Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Hydrology 291 (2004) 52–66
Journal of Hydrology
www.elsevier.com/locate/jhydrol

A neuro-fuzzy computing technique for modeling hydrological time series

P.C. Nayak[a,*], K.P. Sudheer[b,1], D.M. Rangan[c,2], K.S. Ramasastri[d,3]

[a]Scientist, Deltaic Regional Centre, National Institute of Hydrology, Kakinada-533-003, India
[b]Department of Civil Engineering, Indian Institute of Technology Madras, Chennai-600036, India
[c]Tech Gr. II, Deltaic Regional Centre, National Institute of Hydrology, Kakinada-533-003, India
[d]Director, National Institute of Hydrology, Roorkee-247-667, India

Received 29 October 2002; revised 10 November 2003; accepted 10 December 2003

Abstract

Intelligent computing tools such as artificial neural network (ANN) and fuzzy logic approaches are proven to be efficient when applied individually to a variety of problems. Recently there has been a growing interest in combining both these approaches, and as a result, neuro-fuzzy computing techniques have evolved. This approach has been tested and evaluated in the field of signal processing and related areas, but researchers have only begun evaluating the potential of this neuro-fuzzy hybrid approach in hydrologic modeling studies. This paper presents the application of an adaptive neuro fuzzy inference system (ANFIS) to hydrologic time series modeling, and is illustrated by an application to model the river flow of Baitarani River in Orissa state, India. An introduction to the ANFIS modeling approach is also presented. The advantage of the method is that it does not require the model structure to be known a priori, in contrast to most of the time series modeling techniques. The results showed that the ANFIS forecasted flow series preserves the statistical properties of the original flow series. The model showed good performance in terms of various statistical indices. The results are highly promising, and a comparative analysis suggests that the proposed modeling approach outperforms ANNs and other traditional time series models in terms of computational speed, forecast errors, efficiency, peak flow estimation etc. It was observed that the ANFIS model preserves the potential of the ANN approach fully, and eases the model building process.

Keywords: Neural networks; Fuzzy logic; Fuzzy inference system; Time series modeling; Hydrological modeling

1. Introduction

Time series modeling for either data generation or forecasting of hydrologic variables is an important step in the planning and operational analysis of water resources. Traditionally, autoregressive moving average (ARMA) models have been used for modeling water resource time series because such models are accepted as a standard representation of

* Corresponding author. Fax: +91-884-2350054.
E-mail addresses: nayakpc@yahoo.co.in (P.C. Nayak), sudheer@iitm.ac.in (K.P. Sudheer), ranga_mohan@yahoo.co.in (D.M. Rangan), ksr@nih.ernet.in (K.S. Ramasastri).
[1] Fax: +91-44-22578281.
[2] Fax: +91-884-2350054.
[3] Fax: +91-1332-72123.

0022-1694/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.jhydrol.2003.12.010

stochastic time series (Maier and Dandy, 1997). However, such models do not attempt to represent the non-linear dynamics inherent in the hydrologic process, and may not always perform well (Tokar and Johnson, 1999). Time series analysis requires mapping complex relationships between input(s) and output(s), since the forecasted values are mapped as a function of observed patterns in the past. Owing to the difficulties associated with non-linear model structure identification and parameter estimation, very few truly non-linear system theoretic hydrologic models have been reported (e.g. Jacoby, 1966; Amorocho and Brandstetter, 1971; Ikeda et al., 1976). In most cases, linearity or piecewise linearity has been assumed (Natale and Todini, 1976a,b). Recently a growing interest in the modeling of nonlinear relationships has developed and a variety of test procedures for detecting the nonlinearities have evolved (Anders and Korn, 1999). If the aim of analysis is prediction, however, it is not sufficient to uncover the nonlinearities. One needs to describe them through an adequate nonlinear model. Unfortunately, for many applications the theory does not guide the model building process by suggesting the relevant input variables or the correct functional form. This particular difficulty makes it attractive to consider an 'atheoretical' but flexible class of statistical models (Anders and Korn, 1999).

Artificial neural networks (ANN) are essentially semi-parametric regression estimators and are well suited for this purpose, as they can approximate virtually any (measurable) function up to an arbitrary degree of accuracy (Hornik et al., 1989). A significant advantage of the ANN approach in system modeling is that one need not have a well-defined physical relationship for systematically converting an input to an output. Rather, all that is needed for most networks is a collection of representative examples (input–output pairs) of the desired mapping. The ANN then adapts itself to reproduce the desired output when presented with training sample input. The emergence of neural network technology has provided many promising results in the field of hydrology and water resources simulation. A comprehensive review of the application of ANN to hydrology can be found in the findings of the ASCE task committee (2000a,b).

Another soft computing technique, which has very recently received attention in hydrology, is the fuzzy-rule based approach in modeling. First introduced by Zadeh (1965), fuzzy logic and fuzzy set theory are employed to describe human thinking and reasoning in a mathematical framework. Fuzzy-rule based modeling is a qualitative modeling scheme where the system behavior is described using a natural language (Sugeno and Yasukawa, 1993). The last decade has witnessed a few applications of a fuzzy logic approach in water resources forecasting (Fujita et al., 1992; Zhu and Fujita, 1994; Zhu et al., 1994; Stuber et al., 2000; See and Openshaw, 2000; Hundecha et al., 2001; Xiong et al., 2001).

These intelligent computational methods offer real advantages over conventional modeling, including the ability to handle large amounts of noisy data from dynamic and nonlinear systems, especially when the underlying physical relationships are not fully understood. Each of these techniques is proven to be effective when used on their own. However, when combined together, the individual strengths of each approach can be exploited in a synergistic manner for the construction of powerful intelligent systems. In recent years, the integration of neural networks and fuzzy logic has given birth to new research into neuro-fuzzy systems. Neuro-fuzzy systems have the potential to capture the benefits of both these fields in a single framework. Neuro-fuzzy systems eliminate the basic problem in fuzzy system design (obtaining a set of fuzzy if-then rules) by effectively using the learning capability of an ANN for automatic fuzzy if-then rule generation and parameter optimization. As a result, those systems can utilize linguistic information from the human expert as well as measured data during modeling. Such applications have been developed for signal processing, automatic control, information retrieval, database management, computer vision and data classification (e.g. Jang, 1993). However, there is little discussion in the literature of more pragmatic hydrologic applications of this hybrid computing system.

The major objective of this paper is to investigate the potential of neuro-fuzzy systems in modeling hydrologic time series and to assess its performance relative to ANN and other traditional time series modeling techniques such as ARMA. The underlying principle and the neuro-fuzzy computing architecture are also discussed. The applicability of the method is demonstrated by modeling river flow for an Indian basin.

