Rainfall-runoff modeling through hybrid intelligent system

P. C. Nayak,[1] K. P. Sudheer,[2] and S. K. Jain[3]

Received 27 January 2006; revised 16 January 2007; accepted 21 February 2007; published 13 July 2007.

[1] This study explores the potential of integrating two different artificial intelligence techniques, namely neural network and fuzzy logic, effectively to model the rainfall-runoff process from rainfall and runoff information. The integration is achieved through representing fuzzy system computations in a generic artificial neural network (ANN) architecture, which is functionally equivalent to a fuzzy inference system. The model is initialized by a hyperellipsoidal fuzzy clustering (HEC) procedure, which identifies suitable numbers of fuzzy if-then rules through proper partition of the input space. The parameters of the membership functions are optimized using a nonlinear optimization procedure. The consequent functions are chosen to be linear in their parameters, and a standard least squares error method is employed for parameter estimation. The proposed model is tested on two case studies: Narmada basin in India and Kentucky basin in the United States. The results are highly encouraging as the model is able to explain more than 92% of the variance. The performance of the proposed model is found to be comparable to that of an adaptive neural based fuzzy inference system (ANFIS) developed for both the basins. The number of parameters in the proposed model is fewer compared to ANFIS, and the former can be trained in lesser time. It is also observed that the proposed model simulates the peak flow better than ANFIS. Overall, the study suggests that the proposed model can potentially be a viable alternative to ANFIS for use as an operational tool for rainfall runoff modeling purposes.

Citation: Nayak, P. C., K. P. Sudheer, and S. K. Jain (2007), Rainfall-runoff modeling through hybrid intelligent system, *Water Resour. Res.*, *43*, W07415, doi:10.1029/2006WR004930.

1. Introduction

[2] Rainfall-runoff modeling is one of the most important topics in hydrology and it is an essential measure in water resources planning and development. Modeling of rainfall-runoff dynamics is performed not only to provide a flood warning system to reduce flood risks but also in managing reservoirs particularly during the drought periods. It is well understood that the relationship between precipitation and runoff is extremely complex owing to temporal and spatial variability of watershed characteristics, heterogeneity in precipitation, as well as numerous factors involved in generating runoff. Among the components involved in transforming precipitation to runoff, the dominant ones are evaporation, infiltration, soil moisture, overland flow, and channel flow [*Beven*, 2000]. In addition, soil properties, land use, and geomorphology of watersheds also play an important role. Consequently, modeling the rainfall-runoff process is a complex task.

[3] Over the last 25 years, a large number of studies have been undertaken to enhance our understanding of rainfall runoff process. The modeling techniques can be broadly classified into two classes: the theory-driven (conceptual and physically based) approach and the data-driven (empirical and black box) approach [*Solomatine and Dulal*, 2003]. Although the theory-driven models provide reasonable accuracy, the implementation and calibration of such models can typically present various difficulties [*Duan et al.*, 1992]; requiring sophisticated mathematical tools, and some degree of expertise and experience with the model. Conventional systems–theoretic models like autoregressive models and their variations [*Box and Jenkins*, 1976] suffer from being based on the linear systems theory and may only be marginally suitable in capturing the highly complex, dynamic, and nonlinear rainfall-runoff process [*Jain and Srinivasulu*, 2004]. Owing to the difficulty associated with parameter optimization in nonlinear systems, the development of nonlinear system theoretic models are very limited [*Hsu et al.*, 1995].

[4] It is reported that most of the hydrologic models are still far from perfect and hydrologists need to put the models in better compliance with observations prior to use in forecasting [*Moradkhani et al.*, 2005]. In this context, data-driven models (DDM), which can discover relationships from input-output data without having the complete physical understanding of the system, may be preferable. While such models do not consider any physics of the hydrologic processes, they are, in particular, very useful for river flow forecasting where the main concern is with making accurate predictions of flow at specific river locations. During the last decade, there has been an increased interest in applying artificial neural network (ANN) and fuzzy inference system (FIS), which are the most common DDM tools, to river flow forecasting [*ASCE Task*

[1]Deltaic Regional Centre, National Institute of Hydrology, Kakinada, India.
[2]Department of Civil Engineering, Indian Institute of Technology Madras, Chennai, India.
[3]National Institute of Hydrology, Roorkee, India.

0043-1397/07/2006WR004930$09.00

Committee on Application of Artificial Neural Networks in Hydrology, 2000a, 2000b; *Dawson and Wilby*, 2001; *Maier and Dandy*, 2000; *See and Openshaw*, 1999; *Hundecha et al.*, 2001; *Xiong et al.*, 2001; *Xiong and O'Connor*, 2002; *Sen and Altunkaynak*, 2003; *Nayak et al.*, 2005a; *Vernieuwe et al.*, 2005; *Chang et al.*, 2005]. The major reason for such an increasing interest in DDM is that they can be easily applied to a wide range of data types and can handle real-life uncertainty with low-cost solutions. Although there is plethora of studies in hydrology using these two computing techniques (ANN and FIS), there are lots of issues (as discussed in a later section) that need to be addressed by the hydrologists to effectively use these to model the rainfall-runoff processes. While both these techniques have been proved to be effective when used on their own, the individual strengths of each approach can be exploited by effectively synergizing them for the construction of powerful intelligent systems.

[5] The major objective of this paper is to develop a hybrid intelligent system (HIS) for rainfall runoff modeling by combining ANN and FIS. The effectiveness of the proposed HIS is evaluated by developing two real world case studies: (1) for Narmada basin in India, and (2) for Kentucky basin, USA. This paper is organized as follows: after the introduction, a brief overview of the data driven modeling in hydrology and the research issues are discussed. The section on proposed HIS describes the identification of premise parameters using hyperellipsoidal fuzzy clustering technique, which includes optimal number of input-output space partitions and membership function parameter optimization using Levenberg-Maquardt (LM) algorithm. In the succeeding section, the development of the HIS model for rainfall-runoff modeling for selected basins is presented. The results are analyzed and discussed in the results and discussions section. The performance of the proposed HIS model is compared with a competing model [*Nayak et al.*, 2005b] developed for the same basins and is discussed in the subsequent sections. The paper ends with conclusions drawn from this study.

2. Data-Driven Modeling in Hydrology and Research Issues

[6] Data-driven modeling (DDM) is based on the analysis of historical hydro-meteorological data sets describing the system and primarily aims at establishing functional relationships between input (in this case rainfall, evaporation, etc.) and output (in this case runoff). In the context of DDM, ANN and FIS are being extensively used for rainfall-runoff modeling. ANN is well suited for hydrologic modeling [*Connor et al.*, 1994; *Atiya et al.*, 1999] as ANNs can approximate virtually any (measurable) function up to an arbitrary degree of accuracy [*Hornik et al.*, 1989]. FIS uses 'if-then' rules and logical operators to establish qualitative relationships among the variables, and has attracted many applications in hydrology [e.g., *Vernieuwe et al.*, 2005; *Chang et al.*, 2005].

[7] Although there is a plethora of studies using these two computing techniques (ANN and FIS), there are lots of issues that need to be addressed by the hydrologists to effectively use these to model the rainfall-runoff process. For instance, while developing rainfall-runoff models using ANNs, *Hsu et al.* [1995] experienced that the ANN models consistently underpredicted the low flows, and overpredicted the medium flows, *Jain and Srinivasulu* [2004] found that the regular back-propagation algorithm fails to guarantee the optimal weight vector for a model. Many researchers have indicated that ANN models are unable to predict extreme values in the river flow [*Minns and Hall*, 1996; *Dawson and Wilby*, 1998; *Campolo et al.*, 1999; *Sudheer et al.*, 2003]. The applications of FIS to runoff modeling are relatively fewer compared to ANNs, probably owing to the bottleneck of the FIS development in the identification of the antecedent parameters which includes optimal number of if-then rules and shape of the membership function (MF). If the number of MFs is large (and therefore the number of fuzzy rules is large), the system requires large computation time for inference [*Babuska*, 1998]. Moreover, the huge rule base may overfit the system and cause it to lose the capability of generalization [*Setnes*, 2000]. If the parameters of MFs are arbitrarily fixed, the tuning process takes long time, and the tuning procedures get easily trapped in local minima [*Jang*, 1993]. In addition, the data sets play a crucial role in DDM since the data obtained from real processes often contain noise, conflicting subsets, and may not adequately cover the entire input space.

3. Hybrid Modeling Approach

[8] *Nayak et al.* [2005b] reported that the limitations of ANN and FIS are almost complementary and advocated their effective integration to improve the computing potential of a model. In general, this integration aims at overcoming the limitation of individual techniques through hybridization or fusion, leading to hybrid intelligent systems (HIS). A synthesis of various techniques is required to create a HIS, and it is crucial for the design of HIS to primarily focus on the integration and interaction of different techniques, rather than merge different methods to create ever-new techniques [*Abraham*, 2001; *Brown and Harris*, 1994]. Among the applications in water resources that employ integrated models using ANN and FIS, *Deka and Chandramouli* [2003] used it for developing stage discharge relationship and *Kisi* [2005] for suspended sediment estimation. Later, *Deka and Chandramouli* [2005] developed a fuzzy neural network (FNN) model for river flow prediction and reported that it performed better compared to an ANN model. Nonetheless, the performance of FNN for estimating the high flows was not good, indicating room for improvement.

[9] *Nayak et al.* [2004, 2005b] employed an HIS by combining ANN and Takagi-Sugeno (TS) type fuzzy model for flood forecasting (ANFIS), which was originally proposed by *Jang* [1993]. They employed gradient descent learning algorithm (back-propagation) for antecedent parameter optimization and grid partitioning to partition input space in a TS fuzzy model. Grid partitioning refers to dividing the input space into rectangular grids and the fuzzy rules are confined to the corners of the grid [*Brown and Harris*, 1994]. However in grid partitioning, regular partition of the input space may not be able to produce a rule set of acceptable size. If, for example, the data contains regions with several small clusters of different classes, then small rule patches have to be created to correctly classify the data in this region. Owing to the grid partition, however, the fine resolution needed in this particular area is also propagated to areas which are much easier to handle, perhaps because they only contain data belonging to a single class. The grid-partitioning approach enforces a large

number of small identical rule patches, although one large patch would theoretically be able to correctly classify the data in this region. It is to be noted that an effective partition of the input space can decrease the number of rules and thus increase the speed in both learning and application phases. To eliminate the problems associated with grid partitioning, scatter partitioning of the input space into rule patches have been proposed [*Setnes*, 2000; *Chiu*, 1994]. In this type the antecedent parts of the fuzzy rules are positioned at arbitrary locations in the input space. This means that rules are not confined to corners of a rectangular grid. Rather, they can be chosen freely, for example, by a clustering algorithm working on the training data. Attractive features of clustering approach for partitioning are simultaneous identification of the antecedent membership functions (MFs), which attribute to minimum number of fuzzy rules from the data set.

[10] In a recent study to identify the most suitable clustering algorithm for this purpose, a hyperellipsoidal clustering (HEC) technique was found to be more effective compared to other methods by *Nayak and Sudheer* [2007]. Similar conclusions have been reported by *Vernieuwe et al.* [2005]. The advantage of HEC is that the method identifies clusters of different geometrical shapes in the data set. This helps in effective implementation of local approximation. Further, the back-propagation algorithm which is used in ANFIS for optimization of MF parameters is a gradient descent method that may get stuck in local minima and the parameters determined on a basis that may not be globally optimal. Typically, the optimization of MF parameters can be considered as a nonlinear optimization problem that minimizes the mean square error of prediction and any nonlinear optimization procedure can be adapted for this. Levenberg-Marquardt (LM) learning procedure is one of the most efficient higher-order adaptive algorithms and often finds better optima [*Efe and Kaynak*, 2001]. This algorithm may be a viable choice for MF optimization.

