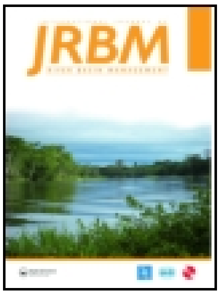

International Journal of River Basin Management
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/trbm20

Hierarchical neurofuzzy model for real-time flood forecasting

Sagarika Rath [a], Purna Chandra Nayak [b] & Chandranath Chatterjee [c]

[a] DHI (India) Water and Environment Pvt. Ltd, 3rd Floor NSIC Bhawan, Okhla Industrial Estate, New Delhi, 110020, India

[b] National Institute of Hydrology, Siddartha Nagar, Kakinada, 533 003, India

[c] Department of Agricultural and Food Engineering, Indian Institute of Technology, Kharagpur, 721 302, India

Accepted author version posted online: 24 Apr 2013. Published online: 06 Jun 2013.

Intl. J. River Basin Management iFirst, 2013, 1–16

Hierarchical neurofuzzy model for real-time flood forecasting

SAGARIKA RATH, *Water Resources Engineer, DHI (India) Water and Environment Pvt. Ltd, 3rd Floor NSIC Bhawan, Okhla Industrial Estate, New Delhi 110020, India.*

PURNA CHANDRA NAYAK, *Scientist 'D', National Institute of Hydrology, Siddartha Nagar, Kakinada 533 003, India. Email: nayakpc@yahoo.co.in (Author for correspondence)*

CHANDRANATH CHATTERJEE, *Associate Professor, Department of Agricultural and Food Engineering, Indian Institute of Technology, Kharagpur 721 302, India.*

ABSTRACT

The current study employs a hierarchical adaptive network-based fuzzy inference system for flood forecasting by developing a rainfall–runoff model for the Narmada basin in India. A hybrid learning algorithm, which combines the least-square method and a back propagation algorithm, is used to identify the parameters of the network. A subtractive clustering algorithm is used for input space partitioning in the fuzzy and neurofuzzy models. The model architectures are trained incrementally each time step and different models are developed to predict one-step and multi-step ahead forecasts. The number of input variables is determined using a standard statistical method. An artificial neural network (ANN) model which uses an Levenberg–Marquardt (LM) back-propagation training algorithm has been developed for the same basin. The results of this study indicate that the hierarchical neurofuzzy model performs better compared to an ANN and the standard fuzzy model in estimating hydrograph characteristics, especially at longer forecast time horizons.

Keywords: hierarchical neurofuzzy model; Takagi–Sugeno fuzzy model; subtractive clustering; flood forecasting

1 Introduction

One of the major concerns that the human society faces is the risk of disasters related to water resources, viz. floods, droughts, pollution, consequences of likely climate changes, etc. India has to confront severe floods due to intense precipitation, primarily during the monsoon period (during the months from June to September), year after year.

The National Commission for Integrated Water Resources Development studied the flood problem in India in detail and concluded that there is no possibility for complete protection against floods (NCIWRD 1999). It therefore suggested that emphasis should be placed on better management of flood plains, flood forecasting, flood preparedness and flood insurance. Flood management is a complex process, requiring the simultaneous consideration of the hydrologic, hydraulic, structural, geotechnical, environmental, social, economic and behavioural aspects of flooding. River flow forecasting is an effective non-structural measure for flood management and can reduce the damages due to flooding to a greater extent. River flow forecasts provide advance warning of expected river stages during floods and are of immense utility in saving life and property. These are also helpful in regulating reservoirs during floods.

Floods are natural phenomena and are inherently complex to model. Flood forecasting is achieved by developing rainfall–runoff models. The relationship between rainfall and runoff is very complex and depends on rainfall and watershed characteristics. The rainfall–runoff process exhibits a higher degree of temporal and spatial variability and is further plagued by nonlinearity of the physical process, conflicting spatial and temporal scales, and uncertainties in parameter estimation. Various statistical models have been proposed for flood forecasting, such as statistical correlation techniques, autoregressive moving average (ARMA) models, etc. Physically based models may be more accurate for flood forecasting, but these models require substantial amount of physical data for their successful operation.