2. Neuro-fuzzy model

Neuro-fuzzy modeling refers to the way of applying various learning techniques developed in the neural network literature to fuzzy modeling or to a fuzzy inference system (FIS). The basic structure of a FIS consists of three conceptual components: a rulebase, which contains a selection of fuzzy rules; a database which defines the membership functions (MF) used in the fuzzy rules; and a reasoning mechanism, which performs the inference procedure upon the rules to derive an output (see Fig. 1). FIS implements a nonlinear mapping from its input space to the output space. This mapping is accomplished by a number of fuzzy if-then rules, each of which describes the local behavior of the mapping. The parameters of the if-then rules (referred to as antecedents or premises in fuzzy modeling) define a fuzzy region of the input space, and the output parameters (also consequents in fuzzy modeling) specify the corresponding output. Hence, the efficiency of the FIS depends on the estimated parameters. However, the selection of the shape of the fuzzy set (described by the antecedents) corresponding to an input is not guided by any procedure (Ojala, 1995). But the rule structure of a FIS makes it possible to incorporate human expertise about the system being modeled directly into the modeling process to decide on the relevant inputs, number of MFs for each input, etc. and the corresponding numerical data for parameter estimation. In the present study, the concept of the adaptive network, which is a generalization of the common backpropagation neural network, is employed to tackle the parameter identification problem in a FIS.

An adaptive network is a multi layered feed forward structure whose overall output behavior is determined by the value of a collection of modifiable parameters. More specifically, the configuration of an adaptive network is composed of a set of nodes connected through directional links, where each node is a process unit that performs a static node function on its incoming signal to generate a single node output. The node function is a parameterized function with modifiable parameters. It may be noted that links in an adaptive network only indicate the flow direction of signals between nodes and no weights are associated with these links. Readers are referred to Brown and Harris (1994) for more details on adaptive networks. Jang (1993) introduced a novel architecture and learning procedure for the FIS that uses a neural network learning algorithm for constructing a set of fuzzy if-then rules with appropriate MFs from the stipulated input–output pairs. This procedure of developing a FIS using the framework of adaptive neural networks is called an adaptive neuro fuzzy inference system (ANFIS).

2.1. ANFIS architecture

The general structure of the ANFIS is presented in Fig. 2. Selection of the FIS is the major concern when designing an ANFIS to model a specific target system. Various types of FIS are reported in the literature (e.g. Mamdani and Assilian, 1975; Tsukamoto, 1979; Takagi and Sugeno, 1985) and each are characterized by their consequent parameters only. The current study uses the Sugeno fuzzy model (Takagi and Sugeno, 1985; Sugeno and Kang, 1988) since

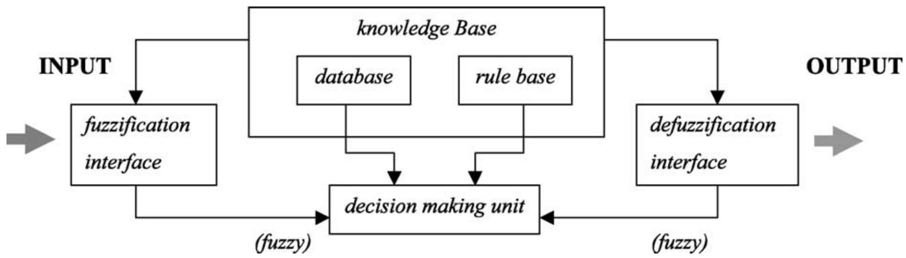

Fig. 1. Fuzzy Inference System with crisp output.

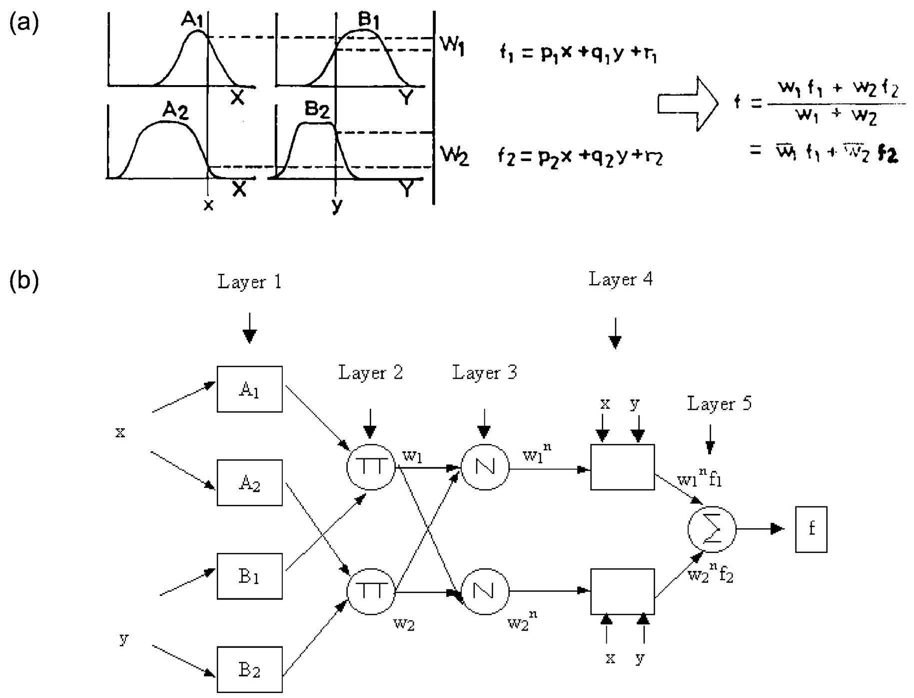

Fig. 2. (a) Fuzzy inference system. (b) Equivalent ANFIS architecture.

the consequent part of this FIS is a linear equation and the parameters can be estimated by a simple least squares error method.

For instance, consider that the FIS has two inputs x and y and one output z. For the first order Sugeno fuzzy model, a typical rule set with two fuzzy if-then rules can be expressed as:

$$\text{Rule 1: If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{then } f_1 = p_1x + q_1y + r_1 \quad (1)$$

$$\text{Rule 2: If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{then } f_2 = p_2x + q_2y + r_2 \quad (2)$$

where A_1, A_2 and B_1, B_2 are the MFs for inputs x and y, respectively; p_1, q_1, r_1 and p_2, q_2, r_2 are the parameters of the output function. Fig. 2(a) illustrates the fuzzy reasoning mechanism for this Sugeno model to derive an output function (f) from a given input vector $[x, y]$.

The corresponding equivalent ANFIS architecture is presented in Fig. 2(b), where nodes of the same layer have similar functions. The functioning of the ANFIS is as follows:

Layer 1: Each node in this layer generates membership grades of an input variable. The node output OP_i^1 is defined by:

$$\mathrm{OP}_i^1 = \mu_{A_i}(x) \text{ for } i = 1, 2 \text{ or} \quad (3)$$

$$\mathrm{OP}_i^1 = \mu_{B_{i-2}}(y) \text{ for } i = 3, 4 \quad (4)$$

where x (or y) is the input to the node; A_i (or B_{i-2}) is a fuzzy set associated with this node, characterized by the shape of the MFs in this node and can be any appropriate functions that are continuous and piecewise differentiable such as Gaussian, generalized bell shaped, trapezoidal shaped and triangular shaped functions. Assuming a generalized bell function as the MF, the output OP_i^1 can be computed as,

$$\mathrm{OP}_i^1 = \mu_{A_i}(x) = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^{2b_i}} \quad (5)$$

where $\{a_i, b_i, c_i\}$ is the parameter set that changes the shapes of the MF with maximum equal to 1 and minimum equal to 0.