4. Hybrid Intelligent Systems (HIS)

[11] The HIS proposed in this paper comprises of combining TS fuzzy model with ANN. There are various types of fuzzy rule-based models in the literature [e.g., *Mamdani and Assilian*, 1975; *Tsukamoto*, 1979; *Takagi and Sugeno*, 1985], and each of them is characterized by their consequent function only. The TS fuzzy model has resulted from an effort to develop a systematic approach to generate fuzzy rules from a given input-output data set [*Takagi and Sugeno*, 1985; *Sugeno and Kang*, 1988], in which the rule consequents are typically taken to be either crisp numbers or linear functions of the inputs. The first-order TS model is described below.

[12] Consider a function $y = f(\mathbf{x})$ being mapped by the TS model, in which y is the dependent variable and $\mathbf{x}$ is the column vector (k-dimensional) of independent variables that have a causal relationship with y. Assume that 'n' number of example (patterns) pairs [$\mathbf{x}$, y] are available for parameter estimation. Considering 'm' rules, the mathematical functioning of the TS model is:

$$\boldsymbol{R}_i\text{: If } x_1 \text{ is } A_{i,1} \text{ and} \ldots \text{and } x_k \text{ is } A_{i,k} \text{ then } y_i = \mathrm{a}_i^T \mathrm{x} + b_i, \quad (1)$$

where $\mathbf{x} \in \Re^k$ is the input variables (antecedent), $y_i \in \Re$ is the output (consequent) of the ith rule R_i, and $\mathbf{a}_i$ and b_i are the parameters of the consequent model. The notation a_i^T refers to the transpose the matrix $\mathbf{a}_i$. A_i is the membership function (MF). A Gaussian MF has the form

$$A(x_k) = e^{\frac{1}{2}((x_k - c_i)/(\sigma_i))^2}, \quad (2)$$

where $\{c_i, \sigma_i\}$ is the parameter set termed as centre and spread function. These parameters with maximum equal to 1 and minimum equal to 0 determine the shapes of the membership function. These parameters are called premise parameters or antecedent parameters.

[13] The number of rules is denoted by m and A is the antecedent fuzzy set (membership function) of the i^{th} rule such that

$$A_i(x_k) : \Re^k \to [0, 1], \qquad i = 1 : m. \quad (3)$$

[14] In case of univariate membership functions μ_{ij} (x_j), the fuzzy antecedent in the TS model is typically defined as an AND-conjunction by means of the product operator

$$A_i(x_k) = \prod_{j=1}^{k} \mu_{ij}(x_k). \quad (4)$$

[15] For the nth input patterns $\mathbf{x}_n$, the total output $\hat{y}(n)$ of the model is computed by aggregating the contribution of the individual rules $y_i(n)$,

$$\hat{y}(n) = \sum_{i=1}^{m} u_{ni} y_i(n), \quad (5)$$

where $\hat{y}(n)$ is the estimated output for the pattern x_n, and u_{ni} is the normalized degree of fulfillment of the antecedent clause of rule $\boldsymbol{R}_i$, defined as

$$u_{ni} = \frac{A_i(x_n)}{\sum_{i=1}^{m} A_i(x_n)}. \quad (6)$$

[16] In this way, a weighted average of the individual rule outputs is computed and a nonlinear function can be approximated.

[17] The basic structure of a fuzzy system consists of three conceptual components: a rule base which contains a selection of fuzzy rules, a database which defines the membership function (MF) used in the fuzzy rules, and a reasoning mechanism which performs the inference procedure upon the rules and a given condition to derive a reasonable output conclusion [*Nayak et al.*, 2005b]. A FIS implements a nonlinear mapping from its input space to an output space. A FIS can utilize human expertise by storing its essential components in a rule base and data base, and perform fuzzy reasoning to infer the overall output value. The derivation of if-then rules and corresponding membership functions depends heavily on the a priori knowledge about the system under consideration. However, there is no systematic way to transform experience or knowledge of human experts to the knowledge base of a FIS.

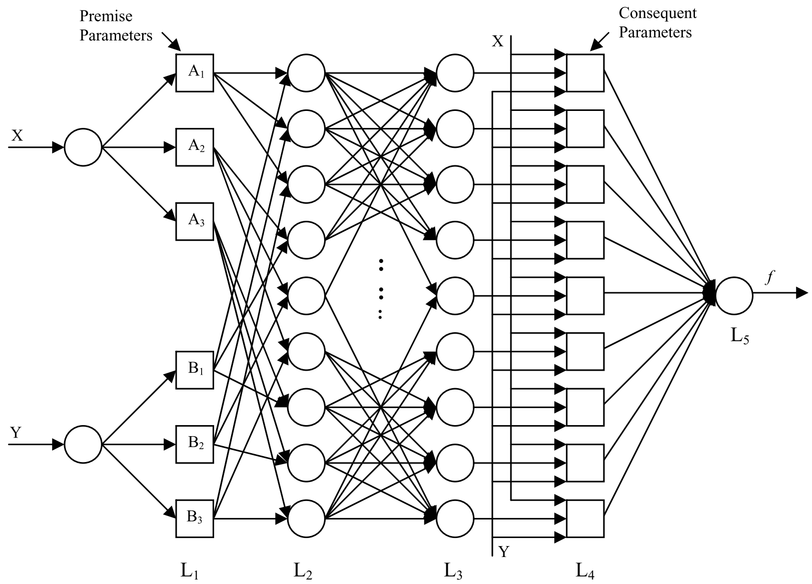

Figure 1. Representation of fuzzy inference system in a functionally equivalent ANN architecture.

[18] ANN learning mechanisms do not rely on human expertise. Owing to the highly parallel structure of an ANN, it is hard to extract structured knowledge from either the weights or the configuration of the ANN. The weights of the ANN represent the coefficients of the hyperplane that partitions the input space into two regions with different output values. If one can visualize the hyperplane structure from the training data then the subsequent learning procedures in an ANN can be reduced. On the contrary, a priori knowledge is usually obtained from the human experts and it is most appropriate to express the knowledge as a set of fuzzy if-then rules. The limitations that arise when these techniques (ANN and FIS) are individually used can be addressed by creating hybrid systems by combining the two techniques. A common way to integrate FIS with ANN is to represent FIS in a general ANN architecture called adaptive neural network, and to use the learning algorithms of ANN to estimate MF parameters [*Jang*, 1993]. However, the conventional ANN learning algorithms (e.g., gradient descent) cannot be directly applied to such a system as the transfer function of the FIS need not usually be 'nondifferentiable.' This problem can be tackled by using differentiable functions in the inference system or by not using the standard neural learning algorithm.

4.1. Integration of FIS and ANN

[19] In theory, ANN and FIS are equivalent in that they both are convertible. Figure 1 shows a neural network representation of FIS with two inputs x and y. Each of these input variables is associated with 3 MFs. The computations are performed in 5 layers (L_1 through L_5). It is to be noted that the directions marked in Figure 1 imply only the flow of information and that no weights are associated with the links.

[20] For a first-order TS model, a common rule set with two fuzzy if-then rules can be written as

Rule 1

$$\text{If x is } A_1 \text{ and y is } B_1, \text{ then } f_1 = p_1 x + q_1 y + r_1 \quad (7)$$

Rule 2

$$\text{If x is } A_2 \text{ and y is } B_2, \text{ then } f_2 = p_2 x + q_2 y + r_2, \quad (8)$$

where x and y are linguistic variables A_1, A_2, B_1, B_2 are corresponding fuzzy sets and p_1, q_1, r_1 and p_2, q_2, r_2 are consequent (linear) parameters. Each node in the first layer (L_1) generates the membership grade of an input variable. These membership grades of each input variable are combined at second layer (L_2) to compute the firing strength of a rule. In the third layer (L_3), the firing strength of each rule is normalized by considering the combined firing strength of all rules. The contribution of each rule toward the model output is computed in the fourth layer (L_4). Overall output of the model is computed at fifth layer (L_5) by combining the signals received from the previous layer.

4.2. Parameter Identification

[21] The parameters for optimization in HIS are the number of rules, the premise parameters, which describe the shape of the MFs, and the consequent parameters, which describe the overall output of the system. Ideally for n domains (MFs) and p input variables, there could be n^p different if-then rules. Each of these rules is characterized by distinct antecedent and consequent parameters. As a result, increasing the number of membership functions on the input variables will increase the number of fuzzy if-then rules; simultaneously it increases the model complexity and hence affects the model parsimony.

[22] In the current study, the parameters of the HIS model were identified in a sequential manner. Initially, the number of rules and the fuzzy antecedents A_i in the rules were determined using a fuzzy clustering algorithm. The initial values of the antecedent parameters identified by the clustering algorithm essentially require further tuning. Fine tuning for the number of rules is also required to evaluate the sensitivity. This is achieved in a nested computational framework where two loops are nested: the parametric (antecedent and consequent parameters) and the structural

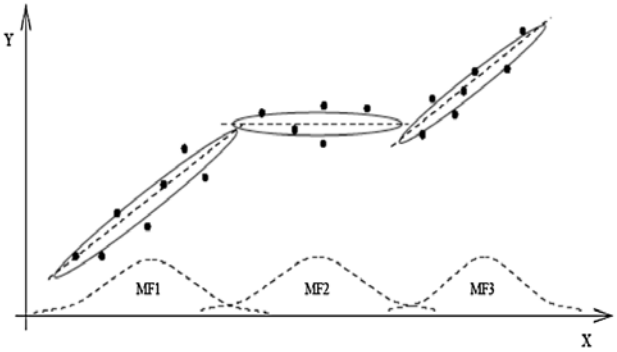

Figure 2. Hyperellipsoidal fuzzy clustering initialization procedure.

(number of rules). The parametric loop (the inner one) searches for the best set of parameters by minimizing a sum-of-squares cost function, which depends exclusively on the training set. In the case of linear TS models, this cost minimization procedure (for estimating antecedent and consequent parameters) can be decomposed into a combination of a least squares problem to estimate the linear parameters of the consequent models f^i and a nonlinear optimization to find the parameters of the MFs. As stated earlier, the current study employed Levenberg-Marquardt algorithm for the optimization of antecedent parameters. The structural loop evaluates the sensitivity of the number of rules identified by the fuzzy algorithm. Details of various algorithms employed for parameter identification are discussed in the following sections.

4.3. Initialization of Antecedent Parameters and Number of Rules by Hyperellipsoidal Clustering

[23] A fuzzy inference system (FIS) can be viewed as a partition in the multidimensional feature space where the number of partitions in each dimension corresponds to the number of fuzzy sets and the corresponding membership function that are defined in that dimension. Consequently, the input space partitioning plays a major role in the optimal architecture of the model. Input space partitioning is carried out in different ways: grid and scattering partitioning. As mentioned earlier, a clear drawback of grid partitioning is that the number of rules grows exponentially and the optimization of antecedent parameter becomes complex. Consequently, while developing a FIS, scattering partitioning is commonly employed [*Setnes*, 2000] for FIS structure identification as the antecedent parameters are obtained directly from the fuzzy clusters. In the present study hyperellipsoidal fuzzy clustering approach has been applied to find out the rules from the hybrid system. Details of the clustering procedure are given below.