Artificial neural network (ANN) and the fuzzy inference system (FIS) models are two popular data-driven modelling (DDM) techniques extensively used for hydrologic forecasting. ANNs are nonlinear regression techniques and are well suited for hydrologic modelling (Connor *et al.* 1994, Atiya *et al.* 1999, Babovic and Keijzer 1999, Tayfur 2006, Tayfur and Singh 2006, Tayfur *et al.*

Received 30 October 2012. Accepted 18 April 2013.

ISSN 1571-5124 print/ISSN 1814-2060 online
http://dx.doi.org/10.1080/15715124.2013.798329
http://www.tandfonline.com

2007), as they can approximate virtually any (measurable) function up to an arbitrary degree of accuracy (Hornik *et al.* 1989). The emergence of neural network technology has provided many promising results in the field of hydrology and water resource simulation (ASCE Task Committee on Application of Artificial Neural Networks in Hydrology 2000a, 2000b, Dawson and Wilby 2001). Similarly, FIS can be seen as a logical DDM approach, which uses if–then rules and logical operators to establish qualitative relationships among the variables in the model. Fuzzy sets serve as a smooth interface between qualitative variables involved in the rules and the numerical domains of the inputs and outputs of the model. The rule-based nature of the fuzzy models allows their use of information expressed in the form of natural language statements, and makes the models transparent to interpretation and analysis. The last decade has witnessed a number of applications of fuzzy logic in hydrologic forecasting (See and Openshaw 1999, Hundecha *et al.* 2001, Xiong *et al.* 2001, Xiong and O'connor 2002, Sen and Altunkaynak 2003, Chang *et al.* 2005, Nayak *et al.* 2005a, Vernieuwe *et al.* 2005, Tayfur 2006, Tayfur and Singh 2006, 2011, Nayak 2010).

Some recent studies have tried to address the limitations of FIS (and ANN) by employing various statistical or conceptual modelling approaches while developing FIS (or ANN) models. For example, See and Openshaw (1999) proposed a hybrid model for improving the forecast accuracy of ANN models, which is based on splitting the data into different subsets before training and developing individual ANN models for each subset. Output of these networks is then recombined via the rule-based fuzzy logic model that has been optimized using a genetic algorithm (GA). They demonstrated the potential of the model on the Ouse River catchment in northern England, and reported that the hybrid model was better in performance than an Auto ARMA model and a naïve model for water level forecasting. Nonetheless, the proposed hybrid model failed to capture the peak flow characteristics. Later, See and Openshaw (2000) combined four individual conventional river flow forecast models (ranging from neural network to statistical models) in order to improve forecast accuracy. The combining of forecasts was achieved through Bayesian and fuzzy logic techniques. The results indicated that the addition of fuzzy logic to crisp Bayesian approaches resulted in better forecasts compared to other combining methods for the Ouse River catchment in northern England. However, in this case too, the peak flow representation by the combined model was poor.

A fuzzy logic-based conceptual rainfall–runoff model was proposed by Ozelkan and Duckstein (2001). They reported that calibration of the fuzzy conceptual rainfall–runoff models using fuzzy least-squares regression technique, with fuzziness introduced into both rainfall and runoff yielded usually more stable parameter estimates than those obtained assuming nonfuzziness or crispness of the rainfall and runoff. Chang *et al.* (2001) suggested a fusion of neural network and fuzzy arithmetic in a counter-propagation fuzzy neural network (CFNN) and applied this to the Da-chi River at Nanhusi in central Taiwan for real-time flood forecasting. The performance of the CFNN was reported to be good in dealing with reconstruction of streamflow.