Layer 2: Every node in this layer multiplies the incoming signals, denoted as $\prod$, and the output OP_i^2 that represents the firing strength of a rule is computed as,

$$\mathrm{OP}_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1, 2. \tag{6}$$

Layer 3: The *i*th node of this layer, labeled as *N*, computes the normalized firing strengths as,

$$\mathrm{OP}_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \qquad i = 1, 2 \tag{7}$$

Layer 4: Node *i* in this layer computes the contribution of the *i*th rule towards the model output, with the following node function:

$$\mathrm{OP}_i^4 = \overline{w_i}f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{8}$$

where $\overline{w}$ is the output of layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set

Layer 5: The single node in this layer computes the overall output of the ANFIS as:

$$\mathrm{OP}_1^5 = \text{Overall output} = \sum_i \overline{w_i}f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{9}$$

2.2. Estimation of parameters

The parameters for optimization in an ANFIS are the premise parameters $\{a_i, b_i, c_i\}$, which describe the shape of the MFs, and the consequent parameters $\{p_i, q_i, r_i\}$, which describe the overall output of the system. The basic learning rule of an adaptive network, the backpropagation algorithm (Rumelhart et al., 1986), which is based on the gradient descent rule, can be successfully applied to estimate these parameters. However, Jang (1991) argues that the gradient descent method is generally slow and is likely to get trapped in local minima. Jang has proposed a faster learning algorithm, which combines the gradient descent method and the least squares estimate (LSE) to identify parameters, as described below:

The adaptive network has one output and is assumed to be

$$\text{output} = F(\vec{I}, S), \tag{10}$$

where $\vec{I}$ is the set of input variables and *S* is the set of parameters. If there exists a function *H* such that the composite function $H \circ F$ is linear in some of the elements of *S*, then these elements can be identified by the least squares method. More formally, if the parameter set *S* can be decomposed into two sets

$$S = S_1 \oplus S_2, \tag{11}$$

(where $\oplus$ represents the direct sum) such that $H \circ F$ is linear in the element S_2, then applying *H* to Eq. (10), we have

$$H(\text{output}) = H \circ F(\vec{I}, S) \tag{12}$$

which is linear in the elements of S_2. Now given values of elements of S_1, the *P* training data can be plugged into Eq. (12) to obtain the matrix equation:

$$AX = B \tag{13}$$

where *X* is the unknown vector whose elements are parameters in S_2. Let $|S_2| = M$, then the dimensions of *A*, *X* and *B* are $P \times M$, $M \times 1$ and $P \times 1$, respectively. Since *P* (number of training data pairs) is usually greater than *M* (number of linear parameters), this is an over-determined problem and generally there is no exact solution to Eq. (13). However, a LSE of *X* can be sought that minimizes the squared error $\|AX - B\|^2$.

From the ANFIS architecture presented in Fig. 2 it is observed that given the values of the premise parameters, the overall output can be expressed as linear combinations of consequent parameters. More precisely, the output *f* can be rewritten as,

$$\begin{aligned} f &= \overline{w_1}f_1 + \overline{w_2}f_2 \\ &= (\overline{w_1}x)p_1 + \overline{(w_1}y)q + (\overline{w_1})r_1 + (\overline{w_2}x)p_2 \\ &\quad + \overline{(w_2}y)q_2 + (\overline{w_2})r_2 \end{aligned} \tag{14}$$

which is linear in the consequent parameters (p_1, q_1, r_1, p_2, q_2 and r_2). As a result, the total number of parameters (*S*) in an ANFIS can be divided into two such that S_1 = set of premise parameters and S_2 = set of consequent parameters. Consequently the hybrid-learning algorithm, which combines the backpropagation gradient descent and least squares method, can be used for an effective search of the optimal parameters of the ANFIS. More specifically, in the forward pass of the hybrid learning algorithm, the node output goes forward until layer 4 and the consequent parameters are identified by the least squares method. In the backward pass, the error signal propagates backwards and the premise parameters are updated by gradient descent. As mentioned earlier, the consequent parameters thus identified are optimal under the condition that the premise parameters are fixed. Accordingly, the hybrid approach converges much faster since it reduces the dimension of the search space of the original back-propagation method. A detailed description of this algorithm can be found in Jang and Sun (1995).

2.3. Defuzzification

It may be noted that the basic ANFIS takes either fuzzy inputs or crisp inputs, but the overall outputs are fuzzy sets. Therefore, a defuzzification strategy is needed to convert a fuzzy set to a crisp value. The crisp output is generally obtained using different defuzzification strategies (Brown and Harris, 1994). The Takagi-Sugeno approach (Takagi and Sugeno, 1985) that is used in the current investigation, however, does not have an explicit defuzzification procedure (Xiong et al., 2001), or rather, it amalgamates two procedures, the logic decision and defuzzification procedures into one composite procedure.

2.3.1. Application of ANFIS to time series modeling

Time series modeling is fundamentally different from the conceptual modeling and simulation of systems in several aspects. Although a time series can be interpreted as an output of a system, it is, by definition, an output of an unknown system (Lopez et al., 1996). Neither the system characteristics nor the input functions are known a priori, and consequently time series analysis must content itself with estimating future output values by means of extrapolation from their own past. This can be addressed in two ways: by using the available information to generate a time series that preserve the statistical properties of the original series or by using known values at each time step to make a forecast in more of an updating approach. The current analysis considered the latter approach. In the present study the potential of an ANFIS model has been investigated for time series modeling of river flow of Baitarani river basin (Fig. 3) in Orissa state of India. The River Baitarani, one of the major rivers of the Orissa State in India, drains an area of 14,218 km^2 to the Bay of Bengal. The average annual rainfall is 1187 mm. The rainfall received in the basin is mainly from the south-west monsoon and lasts from June to October. Nearly 80% of the annual precipitation occurs during these months and it brings heavy flow and creates havoc in lower reaches during the monsoon season. This necessitates an efficient flood forecasting system in the basin. Sudheer et al. (2000) have done a comprehensive study on developing a flood forecasting system for the basin and they have developed an ANN model for this purpose. They have reported that there is room for further improving the forecasts as the ANN was not able to match the peak flows effectively. This provided an impetus to develop an ANFIS model for the basin and investigate its potential. Daily values of flow for a continuous period of 24 years (1972–1995) at Anandapur gauging site were available and have been used in the current study.