[24] Clustering can be depicted as a problem to minimize the sum of the within cluster dissimilarity measures. Given a set of n different samples $\{x_i|x_i \in R^k, i = 1, 2,\ldots, n\}$, which are in the k-dimensional real space, the goal of clustering is to partition these samples to m clusters $\{c_j|c_j \in R^k, j = 1, 2,\ldots, m\}$. Let $\mu = (\mu_{ij}) \in \Re^{nXm}$ be the assignment matrix, where μ_{ij} denotes the membership value of x_i in j^{th} cluster. Vector set $M = (p_1, p_2, \ldots, p_j, \ldots, p_m) \in \Re^{kxm}$ is the set of prototypes of $\boldsymbol{m}$ clusters. The similarity measure between x_i and prototype p_j is $D(x_i, p_j)$. It is well known that the clustering cost function, for example, the sum of the within-cluster dissimilarity measures, is

$$J_j(\mu, p) = \sum_{i=1}^{n}\sum_{j=1}^{m}\mu_{ij}^{\alpha}.D\left(x_i, \mathrm{p_j}\right), \tag{9}$$

subject to

$$0 \le \mu_{ij} \le 1; i = 1, 2, \ldots, n; j = 1, 2, \ldots, m,$$

where α is the fuzzy exponent whose typical value is 2, and

$$\sum_{j=1}^{m}\mu_{ij} = 1, i = 1, 2, ..., n. \tag{10}$$

[25] The function $D(x_i, p_j)$ measures the distance between the data vector x_i and the center p_j of the jth cluster. The most commonly used distance measure is the squared Euclidean distance. A clustering algorithm with Euclidean distance favors hyperspherically shaped clusters of equal size. This has the undesirable effect of splitting large as well as elongated clusters under some circumstances. Indeed, it has been noted that most clusters in real data sets are neither well-isolated nor have the spherical shape. Other distance measures, such as the Mahalanobis distance, can be used to find hyperellipsoid shaped clusters. The squared Mahalanobis distance between a pattern vector x_i and a centre p_j is defined as follows:

$$D\left(x_i, \mathrm{p_j}|\lambda_j\right) = \left(x_i - \mathrm{p_j}\right)^T\lambda_j^{-1}\left(x_i - \mathrm{p_j}\right), \tag{11}$$

where λ_j^{-1} is the inverse of the $[n\ X\ n]$ covariance matrix of the jth cluster. The matrix λ_j is defined as

$$\lambda_j = \frac{\sum_{i=1}^{n}\mu_{ij}^{\alpha}\left(x_i - p_j\right)\left(x_i - p_j\right)^T}{\sum_{i=1}^{n}\left(\mu_{ij}^{\alpha}\right)}, 1 \le j \le m, \tag{12}$$

where p_j is the jth center coordinate and μ_{ij} is the membership between x_i and p_j, and n is the total number of the training samples.

[26] The axes of the ellipsoids (eigenvectors of the scatter matrix) are used to initialize the parameters of the consequent functions. The cluster centers on the input domain are projected to initialize the centers of the antecedents and the scatter matrix is adopted to compute the width of the MFs. An example of fuzzy clustering in the case of a single-input-single-output function modeled by a fuzzy inference system with Gaussian antecedents is represented in Figure 2.

4.4. Optimization of Antecedent Parameters by Levenberg-Marquardt Algorithm

[27] LM is a nonlinear optimization technique, considered as the standard of nonlinear least squares function minimization [*Press et al.*, 1994]. LM method is an approximation to Newton's method. The algorithm uses the second-order derivatives of the cost function so that a better convergence behavior is observed. In the ordinary gradient descent method, only the first-order derivatives are evaluated and

the parameter change information contains solely the direction along which the cost is minimized. In the LM technique, a better parameter change vector is determined. It has the advantage that it requires only the computation of the output gradients with respect to parameters which can be obtained efficiently using the back-propagation technique. A detailed mathematical description of the LM algorithm is given by *Olaru and Wehenkel* [2003].

[28] The objective function used in the LM algorithm is

$$e = d - F(\Phi, u), \tag{13}$$

$$E = \frac{1}{2}e^2, \tag{14}$$

$$\Delta\Phi = -\left(\nabla^2 E(\Phi)\right)^{-1} \nabla E(\Phi), \tag{15}$$

where e is observed output error, d is desired output, F is fuzzy system response, Φ is a generic parameter of fuzzy system, E is cost function, $\Delta\Phi$ is change in parameter Φ, $\nabla^2 E(\Phi)$ is the Hessian matrix and $\nabla E(\Phi)$ is the gradient relevant to the cost of equation (14). The observation error in equation (13) is to minimize the realization cost in equation (14) by utilizing the rule described by equation (15). The objective is to minimize instantaneous cost defined by equation (14). If the Taylor series expansion is applied around the operating point to e, which is a function of Φ, the first derivatives result in the Jacobean is given by

$$E_s = \begin{bmatrix} \frac{\partial e_1}{\partial \Phi_1} & \cdots & \frac{\partial e_1}{\partial \Phi_B} \\ \cdot & \cdot \quad \cdot & \cdot \\ \cdot & \cdot \quad \cdot & \cdot \\ \cdot & \cdot \quad \cdot & \cdot \\ \frac{\partial e_L}{\partial \Phi_1} & \cdots & \frac{\partial e_L}{\partial \Phi_B} \end{bmatrix}. \tag{16}$$

[29] In equation (16), $\boldsymbol{B}$ is the number of adjustable parameters and $\boldsymbol{L}$ is the number of outputs. The final form of the parameter update algorithm is given below:

$$\Delta\Phi = N_\Phi = -\left(E_S^T E_S + qI\right)^{-1} E_S^T e, \tag{17}$$

where I is the identity matrix and q is a time-varying stabilization parameter. Here q will decrease after each reduction in the error function and will increase only when a tentative step would increase the error function.

4.5. Consequent Parameter Identification Using LSE

[30] Let $\mathbf{x}_e$ denote the matrix [$\mathbf{x}$, 1] with rows [x_l^T, 1]. The activation of each rule R_i, i = 1, 2,..., m is gathered in Γ_i which is a diagonal matrix in $\Re^{kXk}$ having the normalized degree of fulfillment u_{ni} as its nth diagonal element. Further, denote by X$'$ the matrix in $\Re^{nXmn}$ composed from matrices obtained by multiplying the matrices Γ_i and $\mathbf{x}_e$, suh that

$$\mathrm{X}' = [\Gamma_1 \mathrm{x}_e, \Gamma_2 \mathrm{x}_e, \ldots, \Gamma_m \mathrm{x}_e]. \tag{18}$$

[31] Denote by θ' the vector in $\Re^{nXmn}$ given by

$$\theta' = \left[\theta_1^T, \theta_2^T, \ldots, \theta_m^T\right]^T, \tag{19}$$

where $\theta_i^T = [a_i^T, b_i]$ *for* $1 \le i \le m$. The model in equation (4) can now be written as a regression model,

$$\mathrm{y} = \mathrm{X}'\theta' + e, \tag{20}$$

where e is the approximation error. From this, the least squares solution to the consequent parameter estimation problem can be written as

$$\theta' = \left[(\mathrm{X}')^T \mathrm{X}'\right]^{-1} (\mathrm{X}')^T \mathrm{y}. \tag{21}$$

4.6. Implementation of Learning Procedure in HIS

[32] As discussed in the previous sections, two types of tuning are implemented, namely structural and parametric tuning. Structural tuning aims to find suitable number of rules through a proper partition of the input space using the clustering algorithm. Once a satisfactory structure is determined, the parametric tuning searches for the optimal MFs using LM algorithm together with the optimal parameters of the consequent models using LSE algorithm. However, HIS also suffers from the problem of equifinality; that is, there may be a number of structure/parameter combinations, all resulting in similar performance. The problem can be addressed by considering minimum structural complexity (in terms of number of rules) and maximum generalization property for the fitted model. An incremental approach was adopted in the current study where different architectures having different complexity (i.e., number of rules) were first assessed in K-fold cross validation [*Stone*, 1974] and then compared across in order to select the best one. The whole learning procedure is represented in the flow chart in Figure 3.

5. Application to Rainfall-Runoff Modeling

[33] The HIS discussed above is illustrated through two examples by developing rainfall-runoff models: (1) for a subbasin of the Narmada River up to the Manot gauging site in India, (2) the Kentucky basin, USA. The rationale behind the selection of these case studies are that (1) the available data were in different timescales (hourly for Narmada, and daily for Kentucky) and (2) the two basins are hydrologically different in terms of climate as well as magnitude of river flow. These applications will help in evaluating the statistical significance of the model proposed in this study.

5.1. Narmada Basin

[34] The subbasin of the Narmada River up to the Manot gauging site lies between East longitudes 80°24$'$ to 81°47$'$ and North latitudes 22°26$'$ to 23°18$'$, most of the part lying in Mandla district and some part in Shadol district of Madhya Pradesh in India (Figure 4). The basin occupies an area of 4980 km^2 and length of the river is about 269 km. Here Narmada flows in a generally northwesterly direction, but just upstream of Manot, it turns in a loop to the south. The climate of the basin is humid and tropical, although at places extremes of heat and cold are often encountered. Topography of the Narmada basin above Manot is hilly with

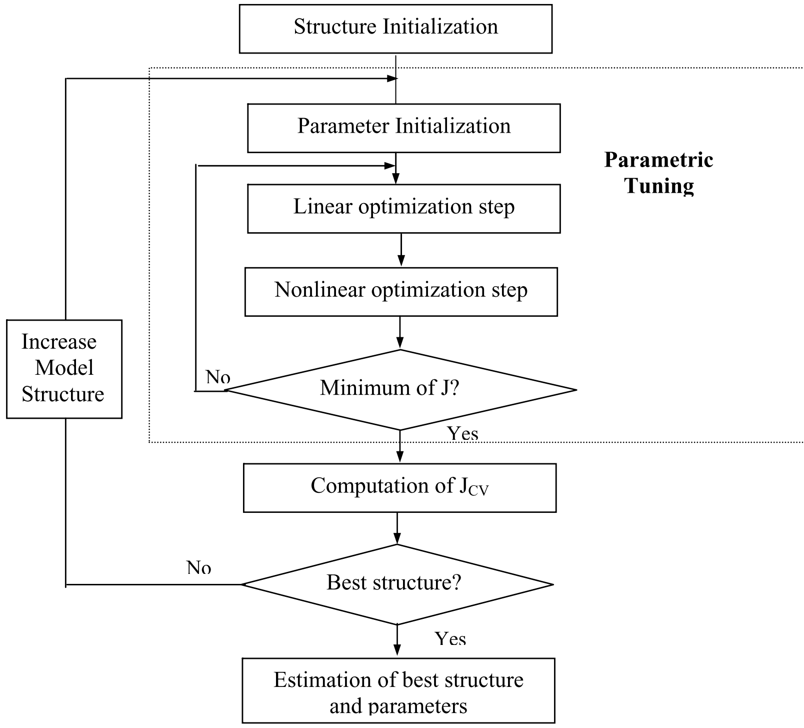

Figure 3. Flow chart of HIS learning procedure.

forest cover, especially in upper reaches. Flat farmland is more evident in the lower reaches. Flat agricultural areas containing bunded fields are interrupted by low hills with a medium to dense forest cover. Topographically, the basin can be divided into three distinct levels: low land areas, hill slopes or semihilly areas, and upland or hilly areas. The elevation ranges from 450 m near the Manot gauging site to 1100 m in the upper part of the basin. The study area consists of mainly black soil. At the end of the dry season, cracks in the soil may be 2 to 6 m deep and support rapid infiltration of the early monsoon rains, thereby inhibiting runoff. In the upland forest areas, surface runoff may be generated at earlier stage in monsoon season because of the shallower soils. The runoff may also be more concentrated in small channels whereas in the flat low land areas, sheet flow may be more prevalent.