Hundecha *et al.* (2001) developed the fuzzy rule-based models to simulate different physical processes involved in the generation of discharge from rainfall for River Neckar in southwest Germany, and compared them with a conceptual model Hydrologiska Byråns Vattenbalansavdelning (HBV). They reported that the fuzzy logic approach of modelling catchment processes was able to reproduce the observed discharge well, but under low and normal flow conditions, no noticeable difference was observed between the HBV and the fuzzy rule-based model. Chang *et al.* (2001) adopted a new approach to improve real-time reservoir operation. The approach combines a GA and an adaptive network-based fuzzy inference system (ANFIS). The GA was used to search the optimal reservoir operating histogram based on a given inflow series, which can be recognized as the base of input–output training patterns in the next step. The ANFIS was then built to create the fuzzy inference system, to construct a suitable structure and parameters, and to estimate the optimal water release according to the reservoir depth and inflow situation. Ludermir and Valenga (2002) presented a fuzzy neural network model for inflow forecast for the Sobradinho Hydroelectric power plant, part of the Chesf (Companhia HidrelC-trica do Siio Francisco-Brazil) system. The model was implemented to forecast monthly average inflow on a one-step-ahead basis. Nayak *et al.* (2004, 2005b) applied ANFIS to hydrologic time series modelling, in the river flow forecasting. Ulke *et al.* (2009) applied ANFIS model for suspended sediment load estimation.

Although previous studies report a promising result in streamflow modelling, by developing fuzzy or neurofuzzy techniques, attention is required to optimize the number of if–then rules. Since FIS or ANFIS perform piecewise approximation of the process at a local level, the effective partitioning of the input space is important. Yet another concern is in the input partitioning methods that are commonly employed, such as grid partitioning or scatter partitioning. 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 a specific region of the input space. Models may suffer the 'curse of dimensionality' when they handle more input vectors (typically more than six inputs) with lengthy data sets. 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 the learning and application phases, thereby reducing the number of parameters which in turn reduces the uncertainty. To eliminate the problems associated with grid-partitioning, scatter partitioning of the input space into rule patches has been proposed (Chiu 1994, Setnes 2000). In this partitioning, the antecedent parts of the fuzzy rules are positioned at arbitrary locations in the input space.

In the current study, FIS and ANFIS models are developed and a subtractive clustering (SC) technique is applied to find out the optimum number of if–then rules from the input–output data pair. The paper is organized as follows. First, we give a brief overview of the Takagi–Sugeno (TS) fuzzy and neurofuzzy models integrated with SC. Following this, the development of fuzzy and neurofuzzy models for flood forecasting is presented. An ANN model has also been developed for comparison. The results are analysed and discussed in the subsequent sections. The conclusions are then presented after an evaluation of the performance by these models in inflow forecasting.

2 Statement of the problem (in fuzzy modelling approach)

Standard fuzzy systems are generally applicable, in the sense that they are universal approximators, and can approximate arbitrary continuous functions to any accuracy. Because of their transparency and general applicability, substantial progress both in the theory and application of fuzzy systems has been achieved during the last four decades. However, as fuzzy systems have been applied to more complicated and high-dimensional systems, the 'curse of dimensionality' has become increasingly apparent as the bottleneck to wider application. The curse of dimensionality of fuzzy systems can be summarized as follows:

Rule dimensionality: The total number of rules in the fuzzy rule base increases exponentially with the number of input variables.

Parameter dimensionality: The total number of parameters in the mathematical formulas of fuzzy systems increases exponentially with the number of input variables.

Data or information dimensionality: The number of data or knowledge set required to identify fuzzy systems increases exponentially with the number of input variables.

As a consequence of the curse of dimensionality, transparency and interpretation (important advantages of fuzzy systems) is damaged as humans are incapable of understanding and justifying hundreds or thousands of fuzzy rules and parameters. Further, as there are often only limited data available for applications, a large number of rules and parameters result in over-fitting, which destroys the generalization ability of fuzzy systems.