3. Model development and testing

There are no fixed rules for developing an ANFIS, even though a general framework can be followed based on previous successful applications in engineering. The goal of an ANFIS is to generalize a relationship of the form:

$$Y^m = f(X^n) \tag{15}$$

where X^n is an n-dimensional input vector consisting of variables $x_1, \ldots, x_i, \ldots, x_n$; Y^m is an m-dimensional output vector consisting of the resulting variables of interest $y_1, \ldots, y_i, \ldots, y_m$. In the flow modeling, values of x_i may include runoff and any other exogenous variables at various time lags, and the value of y_i is generally the flow during subsequent periods. However, the number of antecedent values to include in the vector X^n is not known a priori. A firm understanding of the hydrologic system under consideration plays an important role in the successful implementation of ANFIS. This helps in avoiding loss of information that may result if key input variables are omitted, and also prevents inclusion of spurious input variables that tend to confuse the training process.

In the ANFIS development the selection of appropriate input variables is important since it

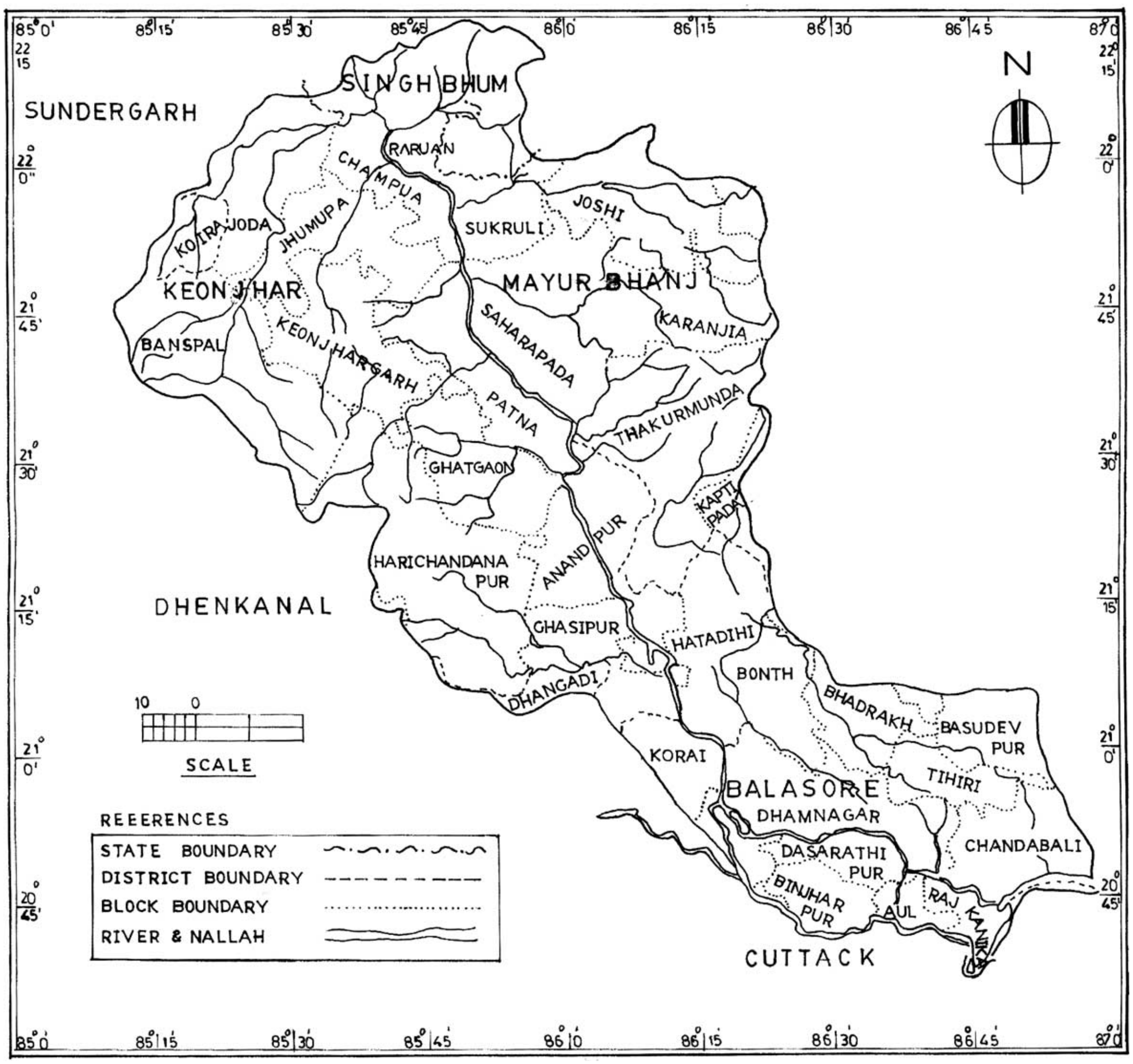

Fig. 3. Basin map of the river Baitarani, India.

provides the basic information about the system being modeled. In addition to the antecedent inflow values, exogenous input variables such as precipitation, evaporation etc. might have an influence on daily river flows. However, in the current study, the problem has been addressed in a univariate time series approach without exogenous input variables. The parameters that need to be selected in the input vector, hence, are the number of runoff values at different time lags that can best represent the time series by an ANFIS model. Determining the number of runoff values involves finding the lags of runoff that have significant influence on the predicted flow. Sudheer et al. (2002a) suggested a statistical procedure for identifying the appropriate input vector for a model. However, their approach, which is based on cross-, auto-, and partial auto-correlation properties of the series, relies on the linear relationship between the variables, and the effect of an additional variable to capture any nonlinear residual dependencies is not assessed in the procedure. Consequently, the current study analyzed different combinations of antecedent flow values and the appropriate input vector has been selected based on the analysis of residuals. The analysis started with one antecedent flow in the input vector and an ANFIS model is constructed. The input vector is then modified by successively adding flow at one more time lag, and a new ANFIS model is developed each time. The goodness of fit statistics are computed during training and validation for each ANFIS model, and the best model is selected based on the analysis of residuals.

Six ANFIS models were developed during the analysis with the corresponding input vectors as follows:

Model 1	$x(t) = f(x[t-1])$
Model 2	$x(t) = f(x[t-1]x[t-2])$
Model 3	$x(t) = f(x[t-1]x[t-2]x[t-3])$
Model 4	$x(t) = f(x[t-1]x[t-2]x[t-3]x[t-4])$
Model 5	$x(t) = f(x[t-1]x[t-2]x[t-3]x[t-4]$ $x[t-5])$
Model 6	$x(t) = f(x[t-1]x[t-2]x[t-3]x[t-4]$ $x[t-5]x[t-6])$

where $x(t)$ corresponds to the river flow at time t.

