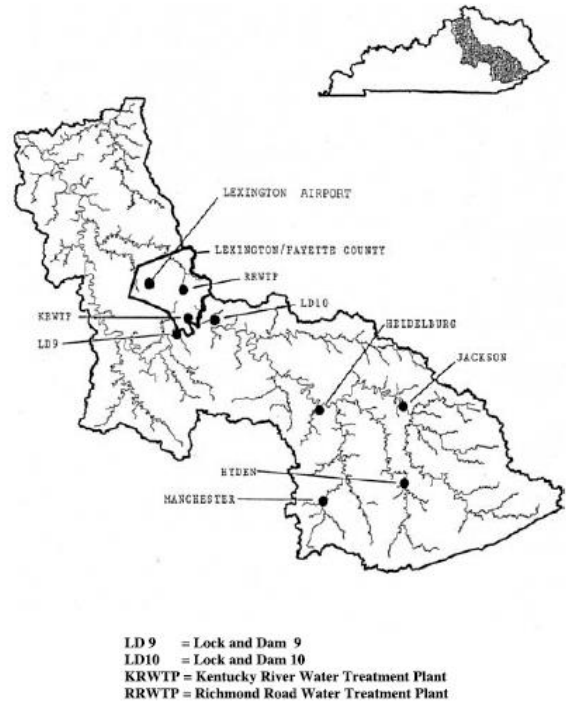

LD 9 = Lock and Dam 9
LD10 = Lock and Dam 10
KRWTP = Kentucky River Water Treatment Plant
RRWTP = Richmond Road Water Treatment Plant
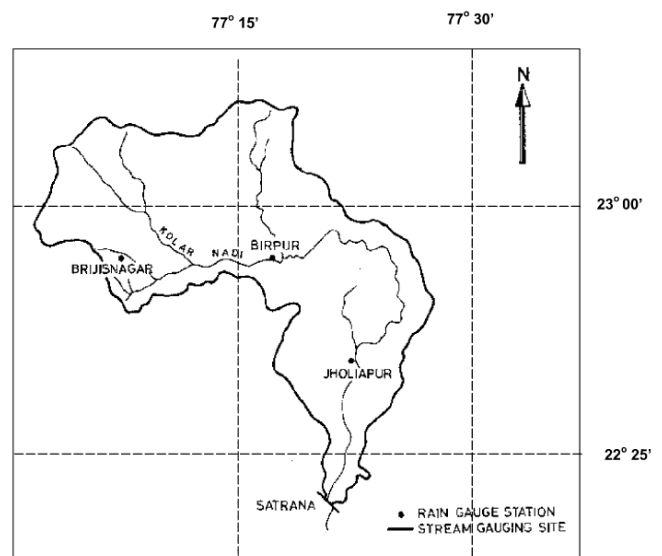
Fig. 3. Kentucky River basin, USA.

Fig. 4. Kolar River basin, India.

Kolar River basin, India

The Kolar River is a tributary of the river Narmada in India that drains an area of about 1350 km^2 before its confluence with Narmada near Neelkanth. In the present study, the catchment area up to the Satrana gauging site (Fig. 4) is considered, which constitutes an area of 903,87 km^2. The 75.3 km long river course lies between north latitude $21^0 \ 09'$ to $23^0 \ 17'$ and east longitude $77^0 \ 01'$ to $77^0 \ 29'$. Topographically, the Kolar sub-basin can be divided into two zones. The upper four fifth, having elevations ranging from 300 to 600 m, is predominately covered by deciduous forests. Soils are skeletal to shallow except near canals where they are relatively deep. In this area, the rocks are weathered and deep fissures are visible. The channel beds are rocky or graveled. General response of this

upper part of the basin to rains is quick. The lower one fifth of the basin consists of a flat-bottomed valley narrowing toward the outlet and having elevations ranging from 300 to 350 m. The area is predominately cultivable and soils are deep and have flat slopes and as such response of this area to rainfall is likely to be quite slow. In this case example, rainfall and runoff data on an hourly interval for Kolar basin during the monsoon season (July, August, and September) for three years (1987–1989) are used. The rainfall data available were in the form of areal average values in the basin. The total available data set was divided into two equal sets in terms of number of patterns employed for calibration and validation of the model.

MODEL DEVELOPMENT
Identification of input vector

One of the most important steps in the model development process is the determination of significant input variables. All of the potential input variables need not be equally informative, since some may be correlated, noisy or have no significant relationship with the output variable being modeled (Maier and Dandy, 2000). Although *a priori* identification using the domain knowledge is widely used in many applications and is necessary to define a candidate set of inputs (e.g. Campolo et al., 1999; Thirumalaiah and Deo, 2000), it is dependent on an expert's knowledge, and hence, is very subjective and case dependent. Intuitively, the preferred approach for determining appropriate inputs and lags of inputs involves a combination of *a priori* knowledge and analytical approaches (Maier and Dandy, 1997). When the relationship to be modeled is not well understood, then an analytical technique, such as cross-correlation, is often employed. Bowden et al. (2005), while reviewing the current state of input selection procedures in water resources applications, report that the cross-correlation methods represent the most popular analytical techniques for selecting appropriate inputs. It is felt that there is good scope for addressing this issue in future studies.

The current study employs a statistical approach suggested by Sudheer et al. (2002) to identify the appropriate input vector. 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. The procedure uses cross-, auto-, and partial auto-correlations between the variables in question at 95% confidence interval. By analyzing these correlogram plots, the significant lags of independent variables that are potentially influencing the output (dependent variable) can be identified. The correlogram for all the three case examples are presented in Fig. 5 (a–h). The input vector identified according to Sudheer et al. (2002) for modeling the river flow in the case of hypothetical time series included a total number of 4 variables, and hence the functional form of SANN in this case is given by:

$$[Q(t+1), Q(t+2), ..., Q(t+8)] = \\ = f[Q(t), Q(t-1), Q(t-2), Q(t-3)], \qquad (9)$$

where $Q(t)$ is the river flow at any time t. Note that the current study explores forecasting the time series up to 8 steps ahead mainly for the demonstration of increased lead time forecasts. However, the SANN structure can be extended and exploited for any lead time.