[35] In the current study, rainfall and runoff data on an hourly interval during the monsoon season (June to October) for two years (1992–1993) are used. The hourly rainfall data are available for four stations over the entire Narmada basin (one of them falling in the study area; see Figure 4) and areal averages of rainfall values, computed by constructing Thiessen polygons, are used in the study.

5.2. Kentucky Basin

[36] Figure 5 shows the map of the Kentucky River basin, which encompasses over 4.4 million acres of the state of Kentucky. Forty separate counties lie either completely

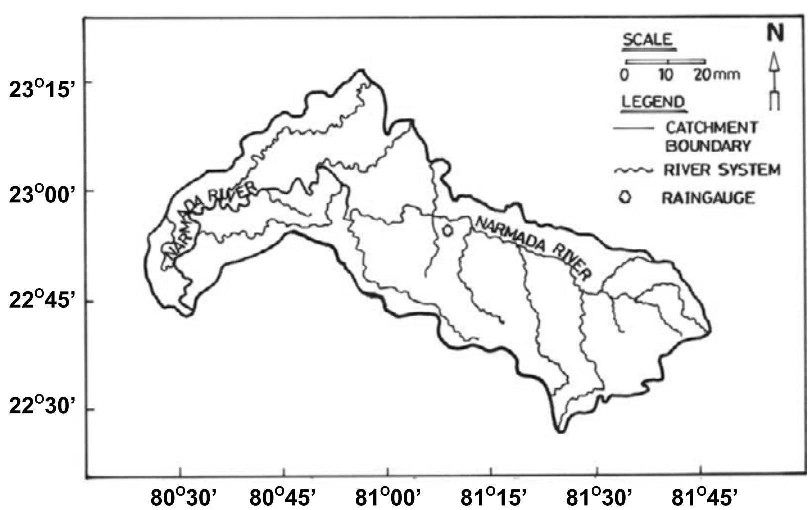

Figure 4. Study area map of the river Narmada.

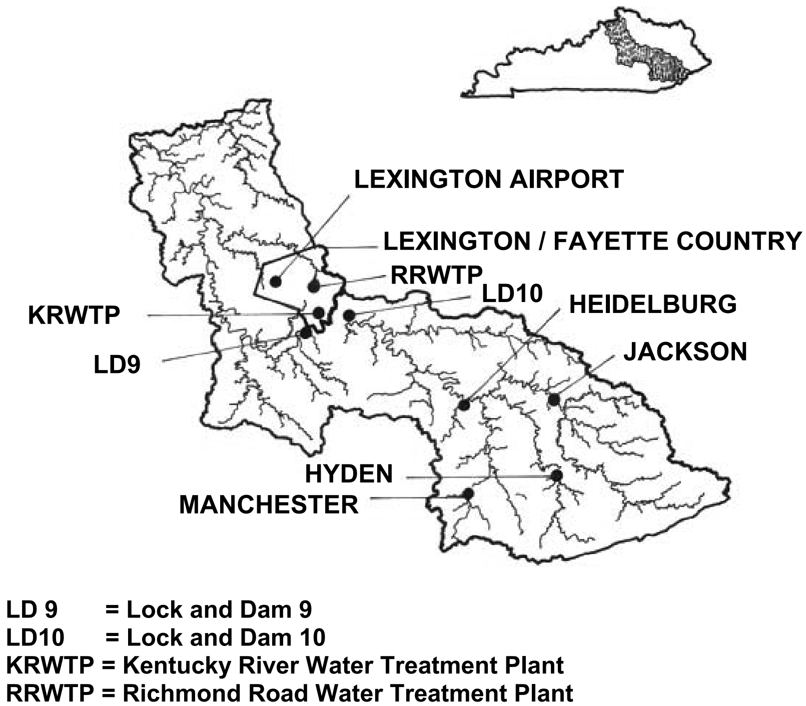

Figure 5. Study area map of River Kentucky.

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 the US Army Corps of Engineers. The drainage area of the Kentucky River at Lock and Dam 10 (LD10) near Winchester, Kentucky is approximately 6300 km^2. The data used in the study presented in this paper include average daily streamflow (m^3/s) from the Kentucky River at LD10 and daily average rainfall (mm) from five rain gauges (Manchester, Hyden, Jackson, Heidelberg, and Lexington Airport) scattered throughout the Kentucky River Basin (see Figure 5). The total length of the available rainfall runoff data was 26 years (1960–1989 with data in some years missing).

5.3. Input Selection and Data Preprocessing

[37] One of the most important steps in the model development process is the determination of significant input variables. Usually, not all of the potential input variables will be equally informative since some may be correlated, noisy or have no significant relationship with the output variable being modeled [*Maier and Dandy*, 2000]. Generally some degree of a priori knowledge is used to specify the initial set of candidate inputs [e.g., *Campolo et al.*, 1999; *Thirumalaiah and Deo*, 2000]. Although a priori identification is widely used in many applications and is necessary to define a candidate set of inputs, 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 [e.g., *Sajikumar and Thandaveswara*, 1999; *Luk et al.*, 2000; *Silverman and Dracup*, 2000; *Coulibaly et al.*, 2000, 2001; *Sudheer et al.*, 2002]. The major disadvantage associated with using cross correlation is that it is only able to detect linear dependence between two variables. Cross correlation is unable to capture any nonlinear dependence that may exist between the inputs and the output, and may possibly result in the omission of important inputs that are related to the output in a nonlinear fashion. *Bowden et al.* [2004], 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 follows that there is good scope for addressing this issue in future studies.

[38] The current study employed 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 correlations, autocorrelations, and partial autocorrelations between the variables in question along with their 95% confidence interval. By analyzing these correlogram plots, the significant lags of independent variables that are potentially influencing the output (dependant variable) can be identified. The correlogram for both the basins are presented in Figures 6 and 7 respectively for Narmada and Kentucky basins. The input vector identified according to *Sudheer et al.* [2002] for modeling the river flow in Narmada included a total number of 6 variables, and hence

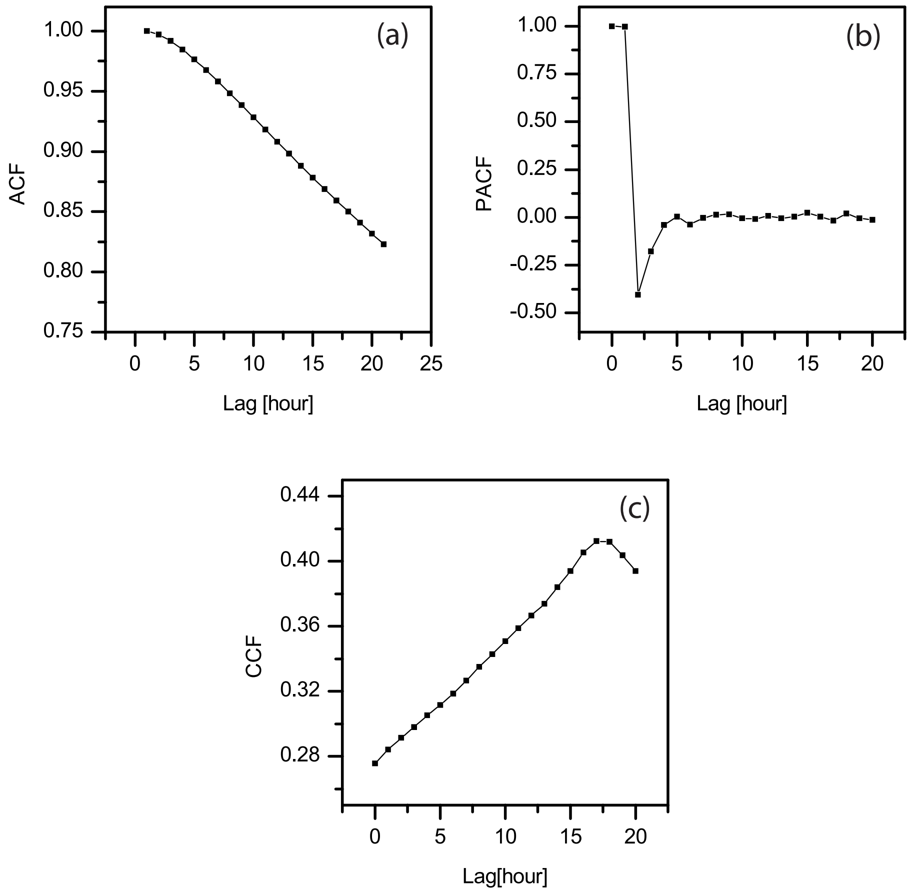

Figure 6. Correlogram plot of the rainfall-runoff series at Narmada basin: (a) autocorrelation (ACF) of runoff series, (b) partial autocorrelation (PACF) of runoff series, and (c) cross (serial) correlation (CCF) between rainfall and runoff.

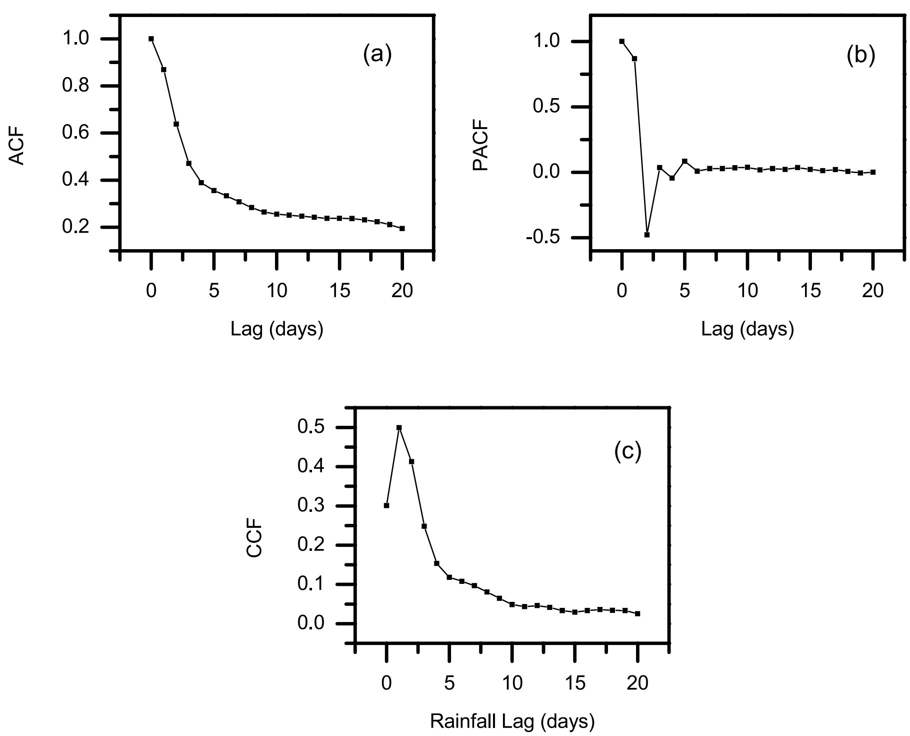

Figure 7. Correlogram plots of the rainfall-runoff series at Kentucky: (a) autocorrelation of runoff series, (b) partial autocorrelation of runoff series, and (c) cross (serial) correlation between rainfall and runoff.