The number of MFs assigned to each input of the ANFIS was initially set to two. The input data are scaled so as to lie in the range of zero to one as suggested by Masters (1993), since the MFs of the ANFIS takes values between zero and one. The values of root mean squared error (RMSE) are used here as the index to check the ability of a model. However, a model with a minimum RMSE may not be sufficient to eliminate the uncertainty in model structure choice (Sudheer et al., 2002b). Therefore, an alternative model selection method called 'cross validation' (Stone, 1974; Eforn and Tibshirani, 1993) is employed for model selection. It may be noted that the uncertainty is not completely removed by cross validation. The motivation of this model selection procedure is the principle that adding model complexity need not result in a better description of an underlying function due to increasing estimation errors. In order to find an approximate degree of complexity, it is appealing to compare the prediction errors of different model specifications. Such prediction errors are obtained by dividing the sample into M subsets ($M = 4$ in the present case), which contain n observations each. The model is repeatedly re-estimated, leaving out one different subset each time. The average RMSE on the M subsets that have been left out defines the cross validation error, and the model structure that gives the minimum value for the error is considered to be the best fit.

The best fit model identified was trained using 6, 12 and 18 years of data and validated using the data for the period 1990–1995, to assess the influence of length of training data on the generalization properties of the model. Various MFs for the ANFIS structure were used to demonstrate the effect of choice of MF on the model performance. The resulting hydrograph was analyzed statistically using various performance indices. The goodness of fit statistics considered are the RMSE between the computed and observed runoff, the coefficient of correlation (CORR) and the model efficiency (EFF).

In most traditional statistical models, the data have to be normally distributed before the model coefficients can be estimated efficiently. If the data are not normally distributed, suitable transformations to normality have to be applied. However, Raman and Sunilkumar (1995) reported that good results could be obtained by using real world observations directly (i.e. without normalizing and standardizing the data prior to input). Until recently many researchers, when developing soft computing models, have not addressed this issue. Fortin et al. (1997) argue that it is unlikely that the model can account for the trends in the data, since they are unable to extrapolate beyond the range of data used for training as reported by Minns and Hall (1996). However, this heuristic is not confirmed by empirical trials (Faraway and Chatfield, 1998). In the current application, this issue is investigated by comparing the models developed on transformed (into normal domain) and non-transformed data prior to input to the models.

4. Results and discussions

The average cross validation error for different model structures considered is presented in Table 1. It may be noted that the results presented in Table 1 are for the models trained using raw data (non-transformed, but scaled). From Table 1, it is apparent that all of the models performed similarly as the average cross validation RMSE does not vary significantly. However, the models showed significant variations in the average cross validation

Table 1
Values of cross validation errors and efficiency for different ANFIS models

	Period of the left-out subset during training	Model number					
		1	2	3	4	5	6
RMSE	1972–1977	0.0303	0.0294	0.0303	0.0389	0.0392	0.0386
	1978–1983	0.0159	0.0161	0.016	0.0161	0.0156	0.0151
	1984–1989	0.0297	0.0287	0.0297	0.0356	0.0359	0.0352
	1990–1995	0.0187	0.0181	0.0183	0.0183	0.0185	0.0184
	Average	0.0237	0.0231	0.0236	0.0272	0.0273	0.0268
EFF	1972–1977	49.97	52.77	49.88	17.56	21.56	18.34
	1978–1983	61.06	59.75	60.4	60.05	59.53	56.34
	1984–1989	51.03	54.45	51.13	29.91	31.23	30.58
	1990–1995	68.45	70.54	69.78	69.85	68.34	68.21
	Average	57.63	59.38	57.80	44.34	45.17	43.37

efficiency (a minimum of 43.37% for Model 6 to a maximum of 59.38% for Model 2). Model 2, which consists of two antecedent flows in input, showed the highest efficiency and the minimum RMSE, and it is selected as the best-fit model for describing the time series of river flow in the Baitarani basin. The trained Model 2 consists of four fuzzy if-then rules that employed the fuzzy intersection operator.

The goodness of fit statistics on the un-scaled flow values, during training as well as validation, for all the models that are trained using 18 years of data (1972–1989), are presented in Table 2. The RMSE statistic is a measure of residual variance and is indicative of the model's ability to predict high flows. Considering the magnitude of the peak flow during the period of study (10339 m^3/s), the ANFIS models were able to compute high flows with reasonable accuracy, as can be evidenced by the low RMSE values (see Table 2). However it is worth noting that all the models have poor efficiency during training, suggesting a large amount of unexplained variance for all the models. These poor efficiencies indicate that an ANFIS model's prediction away from the mean would not be accurate. However, all the models show good efficiency during validation, indicating good generalization properties for the ANFIS models.

The annual mean and standard deviation of the historic flows during the period of study is shown in Fig. 4. It is evident from Fig. 4 that the flow during the period of study is highly varying, as there are wide fluctuations in the mean and standard deviation. It is observed that in the training data set corresponding to the period 1972–1989, the number of patterns containing high flow are significantly less than the number of patterns containing low and medium flows, and therefore the parameters estimated based on this training set have a bias towards the low and medium flow. It can be observed that the period 1990–1995 consists of relatively low and medium flows, and consequently all the models exhibited good performance during validation. It can be observed that the flows are relatively high during the periods 1972–1977 and 1983–1989. This observation substantiates the poor performance of the models that are validated on this period during cross validation (see Table 1), as these high flows are not contained in the training set. The performance indices for Model 2, trained using 6-, 12-, and 18 years of data are presented in Table 3. The results indicate that Model 2, trained using only 6 years (1984–1989) of data,

Table 2
Goodness of fit statistics during training and validation period

	Training			Validation		
	RMSE	EFF	CORR	RMSE	EFF	CORR
Model 1	281.77	50.73	0.7123	194.28	68.95	0.8274
Model 2	272.01	54.09	0.7355	187.73	70.55	0.8401
Model 3	270.11	54.73	0.7398	190.22	69.78	0.8354
Model 4	268.61	55.23	0.7432	189.89	69.89	0.8361
Model 5	278.62	51.03	0.7056	192.65	67.56	0.8245
Model 6	279.51	51.68	0.7121	193.52	68.12	0.8312

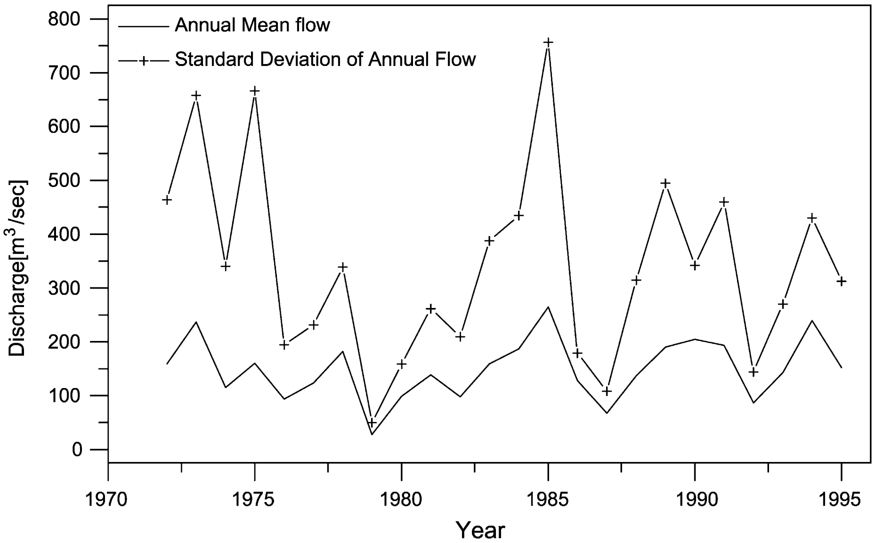

Fig. 4. Mean and standard deviation of annual flow of Baitarani river, India.