Similarly for Kentucky basin from Fig. 5, it can be seen that the most appropriate input vector, according to Sudheer et al. (2002), includes stream flows up to a lag of 2 days and precipitation up to a lag of 2 days along with the current day precipita-

tion. In the case of Kolar basin, the input vector included a total of 5 variables: $[R(t-7), R(t-8), R(t-9), Q(t-1), Q(t-2)]$, where $R(t)$ is the precipitation at any given hour t.
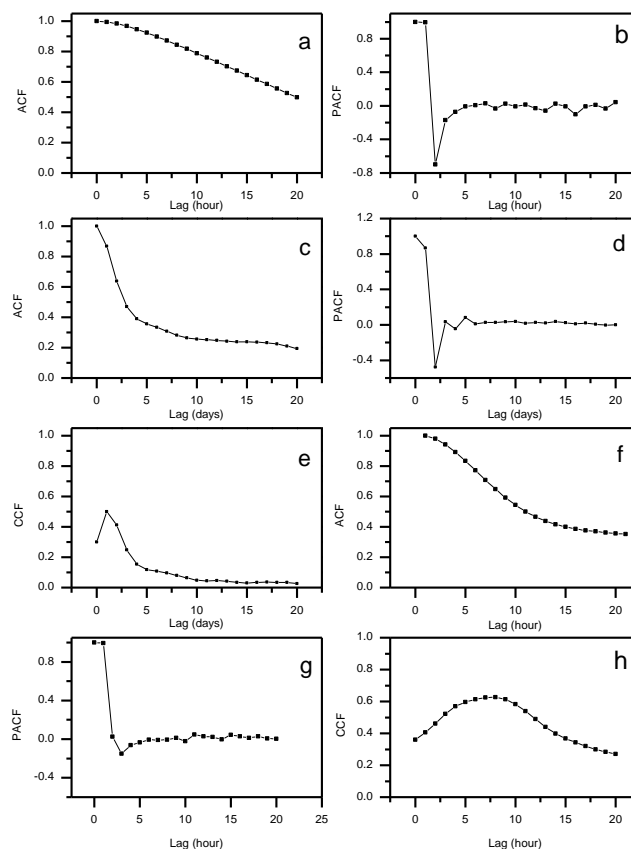

Fig. 5. Autocorrelation (ACF), partial autocorrelation (PACF) and cross correlation (CCF) for the example data sets (a, b: Hypothetical series; c–e: Kentucky basin; f–h: Kolar basin).

Model evaluation

One major problem in assessing ANN models is the use of global statistics (RMSE, correlation, efficiency etc.) in calibration. When this approach is employed for modeling studies, the solution will, in most cases, produce a high or near perfect 'goodness of fit' statistic. Such measures give no real indications of distribution of errors; for instance, the period in which the predictions would be poor. Also, these statistics do not reveal where improvements could be made on the model. Since the neural networks are designed to minimize global errors, a more appropriate metric that identifies real problems or between-network differences is long overdue, and generally analysis is made based on various performance indices. As there is no single universal evaluation test, a multi-criteria assessment is carried out with various global evaluation procedures: (i) coefficient of correlation (CC) between the observed and the forecasted flows; (ii) root mean square error (RMSE) of forecasts; (iii) the standard error of estimate (SEE); (iv) coefficient of efficiency (CE), (Nash and Sutcliffe, 1970). The definition for these statistical evaluation measures are available in many references (e.g. Nayak et al., 2005), and are not provided here.

The above referred error statistics provide relevant information on overall performance of the models, but do not provide specific information about model performance at high flow which is of critical importance in the flood-forecasting context. Hence two additional storm-specific evaluation measures are also considered: (i) percent error in predicted peak flow and

(ii) percent error in total runoff volume. Both these evaluation measures are computed as the ratio of error in predicted flow and the corresponding observed values expressed as a percent. In addition, the predictive uncertainty of the models is evaluated by an appropriate index.

RESULTS AND DISCUSSIONS
SANN model development

Besides the fundamental question of choosing the right set of input variables, defining an adequate neural topology for approximating a function by a neural network still remains an unsatisfactorily solved question. The search for a satisfactory compromise between good data fitting on one side and good generalization properties on the other side has oriented the design of several on-line and off-line model building procedures (Karunanithi et al., 1994; Kwok and Yeung, 1997; Le Cun et al., 1990). But both types of step-by-step evolution do not ensure arriving at the best topology (Chetan and Sudheer, 2006). In hydrological applications, the number of hidden neurons, which is responsible for capturing the dynamic and complex relationships between various input and output variables, is often arrived at after a long trial and error based identification procedure, and the same is followed in this study also.

In the current study, a single hidden layer with sigmoid function nodes is used in all the ANN structures in SANN. The sigmoid activation function is considered in the output layer also. As the sigmoid transfer function has been used in the model, the input-output data have been scaled appropriately to fall within the function limits. The trial and error procedure for identifying the number of hidden neurons started with two hidden neurons initially in each structure, and the number of hidden neurons was increased up to 10 with a step size of 1 in each trial. For each set of hidden neurons, the network was trained in batch mode to minimize the mean square error at the output layer. Various combinations of hidden neurons in the different ANN structures have been tested during the trial. The training was stopped when there was no significant improvement in the efficiency and the model was then tested for its generalization properties. The parsimonious structure that resulted in minimum error and maximum efficiency during the training as well as the testing was selected as the final form of the SANN model.

Since GA has been employed as the parameter estimation procedure in this study, a sensitivity analysis of GA parameters (the number of populations and number of generations) for all the case examples are performed. Based on this analysis the GA parameters have been fixed to be: populations size = 1000 and generations = 5000. Note that the weight parameters of SANN were constrained within the range −10 to +10 based on the experienced gained from the earlier studies conducted in these basins (Jain et al., 2004; Chetan and Sudheer, 2006; Nayak et al., 2007). We implemented GA in MATLAB (Mathworks, 2004) using single point cross over with a probability of 0.3 and stochastic universal sampling for selection to be considered in mating pool.

As stated earlier, the function for updating the error has the provision of considering any linear or nonlinear functional form. An appropriate form of the function has been arrived at by various trials. Assuming the errors to follow Gaussian distribution with zero mean and a variance of 0.5, a function of the following form was tried initially:

$$\varepsilon_{t+1} = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x^2} dx, \tag{10}$$

in which $x = \varepsilon_t$ the network output error at time t. However, it is observed that the forecasted flow series (error corrected) contained a few negative values (6% of total validation data in the hypothetical series; 15% in the case of Kentucky River), which is not practical. It was also observed that these negative values were predicted at the beginning of flood events, though their impact was not visible in subsequent forecasts. This behavior was observed for SANN when the error updating function was changed to a quadratic or a third degree polynomial. However, a simple arithmetic average of the two previous errors, when added back to the network output, the SANN model performance was found to be good and no negative flow forecasts were obtained. The reason for such negative forecasts, specifically at the beginning of a flood event when a nonlinear error model is employed, need further exploration and is beyond the scope of this paper.