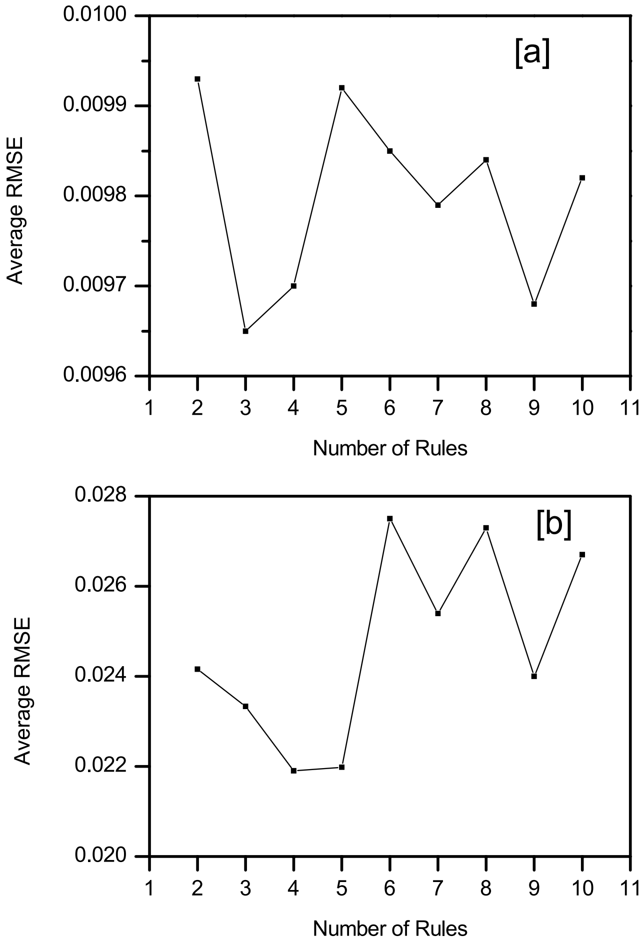

Figure 8. Variation of average RMSE (scaled domain) with number of fuzzy if-then rules during cross validation (a) for Narmada basin and (b) for Kentucky basin.

the functional form of the HIS, in the case of Narmada, for rainfall runoff modeling is given by

$$Q(t) = f[R(t-16), R(t-17), R(t-18), \cdot Q(t-1), Q(t-2), Q(t-3)], \quad (22)$$

where Q(t) and R(t) are river flow and rainfall respectively at any time t in hour.

[39] Similarly for Kentucky basin, from Figure 7, it can be seen that the most appropriate input vector according to *Sudheer et al.* [2002] includes streamflows up to a lag of 2 days and precipitation up to a lag of 2 days along with the current day precipitation.

[40] *Sudheer et al.* [2003] suggest that by following the guidelines used in traditional statistical modeling, the model performance can be improved in the case of ANN based models. They illustrated that since an ANN is doing a local approximation of the input-output function, a local variation induced by the skewness of the data may influence the effectiveness of the model. Consequently, they suggested that an appropriate transformation that reduces the skewness of the original data may help improve the performance of ANN models. Data transformations are often used to simplify the structure of the data so that they follow a convenient statistical model [*Sudheer et al.*, 2003]. In the neuro-fuzzy approach (of modeling the rainfall-runoff process or any other function), the time series (with or without exogenous inputs) is first embedded in a state space using delay coordinates, and the underlying nonlinear mapping is inferred by a local approximation using only the nearby states. Therefore any local variations in the data will certainly influence the model performance. Hence the suggestions *by Sudheer et al.* [2003] can be very much applied to neuro-fuzzy systems also. Accordingly, log-normal transformation was used and the deterministic component in the runoff and rainfall series was removed prior to the modeling in the current study.

[41] While developing any model, the total available samples are generally divided into training and validation sets prior to the model building, and in some cases a cross-validation set is also used. In the majority of data-driven modeling applications in hydrology, the available data are divided arbitrarily into the required subsets. However, recent studies have shown that the way the data are divided can have a significant impact on the results obtained [*Tokar and Johnson*, 1999]. In other words, though the network may fit training samples with great precision, it may fail to simulate flows outside the range of the training data. It has been suggested that the statistical properties (e.g., mean and standard deviation) of the various data subsets need to be considered to ensure that each subset represents the same population [*Shahin et al.*, 2000]. In this study, the rainfall and discharge data during the years 1992 are used for training the HIS, and calibrated model is tested using the data for the year 1993, for Narmada basin. The mean value of river flow for the calibration and validation data was 291.45 m^3/s and 265.86 m^3/s respectively. The standard deviation of the two data sets were 450.05 (calibration) and 387.91 (validation). The training data set for Kentucky basin is for thirteen years (1960–1972) for parameter estimation, and a testing data set of thirteen years (1977–1989) for validation of the model. While dividing the data into training and validation set, care has been taken to ensure similar statistical properties of both the data sets.

5.4. Performance Evaluation of HIS

[42] To evaluate the adequacy of developed model, different evaluation measures are considered and the resulting hydrographs from the developed model are analyzed. These indices include the root mean square error (RMSE) between the computed and observed runoff, coefficient of correlation (CORR), the model efficiency (EFF). These indices are the most commonly used indices in any soft computing application and hence the definition for these indices can be obtained from literature [e.g., *Hsu et al.*, 1995].

[43] The predictive uncertainty of the models is evaluated by an index called the noise-to-signal ratio. The unbiased standard error of estimate (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 the STD, then the model can provide accurate predictions. On the contrary,

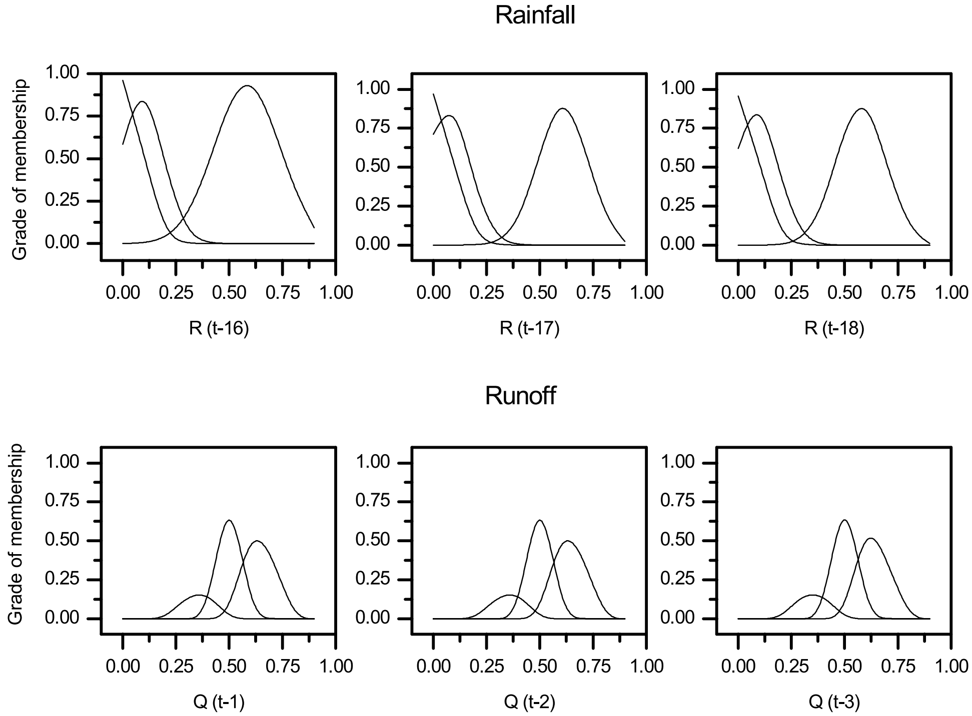

Figure 9. Optimal membership functions for three fuzzy if-then rules (for Narmada basin).

if the ratio is greater than or equal to unity, then the model predictions will not be accurate [*McCuen*, 1993].

[44] Noise to signal ratio

$$NS = \frac{SEE}{\sigma_y}, \tag{23}$$

where

$$SEE = \sqrt{\frac{\sum_{t=1}^{n} \left(y_t^o - y_t^c\right)^2}{v}}, \tag{24}$$

where y_i^o and y_i^c respectively are the observed and computed flow values at time t, and $\overline{y^o}$ and $\overline{y^c}$ are the mean of the observed and computed flow values corresponding to n patterns, v is the degree of freedom and σ_y is the standard deviation of the observed flow.

6. Results and Discussions

6.1. HIS Parameters

[45] The number of rules was determined using hyper-ellipsoidal fuzzy clustering algorithm. As stated earlier, K-fold cross-validation procedure [*Stone*, 1974], considering 10 subsets, was employed to find out the best architecture of the HIS. The values of root mean squared error (RMSE) were used as the index to check the performance during cross validation. The plot showing cross-validation error against the number of rules is presented in Figure 8a, from which it can be observed that average RMSE value is minimum for 3 fuzzy if-then rules in the case of Narmada basin. The optimized membership functions corresponding to 3 fuzzy if-then rules are presented in Figure 9, in which the MF overlapping is clearly visible. Since medium- and high-range flow values were high compared to low values in the data, the MF shows high grade for medium- and high-range values. The input coordinates of the 3 cluster centers obtained are given below:

$$\mathrm{x}_1^* = [0.1495 \quad 0.14691 \quad 0.14998 \quad 0.45236 \quad 0.45229 \quad 0.4522]$$
$$\mathrm{x}_2^* = [0.1566 \quad 0.15704 \quad 0.15601 \quad 045522 \quad 0.45511 \quad 0.4550]$$
$$\mathrm{x}_3^* = [0.1639 \quad 0.16639 \quad 0.16415 \quad 0.45610 \quad 0.45595 \quad 0.4558]$$

where x_i^* represent *i*th cluster centre, and the values in each raw correspond to the vector of the input values given in the functional relationship in the order

$$[\mathrm{R(t-16), R(t-17), R(t-18), Q(t-1), Q(t-2), Q(t-3)}],$$

where R and Q are as defined earlier. Note that the values of coordinates in the x_i^* matrix are log-transformed, standardized values of variable. The corresponding output equations (consequent) derived for each cluster are as follows:

$$\mathrm{y}_1^* = [0.0250 \quad 0.0747 \quad -0.035 \quad 2.3885 \quad -0.0575 \quad -0.5468] \cdot [\mathrm{x}]^\mathrm{T} + 3.629$$
$$\mathrm{y}_2^* = [2.0441 \quad 2.5185 \quad 2.0675 \quad 3.0382 \quad -1.6881 \quad 2.5850] \cdot [\mathrm{x}]^\mathrm{T} + 7.728$$
$$\mathrm{y}_3^* = [0.8328 \quad 1.1985 \quad 1.0510 \quad 0.2565 \quad 2.4531 \quad -1.2356] \cdot [\mathrm{x}]^\mathrm{T} - 12.223$$

where y_i^* is the model output for i^th cluster corresponding to an input vector x.