The bottleneck for FIS development is the identification of the antecedent parameters, which includes the optimal number of if–then rules and the shape of the membership function (MF). If we know the shape of the function being approximated (as in the earlier example) it would be easy to identify the model parameters. This is done generally with the help of 'expert knowledge'. However, this can be done only in the case of 'decision support systems' development. In the case of 'function approximation' this method may not be reliable. If the number of MFs is large (implying large number of fuzzy rules), the system requires large computation time for inference. Moreover, the huge rule base may over-fit the system and cause the system to lose the capability of generalization. If a system requires n input variables each partitioned into m MFs, the total number of rules required to model the system by using one single fuzzy inference system is m^n. As the complexity of the problem increases, the number of required inputs increases too, requiring an exponentially larger number of rules. In order to deal with the problem rule-explosion, the development of hierarchical fuzzy systems has been proposed. In hierarchical systems, the number of rules increases linearly with the number of inputs rather than exponentially.

The partitioning process of the input/output space has a great influence on the performance of a fuzzy and neurofuzzy system in relation to its desirable features (accuracy, generalization, automatic generation of rules, etc.). 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 (Chiu 1994, Setnes 2000). In this type, the antecedent parts of the fuzzy rules are positioned at arbitrary locations in the input space. In the current investigation, the SC algorithm is used as the effective partitioning algorithm in a fuzzy and neurofuzzy model for derivation of if–then rules and models are applied to flood forecasting. A mathematical detail on SC is discussed in the next section.

3 Methodology

This section deals with the description of FIS, ANFIS and ANN in general and the methodology for the development of the three models for real-time flood forecasting at the Mandla gauging site of the Narmada River basin in India.

3.1 FIS model development

Fuzzy inference systems are composed of a set of *IF–THEN* rules. A TS fuzzy model has the following form of fuzzy rules:

$\boldsymbol{R_j}$: if x_1 is A_{1j} and x_2 is A_{2j} and … x_n is A_{nj} then $y = g_j(x_1, x_2, \ldots, x_n)$ $(j = 1, 2, 3, \ldots, \mathrm{N})$

where $g(.)$ is a crisp function of x_i.
Usually, $g_j(x_1, x_2, \ldots, x_n) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n$.
The overall output of the fuzzy model can be obtained by

$$y = \frac{\sum_{j=1}^{N} g_j(.) T_{i=1}^{m_j} \mu_{ij}(x_i)}{\sum_{J=1}^{N} T_{i=1}^{m_j} \mu_{ij}(x_i)}, \tag{1}$$

where $1 \le m_j \le n$ is the number of input variables that appear in the rule premise, N is the number of fuzzy rules, n is the number of inputs, μ_{ij} is the MF for fuzzy set and T is a T-norm for fuzzy conjunction. The TS–FS is a single-stage fuzzy system. It is important to partition the input space using some clustering, grid partitioning, etc. The shapes of MFs in the antecedent parts, and the free parameters in the consequent parts are also to be determined using some adaptive techniques. In the current analysis, a SC algorithm is used to partition the input space and the least-square error method is used for consequent parameter optimization. Brief discussion about the SC algorithm is given below.

The SC method (Chiu 1994) is an extension of the mountain clustering method (Yager and Filev 1994), where the potential is calculated for the data rather than the grid points defined on the data space. As a result, clusters are elected from the system training data according to their potential. The SC compared with mountain clustering has an advantage that there is no need to estimate a resolution for the grid.