Generalization properties of SANN

The performance of the SANN model in forecasting the river flow up to 8 steps ahead in all the three case examples is presented in Table 2 in terms of the performance indices. The efficiency term is an index that evaluates the models' ability to predict values away from the mean and any value above 80% is considered to be 'very good model performance' according to Shamseldin (1997). It is observed that SANN performs with very good efficiency (>80% during calibration as well as validation) in all the three cases. It is noted that the SANN model forecast the flow 8 steps in advance (for hypothetical series) with an efficiency of 99% (during calibration). In the case of Kentucky River, the efficiency of the model to forecast daily river flow 8 days in advance is 71% which is satisfactory for water resources planning. Forecasts at higher accuracy are desired for flood management and are generally expected a few hours in advance. This is achieved by the SANN model (for Kolar basin) as it is able to forecast the flow with an efficiency of 86% at a forecast lead time of 5 hours which falls down to only 84% in 8 hours advance forecasts.

The correlation coefficient is found to be high (Table 2) in all the three cases at all lead times (a value close to 1.0 is ideal) and this clearly illustrates that the SANN model effectively preserves the trend in the river flow series. The RMSE statistic, which is a measure of the residual variance, is ranging from 47.33 m^3/s at 1 hour lead time to 76.69 m^3/s at 8 hour lead time in the case of Kolar River, and is acceptable in the premise of an average flow value of 57.99 m^3/s. In the case of Kentucky River, the RMSE varies between 80.71 m^3/s at 1 day lead time and 123.99 m^3/s at 8 day lead time, and the average flow is 151.85 m3/s. The variation of RMSE between various lead times indicates that the forecasts produced by the SANN models are reliable.

Table 2. Generalization properties of the SANN models at various lead time forecasts for Hypothetical Series, Kentucky and Kolar basins.

Case example	Statistical index	Forecast lead time							
		1-step ahead	2-steps ahead	3-steps ahead	4-steps ahead	5-steps ahead	6-steps ahead	7-steps ahead	8-steps ahead
		Calibration							
Hypothetical	CC[*]	0.9975	0.9942	0.9965	0.9963	0.9966	0.9964	0.9954	0.9960
	CE[#]	0.9950	0.9884	0.9929	0.9925	0.9932	0.9927	0.9906	0.9921
	RMSE (m³/s)	750.1300	1148.7400	901.2930	0.9925	879.1270	908.5820	1030.1100	941.6310
Kentucky (Forecast lead time: days)	CC	0.9462	0.8499	0.8666	0.8626	0.8633	0.8566	0.8591	0.8666
	CE	0.8935	0.7057	0.7416	0.7394	0.7402	0.7268	0.7269	0.7489
	RMSE (m³/s)	77.9250	129.5380	121.4205	121.9420	121.7460	124.8630	124.8480	119.7240
Kolar (Forecast lead time: hours)	CC	0.9830	0.9448	0.9394	0.8998	0.9437	0.9356	0.9008	0.9308
	CE	0.9650	0.8896	0.8787	0.8022	0.8840	0.8669	0.8043	0.8572
	RMSE (m³/s)	36.3370	64.4384	67.6690	86.2647	66.1883	70.7572	85.9607	73.3111
		Validation							
Hypothetical	CC	0.9958	0.9877	0.9915	0.9930	0.9921	0.9909	0.9923	0.9934
	CE	0.9915	0.9755	0.9825	0.9857	0.9833	0.9812	0.9842	0.9868
	RMSE (m³/s)	744.0500	1263.9600	1068.5800	966.5100	1042.9400	1106.3400	1012.2900	920.6120
Kentucky (Forecast lead time: days)	CC	0.9379	0.8311	0.8496	0.8419	0.8378	0.8403	0.8363	0.8451
	CE	0.8772	0.6697	0.7086	0.7015	0.6937	0.6977	0.6835	0.7103
	RMSE (m³/s)	80.7090	132.3960	124.3451	125.8630	127.4790	126.6500	129.6020	123.9900
Kolar (Forecast lead time: hours)	CC	0.9711	0.9444	0.9361	0.9008	0.9322	0.9238	0.8998	0.9249
	CE	0.9405	0.8889	0.8722	0.8043	0.8608	0.8399	0.8022	0.8443
	RMSE (m³/s)	47.329	64.7663	69.3336	85.9609	72.3753	77.7529	86.2645	76.6876

[*]CC: Coefficient of correlation;
[#]CE: Coefficient of Efficiency.

In an analysis to check the SANN model's potential to preserve the statistical properties of the flow series, it is observed that the SANN model preserves mean, standard deviation and skewness effectively. The summary statistics of the observed and the forecasted flow series at various lead times are presented in Table 3 for comparison. It is noted that mean flow is estimated by SANN at all lead times with an error of less than 1% in all the three cases. An original skewness of 3.73 in the Kentucky basin data is reproduced reasonably well by the SANN model. A similar observation can be made in the case of Kolar basin also. In the case of hypothetical series, the summary statistics are preserved satisfactorily despite high magnitude of values.

Prediction of flood hydrograph characteristics of SANN

In order to critically examine the forecasting characteristics of the SANN models, a few typical flood events (during the validation period) were considered, and the models' prediction of these events were evaluated. The prediction characteristics of the flood hydrograph viz. percentage error in predicted peak flow (PE), percentage error in predicted hydrograph volume (VE) along with the time difference to predict the peak flow, are presented in Table 4 for all the three case examples. It is evident from Table 4 that most of the peak flows in the hypothetical series are predicted with reasonable accuracy by SANN. Most of the errors were within ±6%, except one event having a peak flow 25918 m³/s which is predicted with 13% error.

Table 3. Summary statistics of flow forecasted by the SANN model at different lead times during validation period for hypothetical series, Kentucky basin and Kolar basin.

Statistic	Observed value	Forecasted value for lead time							
		1-step ahead	2-steps ahead	3-steps ahead	4-steps ahead	5-steps ahead	6-steps ahead	7-steps ahead	8-steps ahead
		Hypothetical series							
Mean	7106.4	7105.2	7118.2	7133.6	7130.1	7131.3	7083.3	7042.5	6992.9
Std. Dev.	8092.41	8145.7	8075.1	8228.3	8165.1	8262.2	8224.8	8169.6	8035.7
Skewness	1.21	1.22	1.2	1.26	1.23	1.25	1.27	1.2	1.25
		Kentucky basin (forecast time steps: days)							
Mean	5093.87	5093.9	5093.9	5093.9	5094	5093.9	5093.6	5093.5	5093.4
Std. Dev.	8134.67	8037.4	7940.4	7847.5	7544.2	7553.4	7577.8	7830.7	7382
Skewness	4	3.68	3.93	3.93	3.92	3.68	3.95	3.79	3.87
		Kolar basin (forecast time steps: hours)							
Mean	57.99	57.99	57.96	57.99	57.96	57.99	57.96	58	57.97
Std. Dev.	194.01	198.19	194.22	193.84	191.54	198.44	202.13	191.17	200.32
Skewness	7.39	7.26	7.46	7.19	7.34	7.77	8.56	7.3	7.92

Table 4. The SANN-estimated hydrograph characteristics for selected flood events at different forecast lead times during validation period for hypothetical series, Kentucky basin and Kolar basin.