Table 1. Statistical Performance Indices of HIS and ANFIS for Narmada and Kentucky Basin

	Narmada Basin				Kentucky Basin			
	HIS		ANFIS		HIS		ANFIS	
Statistical Indices[a]	Calibration	Validation	Calibration	Validation	Calibration	Validation	Calibration	Validation
CORR	0.997	0.996	0.999	0.990	0.986	0.964	0.984	0.942
Efficiency, %	99.55	99.24	99.78	98.11	95.18	92.99	96.92	87.60
RMSE, m^3/s	33.09	35.78	20.88	53.30	51.57	55.42	42.30	73.73
NS	0.075	0.052	0.047	0.138	0.220	0.265	0.176	0.353

[a]CORR, coefficient of correlation; NS, noise to signal ratio.

[46] In the case of Kentucky basin, the minimum value of average RMSE during cross validation was obtained corresponding to 4 clusters (Figure 8b), indicating 4 rules for the HIS model for Kentucky. The optimized consequent equations for Kentucky basin are as follows:

$$y_1^* = [3390.20 \quad 4404.60 \quad -286.83 \quad 1990.80 \quad 2585.80] \cdot [x]^T + 174630.00$$

$$y_2^* = [13080.0 \quad 11123.00 \quad -2336.40 \quad 11825.00 \quad 12480.00] \cdot [x]^T - 43471.00$$

$$y_3^* = [-148.04 \quad -82.39 \quad -126.56 \quad -163.89 \quad -170.69] \cdot [x]^T - 2136.60$$

$$y_4^* = [3688.3 \quad 2039.30 \quad 573.85 \quad 3791.30 \quad 4314.60] \cdot [x]^T - 133980.00$$

where the variables are as defined earlier.

6.2. HIS Model Performance

[47] The values of the evaluation measures during calibration and validation period for HIS for both the basins are summarized in the Table 1, from which it is very vivid that HIS is able to mimic the rainfall runoff transformation reasonably well. The correlation statistic, which evaluates the linear correlation between the observed and the computed runoff, is consistent during calibration and validation period for both the basins. The RMSE statistic is a measure of residual variance, and its value for the current model varies from 33 to 36 m^3/s, in the case of Narmada indicating a very good performance when compared to the mean flow (hourly) value of 290 m^3/s. The mean daily flow in Kentucky is 170 m^3/s, and the RMSE statistic in this case is found to be of the order of 30% of the mean flow. The HIS performance is very good in terms of the efficiency statistic [*Shamseldin*, 1997], as the calibration and validation efficiency is greater than 99% in the case of Narmada. The noise to signal ratio (NS) illustrates how the noise is hiding the information in the model performance, and a value less than unity is preferred. It should be noted that in the case of HIS the value of NS is close to zero in the case of Narmada, while it is of the order of 22% in the case of Kentucky.

[48] The scatterplots of flows (observed and computed by HIS) during calibration and validation period for both the case examples are presented in Figures 10 and 11 for comparison respectively for Narmada and Kentucky basins. These plots give clear indication of the simulation ability of the developed model across the full range of flows. It is noted that most of the flows tend to fall close to the 45° line (rather reduced scattering), showing a good agreement between observed and forecasted flows, even though the scatter is relatively high in the case of Kentucky. Although the results in general indicate the potential of HIS in effective modeling the rainfall runoff relationship, it is observed from Table 1 as well as Figure 11 that the performance of HIS is relatively lower for Kentucky basin compared to that for Narmada basin; this may be plausibly attributed to the variation in nonlinear dynamics of the mapped relationships in different timescales (hourly in Narmada versus daily in Kentucky).

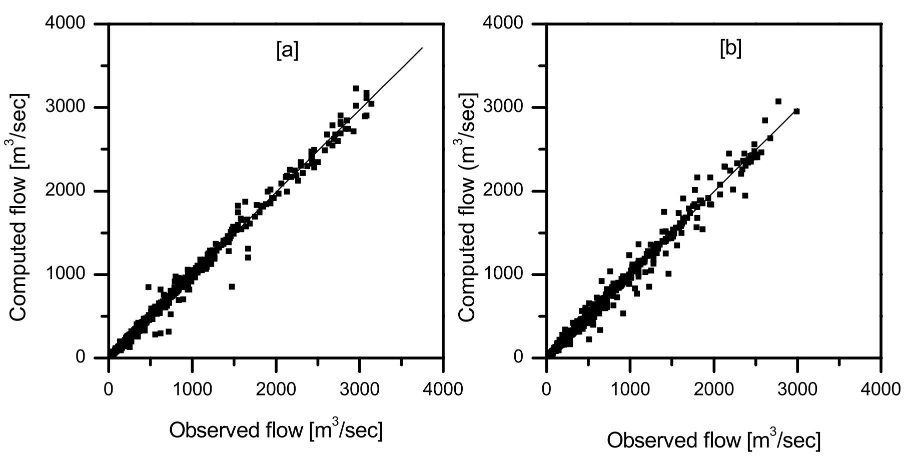

Figure 10. Scatterplots for observed and computed flows (a) calibration and (b) validation period for Narmada basin.

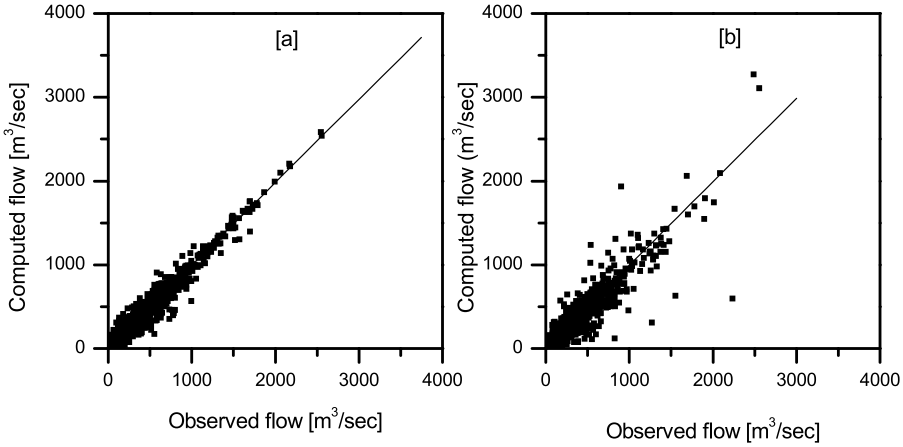

Figure 11. Scatter plots for observed and computed flows (a) calibration and (b) validation period for Kentucky basin.

6.3. Comparison With Other HIS Model

[49] In order to evaluate the potential of the proposed HIS over ANFIS [*Nayak et al.*, 2005b], an ANFIS model was developed for both the basins using the same input parameters. The ANFIS algorithm also produces fuzzy models consisting of the Takagi-Sugeno type rules [*Nayak et al.*, 2004, 2005b]. The major difference between the ANFIS and the proposed HIS is that ANFIS uses back-propagation algorithm to determine premise parameters (to learn the parameters related to membership functions) and least squares estimation to determine consequent parameters. Gaussian membership functions are selected for the model and input space partitioning is carried out using grid partitioning. Note that in the grid partitioning method, ideally for n domains (rules) and p input variables there could be n^p different if-then rules. In the current study, the number of MFs assigned to each input variable has been varied from 2 (64 and 32 rules respectively for Narmada and Kentucky) to 4 (4096 and 1024 rules respectively for Narmada and Kentucky) and it was observed that ANFIS suffers from curse of dimensionality beyond 2 membership functions, which is manifested by an exponentially growing number of rules in relation with the pattern dimensionality. The final model parameters were identified through cross validation technique in the case of ANFIS also.

[50] The performance indices for the developed ANFIS model in estimating flow values in both the case examples are also summarized in the Table 1. It is evident from the Table 1 that the performance of the ANFIS model is close to HIS during training as well as validation. However, it is observed that the RMSE for ANFIS during validation is not in similar range as during calibration, implying a problem of overtraining. It is to be noted that huge rule base may overfit the system and cause it to lose the capability of generalization. It is logical to believe that as ANFIS has more number of parameters, it may outperform HIS. Note that the ANFIS model uses two membership functions for each input variable (6-input runoff model for Narmada), leading to 64 fuzzy partitions in the input space, and thus 64 rules (5-input runoff model for Kentucky implying 32 rules). Consequently, in the case of Narmada basin, the developed ANFIS model has 24 parameters (2 Gaussian MF per input each having 2 parameters) that are optimized through back-propagation and 448 parameters optimized through linear least squares estimation (7 parameters for each consequent equation). On the contrary, the proposed HIS has only 3 fuzzy rules comprising of 36 premise parameters (3 Gaussian MF per input) and 21 consequent parameters. Evidently, despite a significant reduction in the number of rules HIS results in similar performance as that of ANFIS. This observation is found to be true in the case of Kentucky basin too.

[51] Constructing models from data with nontrivial dynamics involves the problem of how to choose the best model from within a class of models, or to choose between the competing classes. The model selection problem involves selecting *k* nonzero elements (λ, the parameters of the model) in a given nonlinear model, $g(\mathbf{x}, \lambda)$. In the absence of proper guidelines for model selection, getting to the best model in the quickest amount of time with the least number of parameters (model parsimony) seems to be a useful goal. Accordingly the proposed HIS could be a viable alternative to ANFIS. Also, for both ANFIS and HIS, interpretable fuzzy knowledge bases can be derived in terms of distinguishable fuzzy sets and rules. However, as the number of rules increases, as in the case of ANFIS, the interpretation of rules becomes more complex.

[52] The indices presented in Table 1 are computed over the entire range of flow; hence in order to have a closer examination of each model's performance, the computed hydrograph by both the models for a typical flood event from both the basins are depicted along with its observed counterpart in Figure 12. It can be observed from Figure 12 that HIS is preserving the peak flows effectively than the ANFIS, while in low and medium ranges of flow the performance of both models is similar.

[53] The improved performance of HIS over ANFIS can be reinforced from the results presented in Table 2, where the error in peak flows (for a few typical flood events) and error in the volume of hydrograph is depicted. Note that the error in peak flow prediction is less for HIS compared to ANFIS in various ranges of flow. Note that for Narmada basin, the first peak event in the Table 2 (2989.23 m^3/s) is the maximum flow observed in the validation period. It is noted that HIS slightly over predicts all the peaks, while