shows high RMSE (294.81 m^3/s) during training. It may be noted that the period 1984–1989 contains the maximum peak flows during the entire study period, as stated earlier. On the contrary, when this period is not considered for training (during cross validation), the model shows training RMSE of only 238.09 m^3/s implying the model's incapability to reproduce high flows. Nevertheless, the performance of Model 2, trained on different data lengths is comparable during validation. It is observed that the relative variation of the performance indices is not significant for ANFIS models with different MFs. Moreover, no significant change is observed during validation of the models (a change of 0.33% in efficiency and 0.987 m^3/s in RMSE during calibration). It is observed that by increasing the number of MFs assigned to each input to the ANFIS, the model performance does not improve; on the contrary it increases the model complexity and parsimony. The foregoing discussions clearly illustrate the potential of ANFIS in time series modeling of the river flow. However, it is observed that unless carefully trained the ANFIS model's performance may not be satisfactory.

4.1. Effect of transforming the data on model performance

A significant observation from the discussed results is that ANFIS models are able to explain only 50–60% of the original variance. It is further observed that the high flow periods in the training data set significantly affects the performance of the model. One of the reasons for this may be that the series is

Table 3
Effect on length of calibration data on the performance of ANFIS

Training period	Calibration			Validation (1990–1995)		
	RMSE	EFF	CORR	RMSE	EFF	CORR
6 years (1984–1989)	294.81	52.48	0.74	186.69	70.46	0.74
12 years (1978–1989)	273.70	53.42	0.74	188.55	70.27	0.76
18 years (1972–1989)	272.01	54.09	0.74	187.73	70.55	0.84

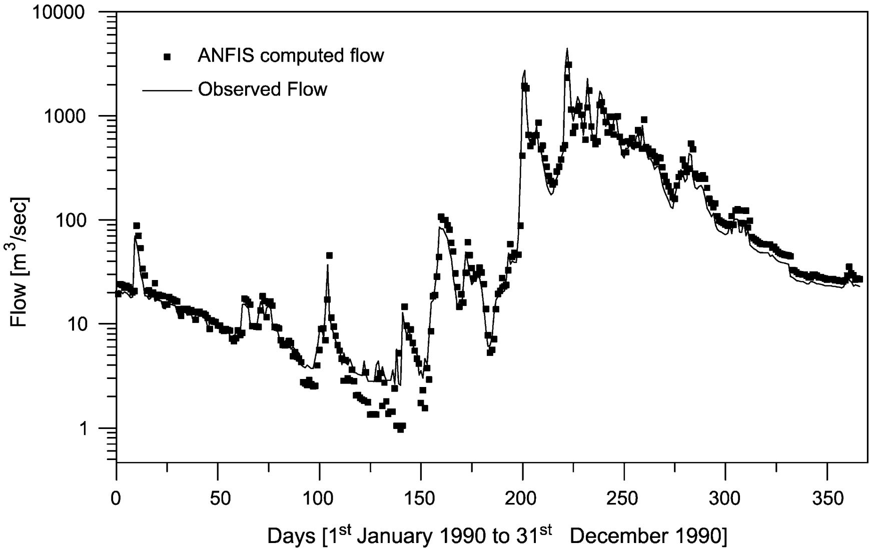

Fig. 5. Computed and observed hydrograph during validation period (non-transformed data).

highly skewed and shows heteroscedasticity, and the deterministic components in the time series data are not removed prior to modeling. The skewness coefficient of the historic data considered in the study is 9.45. Therefore, a transformation to reduce this skewness closer to zero is carried out. Different transformations are made to the original flow series and the Wilson-Hilferty transformation is found to fit well to the given data on the basis of chi-square statistics for different distributions. The procedure for a Wilson-Hilferty transformation can be obtained from Salas et al. (1985).

The Model 2 is re-trained using the transformed flow series for the period 1972–1989 and validated

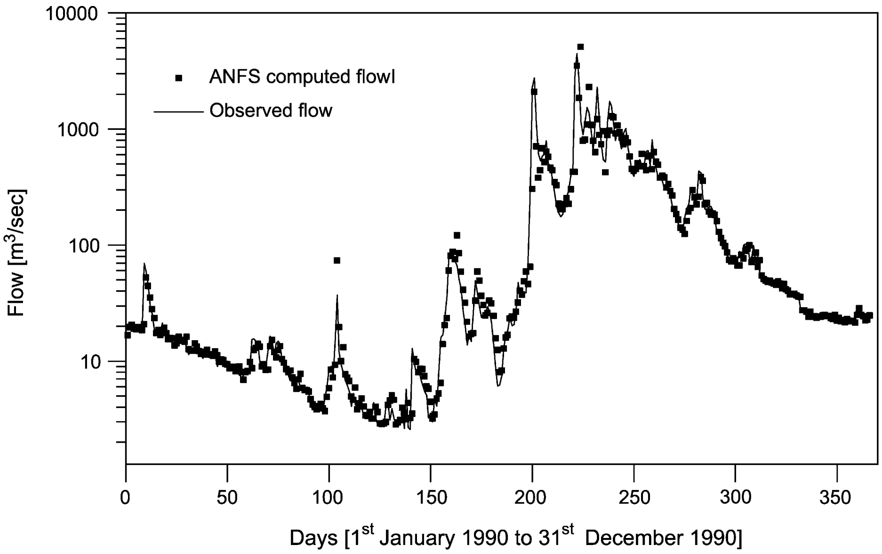

Fig. 6. Computed and observed hydrograph during validation period (transformed data).

using the data for the period 1990–1995. It is observed that the model performance is significantly improved by using transformed data for training. The model showed a training efficiency of 81.95% (an increase of 27.86% from model trained with non-transformed data), and a validation efficiency of 81.55% (a gain of 11%). The correlation coefficient between the computed and observed flow series (transformed) are found to be 0.9053 during training as well as validation. However, the RMSE statistic is slightly deteriorated by the transformation, but was not significant (worsened from 272.01 to 279.02 m^3/s). The Model 2 trained with transformed data resulted in an explained variance of more than 80%. It is observed that the model performance is consistent during training and validation implying the elimination of effect of magnitude of flows. The resulting hydrographs from Model 2 trained using non-transformed and transformed data are presented in Figs. 5 and 6 respectively. While both the figures clearly illustrate the performance of the ANFIS model, Fig. 6 indicates the improved performance of the model compared to a model with non-transformed data. These results indicate that the performance of an ANFIS model is improved if the data used for its development are normally distributed. The improved performance may have resulted because the mean square error function is used to optimize the parameters of the ANFIS structure. However, more studies using transformed data may be required to reinforce this conclusion.