Observed peak flow (m³/s)		Forecasted peak flow for different lead time							
		1-step ahead	2-steps ahead	3-steps ahead	4-steps ahead	5-steps ahead	6-steps ahead	7-steps ahead	8-steps ahead
Hypothetical series									
25918.00	P.E.[1]	−0.29	3.76	0.63	2.35	−13.18	−5.73	3.76	4.50
	T.D.[2]	−1.00	−1.00	−1.00	−1.00	−2.00	−2.00	−2.00	−1.00
	V.E.[3]	−0.54	−1.20	−2.10	−2.21	−3.24	−2.71	−2.30	−2.83
26787.00	P.E.[1]	−1.99	0.60	−10.32	−0.49	−6.21	−7.43	0.60	−0.14
	T.D.[2]	0.00	−1.00	−1.00	3.00	2.00	3.00	−1.00	−1.00
	V.E.[3]	−0.13	−0.29	−0.31	−0.20	−0.30	−0.29	0.26	0.27
Kentucky basin (forecast time steps: days)									
1543.44	P.E.[1]	6.72	4.86	−16.79	−30.06	10.99	18.18	4.90	15.71
	T.D.[2]	0.00	−1.00	0.00	−1.00	−1.00	0.00	−1.00	−1.00
	V.E.[3]	−0.92	−2.05	−2.44	−1.75	−0.71	−1.29	−0.30	−1.29
1291.39	P.E.[1]	−3.70	2.85	−15.04	3.89	8.85	13.05	2.93	12.64
	T.D.[2]	−1.00	−2.00	−1.00	−2.00	−2.00	−1.00	−2.00	−2.00
	V.E.[3]	−0.06	−0.21	−0.26	0.09	0.20	0.22	2.00	3.30
2208.96	P.E.[1]	3.22	2.56	7.27	7.29	14.61	2.89	13.10	18.78
	T.D.[2]	−1.00	−1.00	−2.00	−2.00	0.00	−1.00	−3.00	0.00
	V.E.[3]	0.04	0.01	−2.23	−0.48	−0.71	−1.30	−0.66	−1.55
2084.35	P.E.[1]	10.44	5.57	15.60	−1.06	11.03	11.07	6.09	15.32
	T.D.[2]	0.00	−2.00	0.00	−1.00	−2.00	−2.00	−2.00	−1.00
	V.E.[3]	0.48	2.09	1.68	2.28	1.71	1.64	1.00	1.96
Kolar basin (forecast time steps: hours)									
1392.10	P.E.[1]	−26.95	−7.93	−4.53	10.64	−14.73	−30.66	6.72	−26.23
	T.D.[2]	0.00	0.00	−1.00	−1.00	0.00	−1.00	−1.00	−1.00
	V.E.[3]	0.05	0.13	−0.05	0.18	0.12	0.12	0.15	0.20
1061.60	P.E.[1]	−19.08	−0.90	−7.97	−0.80	−11.61	−9.40	1.54	−18.08
	T.D.[2]	0.00	−1.00	0.00	−2.00	1.00	−2.00	−1.00	−1.00
	V.E.[3]	0.16	0.36	−0.09	0.41	0.39	0.39	0.40	0.40
1294.10	P.E.[1]	−7.22	19.94	−5.93	0.73	−18.64	−13.42	−1.50	−36.62
	T.D.[2]	0.00	1.00	−2.00	−2.00	0.00	−1.00	−2.00	−1.00
	V.E.[3]	0.01	−0.01	0.03	0.01	0.02	0.02	−0.06	−0.03
2029.00	P.E.[1]	7.93	2.49	7.16	3.50	12.40	−34.83	2.44	−2.11
	T.D.[2]	0.00	−1.00	0.00	−1.00	−1.00	1.00	−1.00	−1.00
	V.E.[3]	0.30	1.01	0.41	1.52	1.68	1.68	1.56	1.61

[1]Percent error in forecasted peak flows;
[2]Time difference to peak (hours);
[3]Hydrograph volume error (%); (+ve) values indicate underestimation; (-ve) values indicate overestimation.

Table 5. Estimated hydrograph characteristics at different forecast lead times during validation period for hypothetical series, Kentucky basin and Kolar basin (comparison between SANN, ANN1 and ANN2 model performance at selected events).

Observed peak flow (m³/s)	Lead time	1 step ahead			5 steps ahead			8 steps ahead		
	Model	P.E.[1]	T.D.[2]	V.E.[3]	P.E.[1]	T.D.[2]	V.E.[3]	P.E.[1]	T.D.[2]	V.E.[3]
Hypothetical series										
32429.00	ANN1	−6.46	−1.00	−0.18	−3.31	−5.00	0.77	−13.28	−8.00	2.70
	SANN	−0.01	−1.00	1.23	1.13	−2.00	0.04	2.08	−2.00	−0.42
	ANN2	−4.07	−2.00	−0.48	−1.78	−5.00	1.97	−1.16	−8.00	3.46
Kentucky basin (forecast time steps: days)										
2806.51	ANN1	6.32	0.00	5.04	62.65	−8.00	33.52	78.16	−7.00	68.49
	SANN	8.06	−1.00	0.49	2.73	−2.00	1.56	0.14	−3.00	1.70
	ANN2	0.46	0.00	8.14	21.77	−3.00	39.42	91.90	−9.00	88.71
Kolar basin (forecast time steps: hours)										
2427.70	ANN1	3.19	−1.00	3.97	11.35	0.00	28.25	32.08	0.00	49.02
	SANN	3.01	−1.00	0.21	−7.80	0.00	2.01	−8.21	0.00	3.55
	ANN2	4.61	0.00	7.43	23.22	0.00	29.62	39.24	−1.00	52.66

[1]Percent error in forecasted peak flows;
[2]Time difference to peak (hours);
[3]Hydrograph volume error (%); (+ve) values indicate underestimation; (-ve) values indicate overestimation.