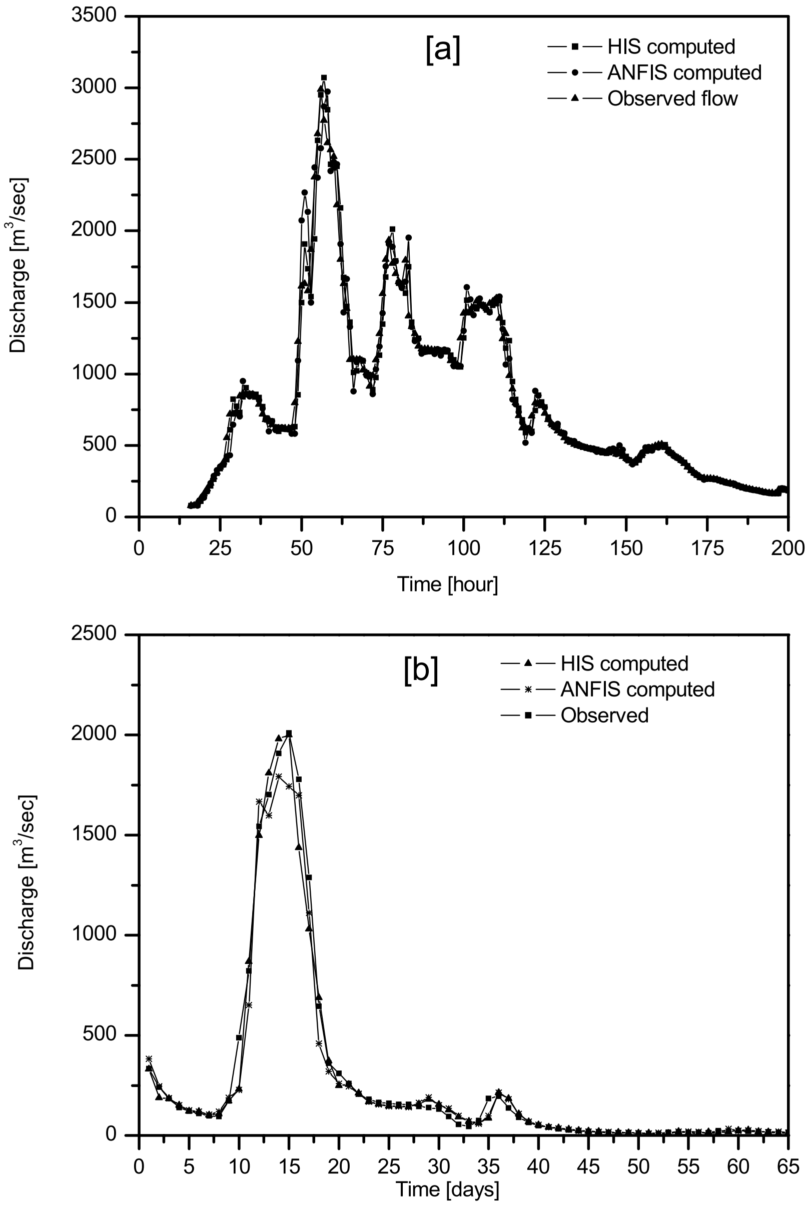

Figure 12. Plot of comparison between HIS and ANFIS model for a typical storm event for (a) Narmada basin and (b) Kentucky basin.

Table 2. Comparison of Model Estimated Hydrograph Characteristics by HIS and ANFIS, for Three Typical Flood Events During Model Validation Period (Narmada Basin and Kentucky Basin)

	HIS		ANFIS	
Observed Peak Flow, m^3/s	Relative Error in Peak Flow, %	Error in Volume Under the Hydrograph, %	Relative Error in Peak Flow, %	Error in Volume Under the Hydrograph, %
		Narmada Basin		
2989.23	2.72	−0.13	3.12	0.33
1868.63	0.98	−0.13	−2.49	−2.44
2519.96	1.00	−0.11	−1.70	−1.70
		Kentucky Basin		
2553.93	−15.35	−0.01	−28.13	0.008
2010.28	−3.58	−0.16	10.87	−0.91
1896.92	1.40	−3.30	−8.52	−119.49

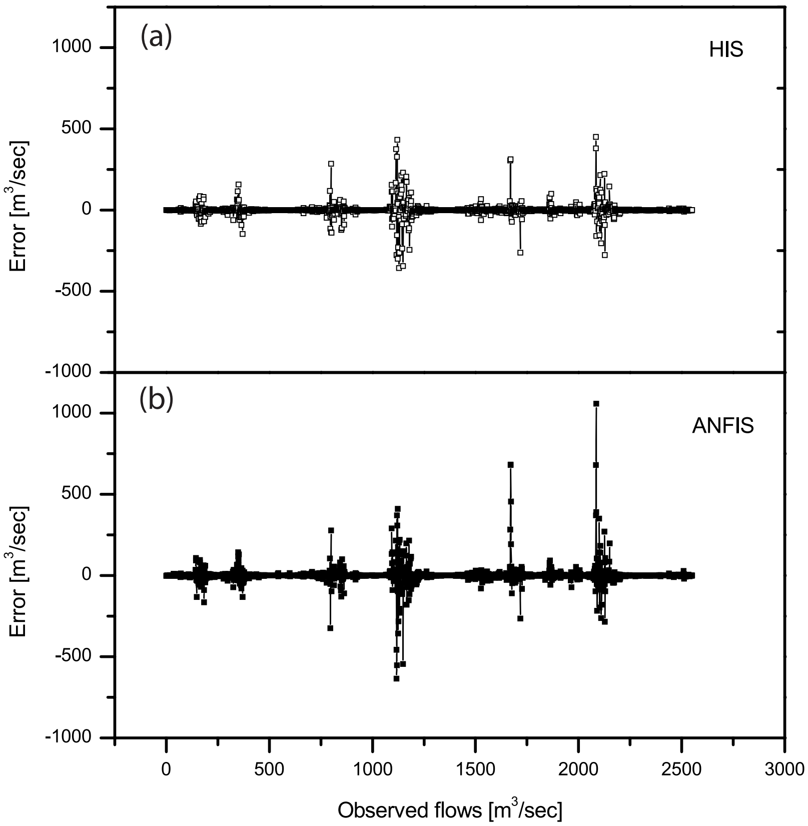

Figure 13. Plot of distribution of error along the magnitude of flow for (a) HIS and (b) ANFIS models during validation period (Narmada basin).

ANFIS underestimates majority of them. The error in estimating peak flow was found to be lesser in the case of HIS compared to ANFIS for Kentucky basin also. It can be observed from Table 2 that a peak flow of 2010.28 m^3/s was computed as 2082.25 m^3/s by the HIS and 1971.63 m^3/s by the ANFIS. The relative error in estimating other peaks were also several times higher for ANFIS as compared to HIS. The results, in general, confirm that the performance of HIS was much superior as compared to ANFIS for both the basins. This observation makes the ANFIS's potential (when grid partitioning is used) questionable for practical application especially in the flood forecasting context. It is evident that the total volume estimated by HIS is much better than ANFIS for all the events considered.

[54] The distribution of forecast errors over the entire range of flow by two models for validation period is presented in Figures 13 and 14 respectively for Narmada and Kentucky basins respectively. It is clear from these figures that the assumption of homoscedastcity of residuals (which is an inherent assumption when mean square error is used as the objective function during parameter estimation) is valid in the case of HIS models in both the case examples. The earlier observation that HIS preserves the peak better, can be further confirmed from Figures 13 and 14.

[55] Apart from the improvement in predicting the peak flow characteristics and parsimonious model for HIS over ANFIS, the HIS takes lesser time in calibration as the number of parameters to be estimated are significantly less for HIS. It is worth mentioning that parameter estimation of the ANFIS model took about 948 s, while HIS was able to learn the process in 563 s on a normal Pentium 4 processor during training for the Narmada case study. This observation of reduction in time for training the model was found to be true for Kentucky basin also.

[56] The results from this study clearly illustrate that the HIS model performs better than the ANFIS model in mapping the rainfall-runoff process. Though the general performance of these models was comparable at a one step ahead forecasts (both hourly as well as daily as illustrated in two case examples), the ANFIS tends to underestimate the peak flows of the hydrograph. It should be noted that the preliminary concepts of both HIS and ANFIS are essentially rooted in the same concepts of fuzzy computing; that is the concept of reconstruction of a single-variable series in a multidimensional phase space to represent the underlying dynamics, and using a local approximation method for making predictions. The only difference between both the models is in terms of the methods employed for the parameter identification. It is clear that since fuzzy computing is based on local approximations, the identification of the local regions play a major role in its performance. This is essentially evident from the current study that the use of a hyperellipsoidal clustering to identify the local regions help HIS to map the process better compared to ANFIS. The foregoing discussions clearly illustrate that the HIS model can be a preferable alternative to ANFIS in modeling the rainfall-runoff process.

7. Summary and Conclusions

[57] This paper presents a new approach to automatically extract fuzzy rules by direct learning from data and build a

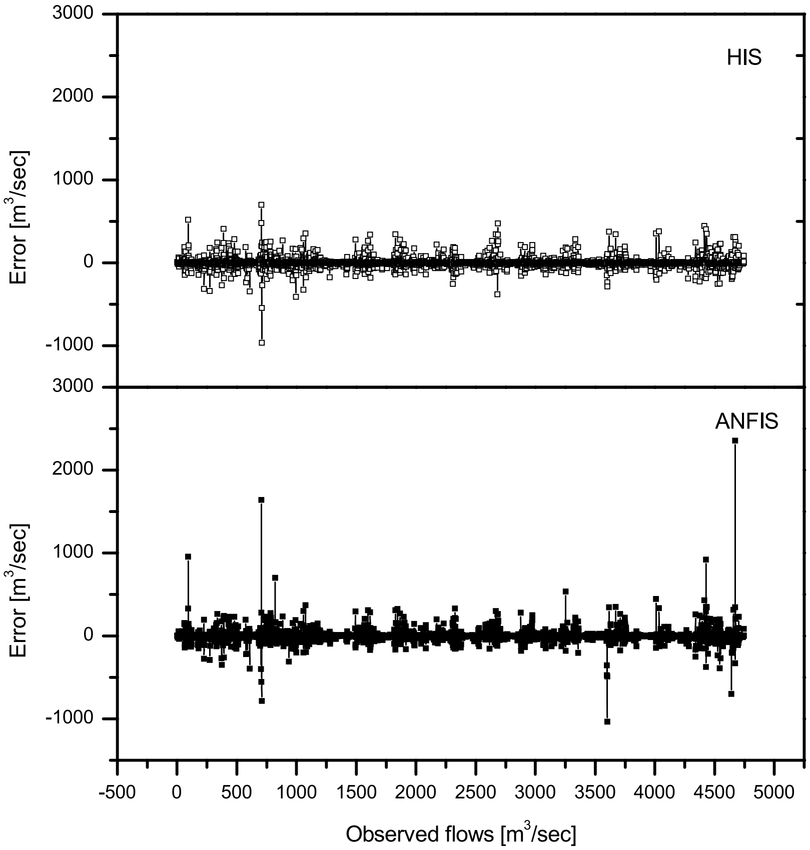

Figure 14. Plot of distribution of error along the magnitude of flow for (a) HIS and (b) ANFIS models during validation period for Kentucky basin.

model that closely represents the process. The approach is based on a neuro-fuzzy model that is designed in such a way that its learning algorithm works in a parameter space with reduced dimensionality. The dimensionality of the reduced space guarantees the generation of human-understandable fuzzy rules. In a conventional fuzzy approach, the MFs and the consequent models are fixed by the model designer according to a priori knowledge. In the present investigation, initialization of the HIS architecture is provided by a hyperellipsoidal fuzzy clustering procedure. Structural learning procedure is applied to find out suitable numbers of fuzzy if-then rules and proper partition of the input space and the Levenberg-Marquardt parametric learning procedure has been used to determine optimal MF. An advantage of the proposed HIS structure is that by using fuzzy set techniques the resulting model is transparent and linguistically interpretable. Therefore the model can be identified with the help of linguistic rules and data gathered from the process. As the proposed HIS model has a lower complexity than other fuzzy models, it does not suffer from the curse of dimensionality (unlike more general fuzzy models) and can be easily implemented.

[58] The potential of the HIS model is illustrated by developing a rainfall-runoff model for two basins: (1) Narmada basin, India, (2) Kentucky basin, USA. A tenfold cross-validation procedure was applied to find out the best model developed from input/output data and the best structure consists of 3 fuzzy if-then rules for Narmada basin and 4 rules for Kentucky basin. The values of three performance evaluation criteria, namely, the coefficient of efficiency, the root-mean-square error, and the coefficient of correlation, were found to be very good and consistent for the HIS model. A comparison between HIS and ANFIS suggests that both may perform similar. The results suggest that the HIS can be a viable alternative to the other, as HIS has certain advantages over ANFIS in terms of having less number of parameters, transparent rules, lesser time requirement etc. It is also noted that HIS simulates the peak flow better than ANFIS, which is of importance in a flood forecasting problem. Further it is also observed that the proposed HIS can be calibrated with lesser time than ANFIS as it makes use of an optimization algorithm for structural identification also.