5. Comparison with other models

In order to assess the ability of ANFIS models relative to that of a neural network model, an ANN model is constructed using the same input parameters to the ANFIS model 2. The Wilson-Hilferty transformed flow series is used for training and validation of ANN. A standard back propagation algorithm is employed for training, and the hidden neurons are optimized by trial and error. The final ANN architecture consists of 2 hidden neurons. The performances of ANN and ANFIS in terms of the performance indices are presented in Table 4. The ANN model is trained using the same training data set as used for the ANFIS, to enable a direct comparison.

Table 4
Performance indices for ANFIS, ANN and ARMA models

Model	Calibration			Validation		
	RMSE	EFF	CORR	RMSE	EFF	CORR
ANFIS	0.4362	81.95	0.9053	0.4225	81.55	0.9032
ANN	0.4376	81.83	0.9046	0.4243	81.39	0.9025
ARMA	0.7562	80.96	0.9022	0.7782	80.70	0.9012

To have a true evaluation of the potential of ANFIS compared to traditional time series models the performance of the ANFIS model has also been compared with that of an ARMA model, and is presented in Table 4. The details of the ARMA model can be seen in Sudheer et al. (2002a).

Table 4 suggests that though the performance of both the ANFIS and the ANN models are similar during training as well as validation, the ANFIS shows a slight improvement over the ANN. It is evident from Table 4 that the ANFIS outperforms the ARMA model in terms of all performance indices. A significant improvement is observed for the ANFIS in the peak flow prediction compared to ANN. Although both models underestimated the peak flow, the ANFIS underestimated it by 11.71% as opposed to 34.26% for the ANN. It may be noted that the ANFIS built on non-transformed data underestimated the peak flow by 34.50%, and it follows that the transformation of data into the normal domain prior to model development also helps improve peak flow estimation.

It appears that while assessing the performance of any model for its applicability in forecasting streamflows, it is not only important to evaluate the average prediction error but also the distribution of prediction errors. The statistical performance evaluation criteria employed so far in this study are global statistics and do not provide any information on the distribution of errors. Therefore, in order to test the robustness of the model developed, it is important to test the model using some other performance evaluation criteria such as average absolute relative error (AARE) and threshold statistics (TS) (Jain and Indurthy, 2003). The AARE and TS not only give the performance index in terms of predicting flows but also the distribution of the prediction errors.

These criteria can be computed as:

$$\mathrm{AARE} = \frac{1}{n}\sum_{i=1}^{n} |\mathrm{RE}_t| \qquad \text{in which,}$$

$$\mathrm{RE}_t = \frac{Q_t^o - Q_t^c}{Q_t^o} 100 \tag{16}$$

where RE_t is the relative error in forecast at time t expressed as percentage, Q_t^o is the observed streamflow at time t, and Q_t^c is the computed streamflow at time t, and n is the total number of testing patterns. Clearly the smaller the value of *AARE* is, the better the performance.

The TS for a level of $x\%$ is a measure of the consistency in forecasting errors from a particular model. The TS are represented as TS_x and expressed as a percentage. This criterion can be expressed for different levels of absolute relative error from the model. It is computed for the $x\%$ level (TL) as:

$$\mathrm{TS}_x = \frac{Y_x}{n} 100 \tag{17}$$

where $Y_{x,}$ is the number of computed streamflows (out of n total computed) for which absolute relative error is less than $x\%$ from the model.

The AARE for the ANFIS model is significantly lower (18.54%) compared to the ANN (30.86%) during validation. Hence, an improved AARE without significant reduction in global evaluation statistics certainly suggests the potential of the ANFIS compared to the ANN. In the case of the ANFIS developed on non-transformed data the AARE is found to be 31.27% and reinforces the earlier considerations that a data transformation improves the ANFIS model performance. Fig. 7 shows the distribution of errors at different threshold levels for both the ANFIS and ANN models. It can be observed from Fig. 7 that about 70% of the forecasted values are within the 20% error level for the ANFIS as opposed to 30% for the ANN and this clearly illustrates the improved performance of the ANFIS over the ANN.

It may be noted that for an ANN model, the modeler has to perform a trial and error procedure to develop the optimal network architecture, while such a procedure is not required in developing an ANFIS model. Another feature, which makes the ANFIS superior to the ANN, is the number of training epochs required for convergence. In the current study, the ANFIS model reached convergence in just 20 epochs, while the ANN model took more than 300 epochs, implying considerable savings in computational time for ANFIS models. The results suggest that the model building process can be simplified when an ANFIS model is developed compared to an ANN, and the ANFIS model preserves the full potential of ANN models in its performance.

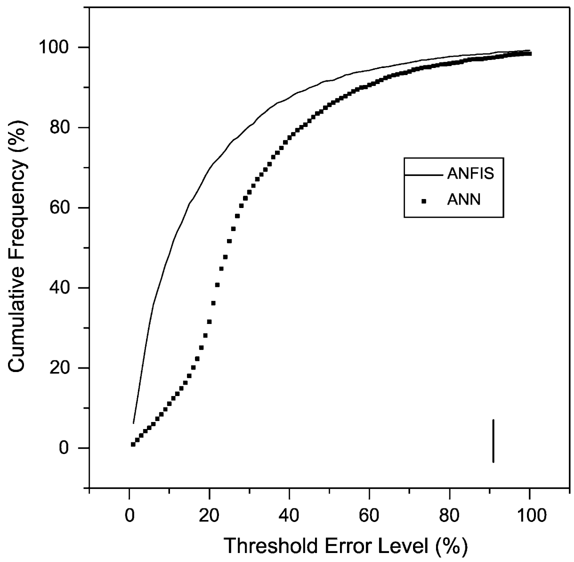

Fig. 7. Distribution of forecast error for ANFIS and ANN models.

6. Summary and conclusions

In this study, the potential of neuro-fuzzy computing techniques for modeling hydrological time series is investigated by developing an ANFIS model for the river flow of Baitarani basin. An appropriate ANN model is developed for the same basin for the purpose of comparing the performance of the ANFIS and ANN models. It is observed that the ANFIS model is capable of preserving the statistical properties of the time series. However, unless carefully trained, the model might show poor performance. It is found that the ANFIS also outperforms traditional ARMA models. The ANFIS preserves the full potential of ANN models, and simplifies the model building process. While global evaluation measures were comparable for both the ANFIS and the ANN model, the analyses suggest that the ANFIS outperforms the ANN in terms of distribution of errors. The trial and error procedure for identifying an appropriate ANN architecture is eliminated by the ANFIS model, and the ANFIS models save considerable computational time. The analyses of the results indicate that the performance of ANFIS models is significantly improved if the input data are transformed into the normal domain prior to model building. The results of the study are highly encouraging and suggest that an adaptive neuro-fuzzy approach is viable for modeling river flow series.