In the case of Kentucky basin, the SANN model is found to predict all the peak flows within an error of ±15%. The error is minimal in the case of Kolar basin also. It is worth mentioning that the error in peak flow prediction is not biased by the magnitude of the peak flow in any of these case examples. However, the SANN model delays the peak flow prediction in some cases, especially at higher lead times. It is noted from that the volume of hydrograph is well preserved by the SANN model. The maximum error (hydrograph volume) produced by the

SANN model is 4.66% in the Kolar flow series when forecasted 7 hours in advance. Note that the volume error is less than 1% in all the cases for 1-hour lead forecast (except for the hydrograph corresponding to peak flow of 32,429 m³/s).

Predictive uncertainty of SANN

The predictive uncertainty of the models is evaluated by an index called noise-to-signal ratio (NS). The unbiased SEE is a measure of the unexplained variance (Tokar and Johnson,

1999). It is usually compared with the standard deviation of the observed values of the dependent variable (STD). The ratio of SEE to STD, called the noise-to-signal ratio, indicates the degree to which noise hides the information (Gupta and Sorooshian, 1985). If the SEE is significantly smaller than STD, then the model can provide accurate predictions. On the contrary, if the ratio is greater than or equal to unity, then the model predictions will not be accurate (McCuen, 1993). The noise-to-signal ratio of SANN model in all the three case examples is presented in Fig. 6, from which it may be noted that the NS value is considerably less than unity for all the lead times in each case example, and this confirms the adequacy of SANN models.

Comparison of SANN with Traditional ANN models

The foregoing discussions clearly illustrate the potential of SANN in forecasting the river flow at higher lead times. However, the confidence in the model can be built only when its performance is compared with that of the traditional ANN models. In order to have a true comparison, the traditional ANN models of the form of Eq. (2) and Eq. (3) are built for the same case examples using same input vector (these two models are referred as ANN1 and ANN2 in this paper). The parameters of ANN1 and ANN2 are optimized using genetic algorithm.
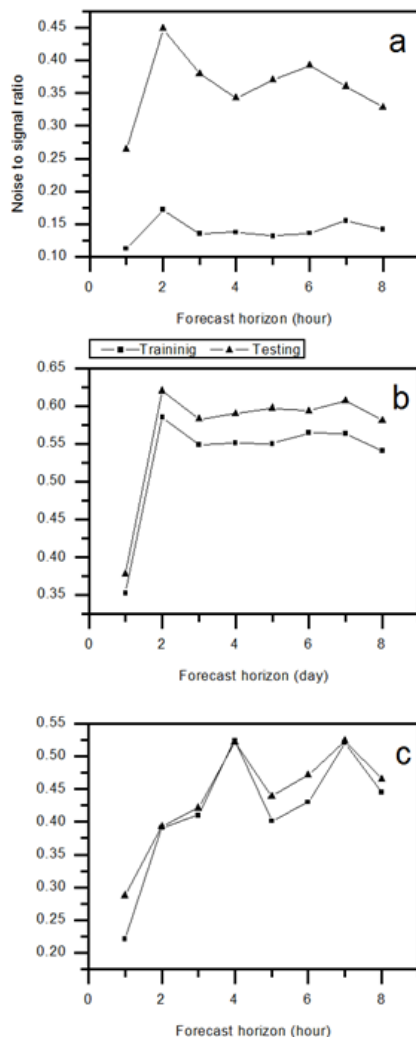

Fig. 6. Nose to signal ratio of the SANN model (a) Hypothetical series; (b) Kentucky; (c) Kolar.

The variation of efficiency along the forecast time horizon is presented in Fig. 7 for all the three models (ANN1, ANN2 and SANN). It is evident from Fig. 7 that SANN performs much better compared to the others. As expected, the ANN1 and ANN2 models show a progressive deterioration in efficiency as the lead time increases. It is noted that the performance of the three models are comparable at lower forecast time horizon: up to 2 steps ahead in the case of hypothetical series; up to 2 days in the case of Kentucky except ANN2; up to 4 hours in the case of Kolar for all models. This performance is well in line with the suggestion of Campolo et al. (1999) that the capacity of a basin to respond to a perturbation is more accurate when recent input information is used. Note that for Kentucky basin, a maximum of 2 antecedent information (runoff as well as rainfall) is available in the input vector. In the case of Kolar basin, information about the basin saturation is available in the form of antecedent rainfall information at 7-, 8- and 9-hours lag which essentially leads to the interflow component of the river flow (Chetan and Sudheer, 2006). This rainfall information, along with values of previous hour flow (up to 2 hours lag) represents the basin moisture status effectively. Hence, the ANN1 and ANN2 models show inferior performance at higher lead time. However, for SANN, the information about the basin saturation level is available in the form of additional input from a previous network structure that feeds the current flow (error corrected forecast); hence perform better at higher lead times.

The performance of these models (SANN, ANN1 and ANN2) in predicting the hydrograph characteristics are compared in Table 5 for a typical flood event (note that all the events are not presented here for brevity). It is observed that the traditional ANN models' performance is not so good as the SANN´s. It is also noted that the ANN1 and ANN2 models fail to preserve the summary statistics of the flow series in all the case examples (Fig. 8). Specifically, they fail to capture the mean and the standard deviation of the observed flow series.

The superior performance of SANN, compared to the ANN1 and ANN2 models, is evident from Fig. 9 which depicts the distribution of prediction errors during validation. It can be observed from Fig. 9 that 92% of the total hypothetical flow data are forecasted within an error of 20% by the SANN model at all lead times. Whereas for the ANN1 and ANN2 models, the corresponding values (forecasts within 20% error) were 62% and 60% respectively for 1-step ahead forecasts which is further less at higher lead times. Note that only selected lead time errors (1-step ahead, 5-steps ahead, and 8-steps ahead) are presented herein for brevity. It is worth mentioning that the distribution of errors by the SANN model in all the three case examples is almost similar at all lead times which can be attributed to the error correction implemented in the SANN model. It is noted that error distribution by the ANN1 and ANN2 models gets more flattened as the lead time increases. In the case of Kolar basin, it is to be noted that 25% of the predictions are within 1% error even at forecasts in 8 hours advance. In the case of Kentucky basin, even though the accuracy of forecasts by the SANN model is higher compared to the other two models, it is observed that the magnitude of error is high (65% of total data predicted within 20% error). While the performance of SANN is consistent in the case of Kentucky basin at all lead times, unlike the other two models, the higher magnitude of error requires further examination. One plausible reason for this could be that the dynamics of the relationship in daily time steps may be a bit complex compared to the dynamics in hourly time step.