[59] **Acknowledgments.** We wish to express our sincere gratitude to three anonymous reviewers whose constructive comments and creative suggestions have helped us to improve the quality of this manuscript. We also wish to thank the Indian Institute of Technology Madras, Chennai, for funding and extending facilities to perform this work.

References

Abraham, A. (2001), Neuro-fuzzy systems: State-of-the-art modeling techniques, in *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence*, edited by J. Mira et al., pp. 269–276, Springer, New York.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000a), Artificial neural networks in hydrology I: Preliminary concepts, *J. Hydrol. Eng.*, *5*, 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*, 124–137.

Atiya, A. F., S. El-Shoura, S. I. Shaheen, and M. El-Sherif (1999), A comparison between neural-network forecasting techniques—Case study: River flow forecasting, *IEEE Trans. Neural Networks*, *10*, 402–409.
Babuska, R. (1998), *Fuzzy Modeling for Control*, Springer, New York.
Beven, K. J. (2000), *Rainfall-Runoff Modelling: The Primer*, John Wiley, Hoboken, N. J.
Bowden, G. J., G. C. Dandy, and H. R. Maier (2004), Input determination for neural network models in water resources applications. Part 1—Background and methodology, *J. Hydrol.*, *301*, 75–92.
Box, G. E. P., and G. M. Jenkins (1976), *Time Series Analysis, Forecasting and Control*, Holden-Day, Boca Raton, Fla.
Brown, M., and C. Harris (1994), *Neurofuzzy Adaptive Modelling and Control*, Prentice-Hall, Upper Saddle River, N. J.
Campolo, M., P. Andreussi, and A. Soldati (1999), River flood forecasting with neural network model, *Water Resour. Res.*, *35*, 1191–1197.
Chang, L.-C., F.-J. Chang, and Y.-H. Tsai (2005), Fuzzy exemplar-based inference system for flood forecasting, *Water Resour. Res.*, *41*, W02005, doi:10.1029/2004WR003037.
Chiu, S. (1994), Fuzzy model identification based on cluster estimation, *J. Intell. Fuzzy Syst.*, *2*, 267–278.
Connor, J. T., R. D. Martin, and L. E. Atlas (1994), Recurrent neural networks and robust time series prediction, *IEEE Trans. Neural Networks*, *5*, 240–254.
Coulibaly, P., F. Anctil, and B. Bobee (2000), Daily reservoir inflow forecasting using artificial neural networks with stopped training approach, *J. Hydrol.*, *230*, 244–257.
Coulibaly, P., F. Anctil, and B. Bobee (2001), Multivariate reservoir inflow forecasting using temporal neural network, *J. Hydrol. Eng.*, *6*, 367–376.
Dawson, D. W., and R. Wilby (1998), An artificial neural network approach to rainfall-runoff modeling, *Hydrol. Sci. J.*, *43*, 47–65.
Dawson, D. W., and R. Wilby (2001), Hydrological modeling using artificial neural networks, *Prog. Phys. Geogr.*, *25*, 80–108.
Deka, P., and V. Chandramouli (2003), A fuzzy neural network model for deriving the river stage discharge relationship, *Hydrol. Sci. J.*, *48*, 197–209.
Deka, P., and V. Chandramouli (2005), Fuzzy neural network model for hydrologic flow routing, *J. Hydrol. Eng.*, *10*, 302–314.
Duan, Q., S. Sorooshian, and V. K. Gupta (1992), Effective and efficient global optimization for conceptual rainfall runoff models, *Water Resour. Res.*, *28*, 1015–1031.
Efe, M. O., and O. Kaynak (2001), A novel optimization procedure for training of fuzzy inference systems by combining variable structure systems technique and Levenberg-Marquardt algorithm, *Fuzzy Sets Syst.*, *122*, 153–165.
Gupta, V. K., and S. Sorooshian (1985), The relationship between data and the precision of parameter estimates of hydrologic models, *J. Hydrol.*, *81*, 57–77.
Hornik, K., M. Stinchcombe, and H. White (1989), Multilayer feed forward networks are universal approximators, *Neural Networks*, *2*, 359–366.
Hsu, K., V. K. Gupta, and S. Sorooshian (1995), Artificial neural network modeling of the rainfall-runoff process, *Water Resour. Res.*, *31*, 2517–2530.
Hundecha, Y., A. Bardossy, and H.-W. Theisen (2001), Development of a fuzzy logic based rainfall-runoff model, *Hydrol. Sci. J.*, *46*, 363–377.
Jain, A., and S. Srinivasulu (2004), Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques, *Water Resour. Res.*, *40*, W04302, doi:10.1029/2003WR002355.
Jang, J.-S.-R. (1993), ANFIS: Adaptive network based fuzzy inference system, *IEEE Trans. Syst. Man Cybernetics*, *23*, 665–683.
Kisi, O. (2005), Suspended sediment estimation using neuro-fuzzy and neural network approaches, *Hydrol. Sci. J.*, *50*, 683–696.
Luk, K. C., J. E. Ball, and A. Sharma (2000), A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting, *J. Hydrol.*, *227*, 56–65.
Maier, H. R., and G. C. Dandy (1997), Determining inputs for neural network models of multivariate time series, *Microcomput. Civ. Eng.*, *12*, 353–368.
Maier, H. R., and G. C. Dandy (2000), Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and application, *Environ. Modell. Software*, *15*, 101–124.
Mamdani, E. H., and S. Assilian (1975), An experiment in linguistic synthesis with a fuzzy logic controller, *Int. J. Man-Mach. Stud.*, *7*, 1–13.
McCuen, R. H. (1993), *Microcomputer Applications in Statistical Hydrology*, Prentice-Hall, Upper Saddle River, N. J.
Minns, A. W., and M. J. Hall (1996), Artificial neural networks as rainfall-runoff models, *Hydrol. Sci. J.*, *41*, 399–417.
Moradkhani, H., S. Sorooshian, H. V. Gupta, and P. R. Houser (2005), Dual state-parameter estimation of hydrological models using ensemble Kalman filter, *Adv. Water Resour.*, *28*, 135–147.
Nayak, P. C., and K. P. Sudheer (2007), Fuzzy model identification based on cluster estimation for reservoir inflow forecasting, *Hydrol. Processes*, doi:10.1002/hyp.6644, in press.
Nayak, P. C., K. P. Sudheer, D. M. Rangan, and K. S. Ramasastri (2004), A neuro-fuzzy computing technique for modeling hydrological time series, *J. Hydrol.*, *291*, 52–66.
Nayak, P. C., K. P. Sudheer, and K. S. Ramasastri (2005a), Fuzzy computing based rainfall-runoff model for real time flood forecasting, *Hydrol. Processes*, *19*, 955–968, doi:10.1002/hyp.5553.
Nayak, P. C., K. P. Sudheer, D. M. Rangan, and K. S. Ramasastri (2005b), Short-term flood forecasting with a neurofuzzy model, *Water Resour. Res.*, *41*, W04004, doi:10.1029/2004WR003562.
Olaru, C., and L. Wehenkel (2003), A complete fuzzy decision tree technique, *Fuzzy Sets Syst.*, *138*, 221–254.
Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1994.), *Numerical Recipes in C—The Art of Scientific Computing*, 2nd ed., Cambridge Univ. Press, New York.
Sajikumar, S., and B. S. Thandaveswara (1999), A non-linear rainfall-runoff model using an artificial neural network, *J. Hydrol.*, *216*, 32–55.
See, L., and S. Openshaw (1999), Applying soft computing approaches to river level forecasting, *Hydrol. Sci. J.*, *44*, 763–779.
Sen, Z., and A. Altunkaynak (2003), Fuzzy awakening in rainfall-runoff modeling, *Nord. Hydrol.*, *35*, 31–43.
Setnes, M. (2000), Supervised fuzzy clustering for rule extraction, *IEEE Trans. Fuzzy Syst.*, *8*, 416–424.
Shahin, M. A., H. R. Maier, and M. B. Jaksa (2000), Evolutionary data division methods for developing artificial neural network models in geotechnical engineering, *Res. Rep. R 171*, Dep. of Civ. and Environ. Eng., Univ. of Adelaide, Adelaide, South Aust., Australia.
Shamseldin, A. Y. (1997), Application of a neural network technique to rainfall runoff modeling, *J. Hydrol.*, *199*, 272–294.
Silverman, D., and J. A. Dracup (2000), Artificial neural networks and long-range precipitation in California, *J. Appl. Meteorol.*, *31*, 57–66.
Solomatine, D. P., and K. N. Dulal (2003), Model trees as an alternative to neural networks in rainfall-runoff modeling, *Hydrol. Sci. J.*, *48*, 399–411.
Stone, M. (1974), Cross validation choice and assessment of statistical predictions, *J. R. Stat. Soc. B*, *36*, 44–47.
Sudheer, K. P., A. K. Gosain, and K. S. Ramasastri (2002), A data-driven algorithm for constructing artificial neural network rainfall-runoff models, *Hydrol. Processes*, *16*, 1325–1330.
Sudheer, K. P., P. C. Nayak, and K. S. Ramasastri (2003), Improving peak flow estimates in artificial neural network river flow models, *Hydrol. Processes*, *17*, 677–686.
Sugeno, M., and G. Kang (1988), Structure identification of fuzzy model, *Fuzzy Sets Syst.*, *26*, 15–33.
Takagi, T., and M. Sugeno (1985), Fuzzy identification of systems and its application to modeling and control, *IEEE Trans. Syst. Man Cybernetics*, *15*, 116–132.
Thirumalaiah, K., and M. C. Deo (2000), Hydrological forecasting using neural networks, *J. Hydrol. Eng.*, *5*, 180–189.
Tokar, A. S., and A. Johnson (1999), Rainfall-runoff modeling using artificial neural networks, *J. Hydrol. Eng.*, *4*, 232–239.
Tsukamoto, Y. (1979), An approach to fuzzy reasoning method, in *Advances in Fuzzy Set Theory and Application*, edited by M. M. Gupta, R. K. Ragade, and R. R. Yager, pp. 137–149, Elsevier, New York.
Vernieuwe, H., O. Georgievab, B. De Baets, V. R. N. Pauwels, N. E. C. Verhoest, and F. P. De Troch (2005), Comparison of data-driven Takagi-Sugeno models of rainfall-discharge dynamics, *J. Hydrol.*, *302*, 173–186.
Xiong, L. H., and K. M. O'Connor (2002), Comparison of four updating models for real-time river flow forecasting, *Hydrol. Sci. J.*, *47*, 621–639.
Xiong, L. H., and A. Y. Shamseldin (2001), A nonlinear combination of the forecasts of rainfall-runoff models by the first order Takagi-Sugeno fuzzy system, *J. Hydrol.*, *245*, 196–217.

S. K. Jain, National Institute of Hydrology, Roorkee-247 667, India. (skj@nih.ernet.in)
P. C. Nayak, Deltaic Regional Centre, National Institute of Hydrology, Siddartha Nagar, Kakinada-533 003, India. (nayakpc@yahoo.co.in)
K. P. Sudheer, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai-600 036, India. (sudheer@iitm.ac.in)