References

Amorocho, J., Brandstetter, A., 1971. A critique of current methods of hydrologic systems investigations. Eos Transactions of AGU 45, 307–321.

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

ASCE Task Committee, 2000a. Artificial neural networks in hydrology-I: Preliminary concepts. Journal of Hydrologic Engineering, ASCE 5 (2), 115–123.

ASCE Task Committee, 2000b. Artificial neural networks in hydrology-II: Hydrologic applications. Journal of Hydrologic Engineering, ASCE 5 (2), 124–137.

Brown, M., Harris, C., 1994. Neurofuzzy Adaptive Modeling and Control. Prentice Hall.

Eforn, B., Tibshirani, R.J., 1993. An Introduction to the Bootstrap. Chapman and Hall, London.

Faraway, J., Chatfield, C., 1998. Time series forecasting with neural networks: a comparative study using the airline data. Applied Statistics 47 (Part 2), 231–250.

Fortin, V., Quarda, T.B.M.J., Bobee, B., 1997. Comments on the use of artificial neural networks for the prediction of water quality parameters by H.R. Maier and G.C. dandy. Water Resources Research 33 (10), 2423–2424.

Fujita, M., Zhu, M.-L., Nakoa, T., Ishi, C., 1992. An application of fuzzy set theory to runoff prediction. Proceedings of the Sixth IAHR International Symposium on Stochastic Hydraulics Taipei, Taiwan, 727–734.

Hornik, K., Stichcombe, M., White, H., 1989. Multi layer feed forward networks are universal approximators. Neural Networks 2, 359–366.

Hundecha, Y., Bardossy, A., Theisen, H.-W., 2001. Development of a fuzzy logic based rainfall-runoff model. Hydrological Sciences Journal 46 (3), 363–377.

Ikeda, S., Ochiai, M., Sawaragi, Y., 1976. Sequential GMDH algorithm and its applications to river flow prediction. IEEE transactions of system management and cybernetics 6 (7), 473–479.

Jacoby, S.L.S., 1966. A mathematical model for non-linear hydrologic systems. Journal of Geophysics Research 71 (20), 4811–4824.

Jain, A., Indurthy, S.K.V.P., 2003. Comparative analysis of event based rainfall-runoff modeling techniques-deterministic, statistical, and artificial neural networks. Journal of Hydrologic Engineering ASCE 8 (2), 93–98.

Jang, J.-S.R., 1991. Rule extraction using generalized neural networks. In Proceedings of the fourth IFSA World Congress 4, 82–86.Volume for Artificial Intelligence.

Jang, J.-S.R., 1993. ANFIS: adaptive network based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23 (3), 665–683.

Jang, J.-S.R., Sun, C.-T., 1995. Neuro-fuzzy modeling and control. Proceedings IEEE 83 (3), 378–406.

López, J., Cembrano, G., Cellier, F.E., 1996. Time Series Prediction Using Fuzzy Inductive Reasoning: A Case Study, Proc. ESM'96, European Simulation Multi Conference, Budapest, Hungary, pp. 765–770.

Maier, H.R., Dandy, G.C., 1997. Determining inputs for neural network models of multivariate time series. Microcomputers in Civil Engineering 12, 353–368.

Mamdani, E.H., Assilian, S., 1975. An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7 (1), 1–13.

Masters, T., 1993. Practical Neural Networks Recipes C++. Academic Press, San Diego.

Minns, A.W., Hall, M.J., 1996. Artificial Neural networks as Rainfall-Runoff models. Hydrological Sciences Journal 41 (3), 399–417.

Natale, L., Todini, E., 1976a. A stable estimator for linear models, 1, theoretical development and Monte Carlo experiments. Water Resources Research 12 (4), 664–671.

Natale, L., Todini, E., 1976b. A stable estimator for linear models, 2. Real world hydrologic applications. Water Resources Research 12 (4), 672–676.

Ojala, T., 1995 Neuro-Fuzzy systems in control. M Sc. Thesis, Tempere Univeristy of Technology, Tampere, Finland.

Raman, H., Sunilkumar, N., 1995. Multivariate modeling of water resources time series using artificial neural networks. Journal of hydrological sciences 40, 145–163.

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

Salas, J.D., Tabios, Q.V., Bartolini, P., 1985. Approaches to multivariate modeling of water resources time series. Water Resources Bullettin 21 (4), 683–708.

See, L., Openshaw, S., 2000. Applying soft computing approaches to river level forecasting. Hydrological Sciences Journal 44 (5), 763–779.

Stone, M., 1974. Cross validation choice and assessment of statistical predictions. Journal of Royal Statistical Society, B 36, 44–47.

Stuber, M., Gemmar, P., Greving, M., 2000 Machine supported development of fuzzy-flood forecast systems, European Conference on Advances in Flood Research, Potsdam, PIK report

Nr.65, Axel Bronstert, Christine Bismuth, Lucas Menzel (Ed.), Reprint of Proceedings, 2, 504-515.

Sudheer, K.P., Nayak, P.C., Rangan, D.M., 2000. Rainfall runoff modeling using artificial neural network technique, Report No. CS/AR-16/1999-2000, National Institute of Hydrology, Roorkee, India.

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

Sudheer, K.P., Gosain, A.K., Rangan, D.M., Saheb, S.M., 2002b. Modeling evaporation using artificial neural network algorithm. Hydrological Processes 16, 3189–3203.

Sugeno, M., Yasukawa, T., 1993. A fuzzy-logic based approach to qualitative modeling. IEEE Transactions On Fuzzy Systems 1 (1), 7–31.

Sugeno, M., Kang, G.T., 1988. Structure identification of fuzzy model. Fuzzy Sets and Systems 28, 15–33.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15 (1), 116–132.

Tokar, A.S., Johnson, P.A., 1999. Rainfall runoff modeling using artificial neural network. Journal of Hydrologic Engineering, ASCE 4 (3), 232–239.

Tsukamoto, Y., 1979. An approach to fuzzy reasoning method. In: Gupta, M.M., Ragade, R.K., Yager, R.R. (Eds.), Advances in Fuzzy Set Theory and Application, North-Holland, Amsterdam, pp. 137–149.

Xiong, L.H., Shamseldin, A.Y., O'Connor, K.M., 2001. A nonlinear combination of the forecasts of rainfall-runoff models by the first order Takagi-Sugeno fuzzy system. Journal of Hydrology 245 (1–4), 196–217.

Zadeh, L.A., 1965. Fuzzy sets. Information and Control 8 (3), 338–353.

Zhu, M.-L., Fujita, M., 1994. Comparison between fuzzy reasoning and neural network method to forecast runoff discharge. Journal of Hydroscience and Hydraulic Engineering 12 (2), 131–141.

Zhu, M.-L., Fujita, M., Hashimoto, N., Kudo, M., 1994. Long lead time forecast of runoff using fuzzy reasoning method. Journal of Japan Society of Hydrology and Water Resources 7 (2), 83–89.