The foregoing discussions clearly illustrate the potential of SANN model in effectively forecasting the river flow at higher

lead times. The results indicate that providing information on the saturation level of the basin in terms of the predicted values of river flow effectively helps capture the nonlinear dynamics. An interesting observation is that the model complexity is not significantly increased in the SANN model compared to ANN1 and ANN2. Further, simultaneous optimization of parameters

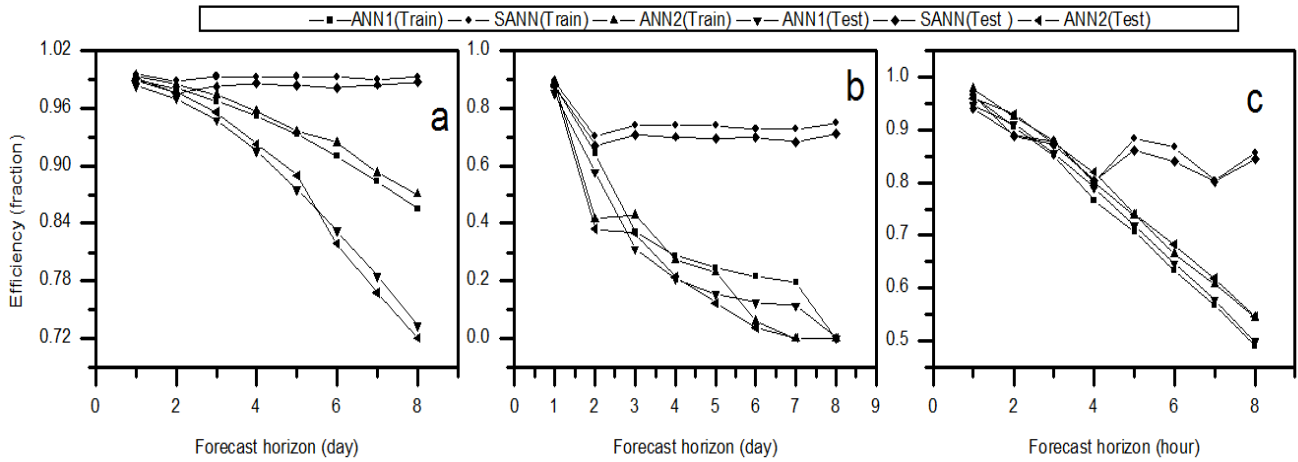helps reducing the uncertainty in the model predictions, though the training of SANN is a bit complex.

Fig. 7. Variation of efficiency along the prediction time horizon for the SANN, ANN1 and ANN2 models: (a) Hypothetical series; (b) Kentucky; (c) Kolar.
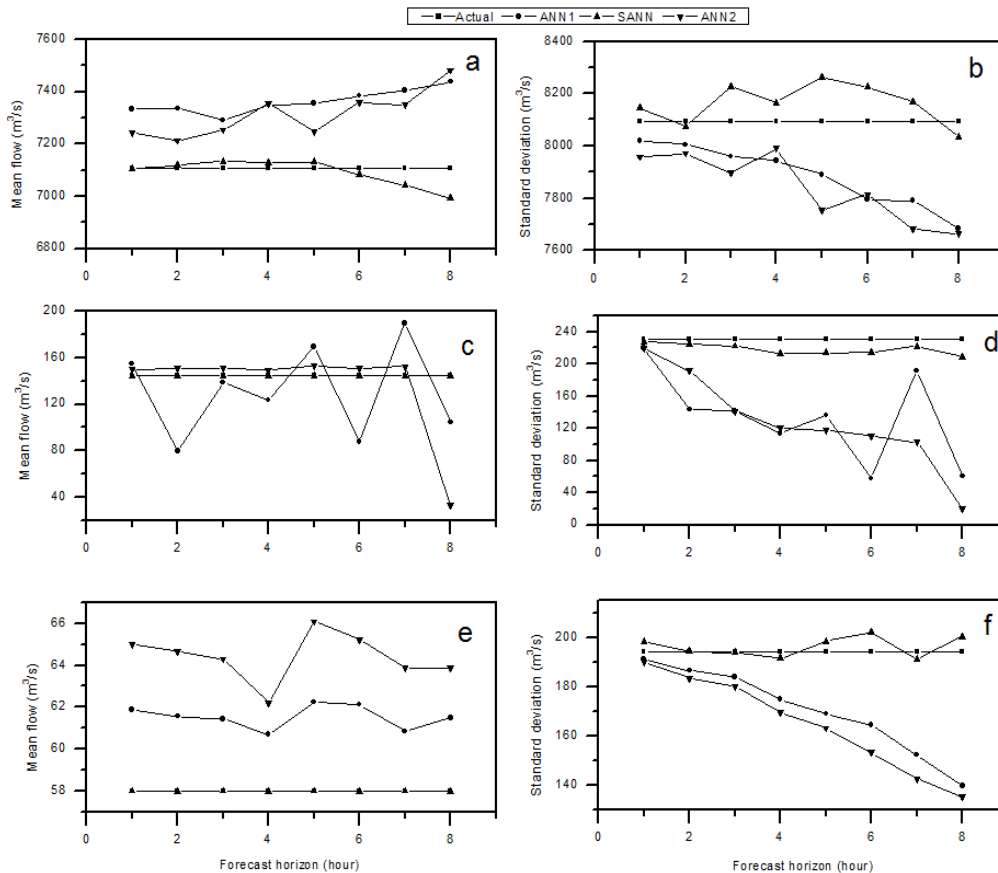
Fig. 8. Preservation of summary statistics along the forecast time horizon for the SANN, ANN1 and ANN2 models (a, b: Hypothetical series; c–d: Kentucky basin; e–f: Kolar basin).
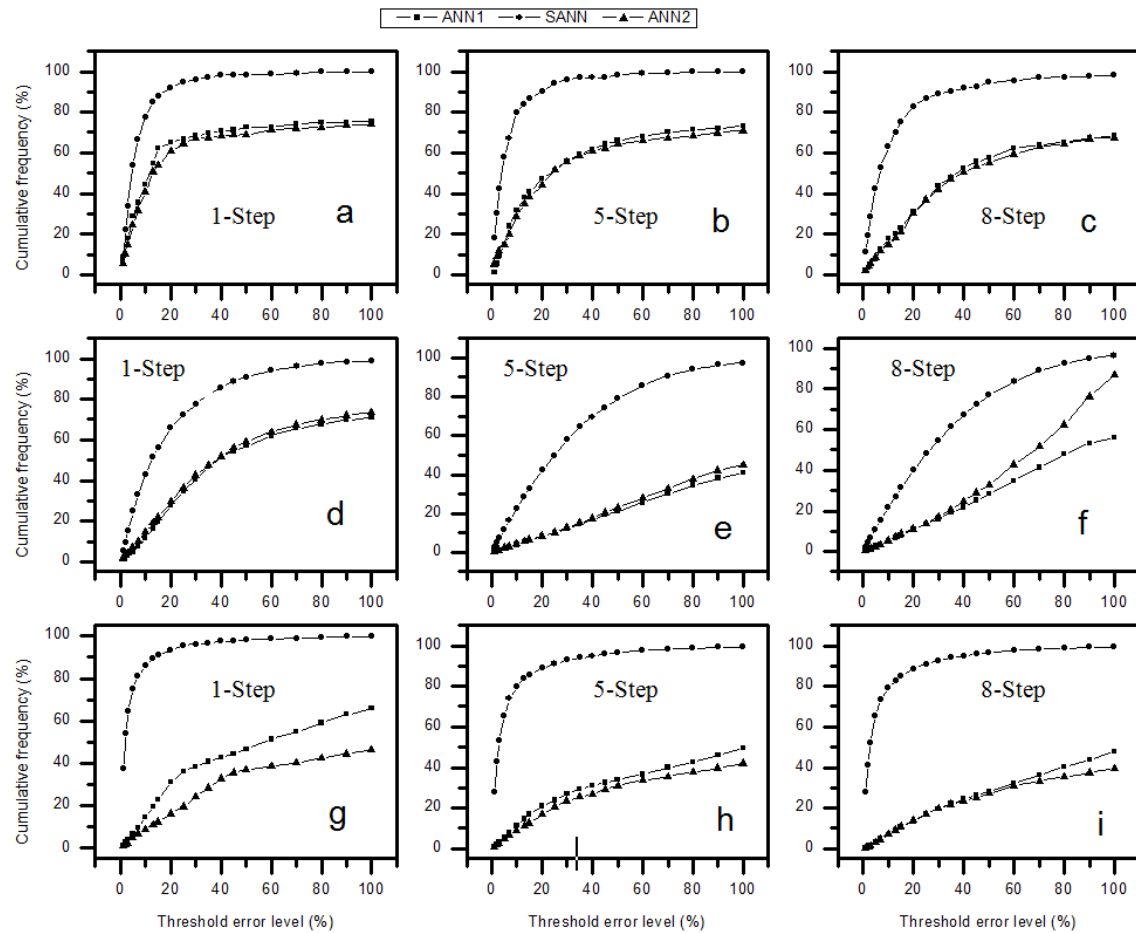
Fig. 9. Distribution of prediction errors during validation for all cases at 1, 5 and 8 steps ahead (a–c: Hypothetical series; d–f: Kentucky basin; g–i: Kolar basin).

SUMMARY AND CONCLUSIONS

A novel ANN-based modeling framework is proposed for achieving improved higher lead time flood forecasts. The objective of the paper is twofold: (i) to demonstrate the potential of the proposed computing scheme in extending the lead time of forecast; and (ii) to evaluate the relative merits and demerits of this scheme with reference to the traditional ANN modeling approaches. In the proposed model, termed as SANN, a series of ANNs are connected sequentially to extend the lead time of forecast, each of them taking a forecast value from an immediate preceding network as input. The output of each network is modified by adding an expected value of error so that the residual variance of the forecast series is minimized. The model has been applied to three case examples and the results suggest that SANN is capable of providing accurate forecasts at higher lead times (up to 8 steps ahead). A very close fit (>94% efficiency) was obtained between computed and observed flows up to 1 hour in advance for all the cases and the deterioration in fit was not significant when the lead time of forecast was increased. A comparative analysis of prediction accuracy of the SANN model and the traditional ANN models indicates that the performance of SANN is considerably better than the others. It is observed that the predictive uncertainty in the case of the SANN model is much less (in terms of noise to signal statistic) compared to the other models. The results of the study are highly encouraging and suggest that the proposed SANN app-

roach is viable for developing short-term forecasts of river flow series which can be effectively employed in developing flood management measures.

REFERENCES

Abrahart, R.J., See, L.M., Dawson, C.W., Shamseldin, A.Y., Wilby, R.L., 2010. Nearly Two Decades of Neural Network Hydrological Modeling. In: Sivakumar, B., Berndtsson, R., (Eds.): Advances in data-based approaches for hydrologic modeling and forecasting. World Scientific.

Arduino, G., Reggiani, P., Todini, E., 2005. Recent advances in flood forecasting and flood risk assessment. Hydrology and Earth System Sciences, 9, 4, 280–284.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a. Artificial neural networks in hydrology I: Preliminary concepts. J. Hydrol. Eng., 5, 2, 115–123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000b. Artificial neural networks in hydrology II: Hydrologic applications. J. Hydrol. Eng., 5, 2, 124–137.

Birkundavyi, S., Labib, R., Trung, H.T., Rousselle, J., 2002. Performance of neural networks in daily streamflow forecasting. J. Hydrol. Eng., 7, 5, 392–398.

Bowden, G.J., Dandy, G.C., Maier, H.R., 2005. Input determination for neural network models in water resources applications. Part 1-Background and methodology. J. Hydrol., 301, 1–4, 75–92.

Bruen, M., Yang, J., 2005. Functional networks in real-time flood forecasting-novel application. Advances in Water Resour., 28, 899–909.

Campolo, M., Andreussi, P., Soldati, A., 1999. River flood forecasting with neural network model. Water Resour. Res., 35, 4, 1191–1197.

Corani, G., Guariso, G., 2005. An application of pruning in the design of neural networks for real rime flood forecasting. Neural Computation and Application, 14, 66–77.

Coulibaly, P., Anctil, F., Bobee, B., 2001. Multivariate reservoir inflow forecasting using temporal neural network. J. Hydrol. Eng., 6, 5, 367–376.

Coulibaly, P., Anctil, F., Bobee, B., 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. J. Hydrol., 230, 244–257.

Coulibaly, P., 2003. Impact of metrological predictions on real-time spring flow forecasting. Hydol. Process., 17, 3791–3801.

Coulibaly, P., Hache, M., Fortin, V., Bobee, B., 2005. Improving daily reservoir inflow forecasts with model combination. J. Hydrol., 10, 2, 91–99.

Dawson, C.W., Wilby, R.L., 2001. Hydrological modelling using artificial neural networks. Progress in Physical Geography, 25, 1, 80–108.

Dawson, C.W., See, L.M., Abrahart, R.J., Heppenstall, A., 2006. Symbiotic adaptive neuro-evolution applied to rainfall–runoff modelling in northern England. Neural Networks, 19, 236–247.

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

Duan, Q., Sorooshian, S., Gupta, V., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour. Res., 28, 4, 1015–1031.

French, M.N., Krajewski, W.F., Cuykendall, R.R., 1992. Rainfall forecasting in space and time using a neural network. J. Hydrol., 137, 1–31.

Gupta, V.K., Sorooshian, S., 1985. The relationship between data and the precision of parameter estimates of hydrologic models. J. Hydrol., 81, 1–2, 57–77.

Holland, J.H., 1975. Adaptation in Natural and Artificial System. Mass. Inst. of Technol. Cambridge.

Hsu, K., Gupta, V.H., Sorroshian, S., 1995. Artificial neural network modeling of the rainfall-runoff process. Water Resour. Res., 31, 10, 2517–2530.

Chang, F.J., Chiang, Y.M., Chang, L.C., 2007. Multi-step-ahead neural networks for flood forecasting. Hydrol. Sciences-J., 52, 1, 114–130.

Chetan, M., Sudheer, K.P., 2006. A hybrid linear-neural model for river flow forecasting. Water Resour. Res., 42, 4, W04402.

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 Resour. Res., 40, 1029–2003.

Jain, A., Sudheer, K.P., Srinivasulu, S., 2004. Identification of physical processes inherent in artificial neural network rainfall runoff models. Hydrol. Process., 18, 3, 571–581.

Karunanithi, N., Grenney, W.J., Whitley, D., Bovee, K., 1994. Neural networks for river flow prediction. J. Computing in Civil Eng., 8, 2, 201–220.

Khu, S.T., Liong, S.Y., Babovic, V., Madsen, H., Muttil, N., 2001. Genetic programming and its application in real-time runoff forecasting. J. Am. Water Resour. Assoc., 37, 2, 439–451.

Kneale, P.E., See, L., Smith, A., 2001. Towards defining evaluation measures for neural network forecasting models. Proceedings of the Sixth International Conference on GeoComputation, University of Queensland, Australia, available at http://www.geocomputation.org/2001/papers/kneal.pdf (Last accessed on March 2, 2007).

Kwok, T.Y., Yeung, D.Y., 1997. Objective functions for training new hidden units in constructiveneural networks. Neural Networks, IEEE, 8, 5, 1131–1148.

Le Cun, Y., Denker, J.S., Solla, S.A., 1990. Optimal brain damage. In: Advances in Neural Information Processing Systems 2. Morgan Kaufmann, San Mateo, California, pp. 598–605.

Madsen, H., 2000. Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. J. Hydrol., 235, 276–288.

Madsen, H., Skotner, C., 2005. Adaptive state updating in real-time flow forecasting–a combined filtering and error forecasting procedure. J. Hydrol., 308, 302–312.

Madsen, H., Butts, M.B., Khu, S.T., Liong, S.Y., 2000. Data assimilation in rainfall-runoff forecasting, Hydroinformatics. 4th International Conference on Hydroinformatics, Cedar Rapids, Iowa, USA, 23–27.

Maier, H.R., Dandy, G.C., 1997. Determining inputs for neural network models of multivariate time series. Microcomput. Civ. Eng., 12, 353– 368.

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.

Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P., 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. Environmental Modelling & Software, 25, 8, 891–909.

Mathworks, 2004. Genetic algorithm and direct search tool box for use in MATLAB. The MathWorks, Inc., Natick, MA.

McCuen, R.H., 2003. Modeling hydrologic change. CRC, Boca Raton, Fla.

McCulloch, W.S., Pitts, W., 1943. A logic calculus of the ideas immanent in nervous activity. Bull. of Math. Biophys., 5, 115–133.

Moore, R.J., 1986. Advances in real-time forecasting practice. Invited paper, Symposium on Flood Warning System, Winter Meeting of the River Enginnering Section, The Institution of Water Engineers and Scientists, 23 pp.

Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models: 1. A discussion of principles. J. Hydrol., 10, 282–290.

Nayak, P.C., Sudheer, K.P., Rangan, D.M., Ramasastri, K.S., 2005. Short-term flood forecasting with a neurofuzzy model. Water Resour. Res., 41, 4, W04004.

Nayak, P.C., Sudheer, K.P., Jain, S.K., 2007. Rainfall-runoff modelling through hybrid intelligent system. Water Resour. Res. 43, 7, W07415.

Nazemi, A., Pookhadem, H.N., Mohammad, R., Akbarzadeh, T., Hosseini, S.M., 2003. Evolutionar neural network modeling for describing rainfall-runoff process. Hydrology Days, 224–235.

Pan, T.Y., Wang, R.Y., 2004. State space neural networks for short term rainfall-runoff forecasting. J. Hydrol., 297, 32–50.

Parasuraman, K., Elshorbagy, A., 2007. Cluster-Based Hydrologic Prediction Using Genetic Algorithm-Trained Neural Networks. J. Hydrol. Eng., 12, 1, 52–62.

Refsgaard, J.C., 1997. Validation and intercomparison of different updating procedures for real-time forecasting. Nordic Hydrol., 28, 65–84.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature, 323, 533–536.

Sajikumara, N., Thandaveswara, B.S., 1999. A non-linear rainfall–runoff model using an artificial neural network. J. Hydrol., 216, 32–55.

Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall runoff modelling. J. Hydrol., 199, 272–294.

Shamseldin, A.Y., O'Connor, K.M., 2001. A non-linear neural network technique for updating of river flow forecasts. Hydrology and Earth System Sciences, 5, 4, 577–597.

Sudheer, K.P., 2009. Real time flood forecasting by artificial neural network by extending the forecast lead time. Journal of Hydrological Research and Development, 24, 89–108.

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.

Sudheer, K.P., Nayak, P.C., Ramasastri, K.S., 2003. Improving peak flow estimates in artificial neural network river flow models. Hydrol. Process., 17, 3, 677–686.

Thirumalaiah, K., Deo, M.C., 2000. Hydrological forecasting using neural networks, J. Hydrol. Eng., 5, 2, 180–189.

Tokar, A.S., Johnson, A., 1999. Rainfall-runoff modeling using artifical neural networks. J. Hydrol. Eng., 4, 3, 232–239.

US West Optical Company, 1989a. HYDRODATA: USGS daily and peak flows values. Denver, Colo.

US West Optical Company, 1989b. CLIMATEDATA: Daily total rainfall and average temperature values. Denver, Colorado, USA.

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.

WMO, 1992. Simulated real-time intercomparison of hydrological models. Operational Hydrology Report No. 38, World Meteorological Organization, Ganeva.

Zealand, C.M., Burn, D.H., Simonovic, S.P., 1999. Short term streamflow forecasting using artificial neural networks. J. Hydrol., 214, 32–48.