SC algorithm uses data points as candidates for cluster centres. Data matrix with n data point $\{x_1, x_2, \ldots, x_n\}$in k dimensional space includes also the output and it is normalized within the hypercube. It is considered that each data point is a potential cluster centre and defines a measure of the potential of the data point x_i as

$$P_i = \sum_{j=1}^{n} \mathrm{e}^{-\alpha \|x_i - x_j\|^2}, \tag{2}$$

where $\alpha = 4/r_a^2$ and the data point with many neighboring data points will have a high potential value. The constant r_a is effectively the radius defining a neighborhood; the data point outside this radius has little influence on the potential. After the potential of every data point has been computed, the data point with the highest potential as the first cluster is selected. Let x_1^* be the location of the first cluster centre and p_1^* be its potential value. Then the potential of each data point x_i may be revised by formula

$$p_i = p_i - p_1^* \mathrm{e}^{-\beta \|x_i - x_1^*\|^2}, \tag{3}$$

where $\beta = 4/r_b^2$ and r_b is a positive constant. After reduction is done, the data point with the highest potential is selected as cluster x_2^*. Next, it selects the data point with the highest remaining potential as the second cluster centre. Similarly, it reduces the potential of each data point according to their distance to the second cluster centre. The process is repeated until a given threshold for the potential is obtained such that $(P_k^*/P_1^*) < \varepsilon$. The choice of ε is an important factor affecting the results; if ε is too large, too few data points will be accepted as cluster centres and if ε is too small, too many cluster centres will be generated. It is important to note that the influence of neighboring data decays exponentially with the square of the distance instead of the distance itself. So for a compact cluster (small r_a), a lot of data points are close to its centre. Each cluster centre x_i^* may be considered as a fuzzy rule that describes the system behaviour. Given an input vector y, the degree to which rule i is fulfilled is defined as

$$\mu_i = \mathrm{e}^{-\alpha \|y - y_i^*\|^2}, \tag{4}$$

where α is the constant defined by Eq. 2. The output vector may be computed using Eq. 1. Eqs 1 and 4 provide a simple and direct way to translate a set of cluster centres into a TS fuzzy model.

In SC, the procedure automatically determines the number of clusters, which is related to the value of r_a. A larger value of r_a leads to a large radius for each cluster and all data can be clustered in a few cases. In contrast, a small value of r_a means more compact clusters but the number of classes will be larger. The cluster radius may be fixed after several trials. It may be noted that the radius specifies the range of influence of the cluster centre of each input and output dimension. Assuming that the cluster radius falls within the hyper box of the unit dimension, a smaller cluster radius will usually yield more clusters in the data, and hence more number of rules. Simultaneously, it increases the model complexity and hence decreases the parsimony.

3.2 Neurofuzzy model development

ANFIS is a powerful universal approximator for vague and fuzzy systems (Lin and Chen 2006). The basic structure of adaptive network consists of three main conceptual parts: a FIS which includes three components: a rule base, a data base, a reasoning mechanism and a learning mechanism consisting of a multilayer feed-forward network (Nayak *et al*. 2004). The adaptive network based on the Sugeno fuzzy inference model provides a deterministic system of output equations and thus the parameters can be easily estimated (Takagi and Sugeno 1985). The architecture of ANFIS consists of five layers (Figure 1). A brief sketch of the operation of these layers is as follows.

The first layer (input nodes) is composed of nodes which generate membership grades according to the appropriate MFs and the set of parameters to minimize are determined accordingly. In the second layer (rule nodes), the AND or the OR operator is applied to compute every possible conjunction of the decision rules. In the third layer (average nodes), the main purpose is the normalization of the conjunctive MFs in order to rescale the inputs. The fourth layer (consequent node) is a standard perceptron in which the Sugeno fuzzy inference model is used to associate the normalized MF towards the total output, in other words, the contribution of each rule is computed. For more details on the ANFIS model development, readers are referred to Nayak *et al.* (2004, 2005b).

3.3 ANN model development

ANNs are massively parallel systems composed of many processing elements connected by links of variables weights. The ANN is characterized by its architecture that represents the pattern of connection between the nodes, its method of determining the connection weights and the activation function. The ANN consists of a number of artificial neurons known as *processing elements* or *nodes*. Each node performs a mapping of its inputs to its output in a three-step process: firstly, it calculates the sum of the activation of its inputs, and then decides its new activation level based on the derived sum, and finally generates an

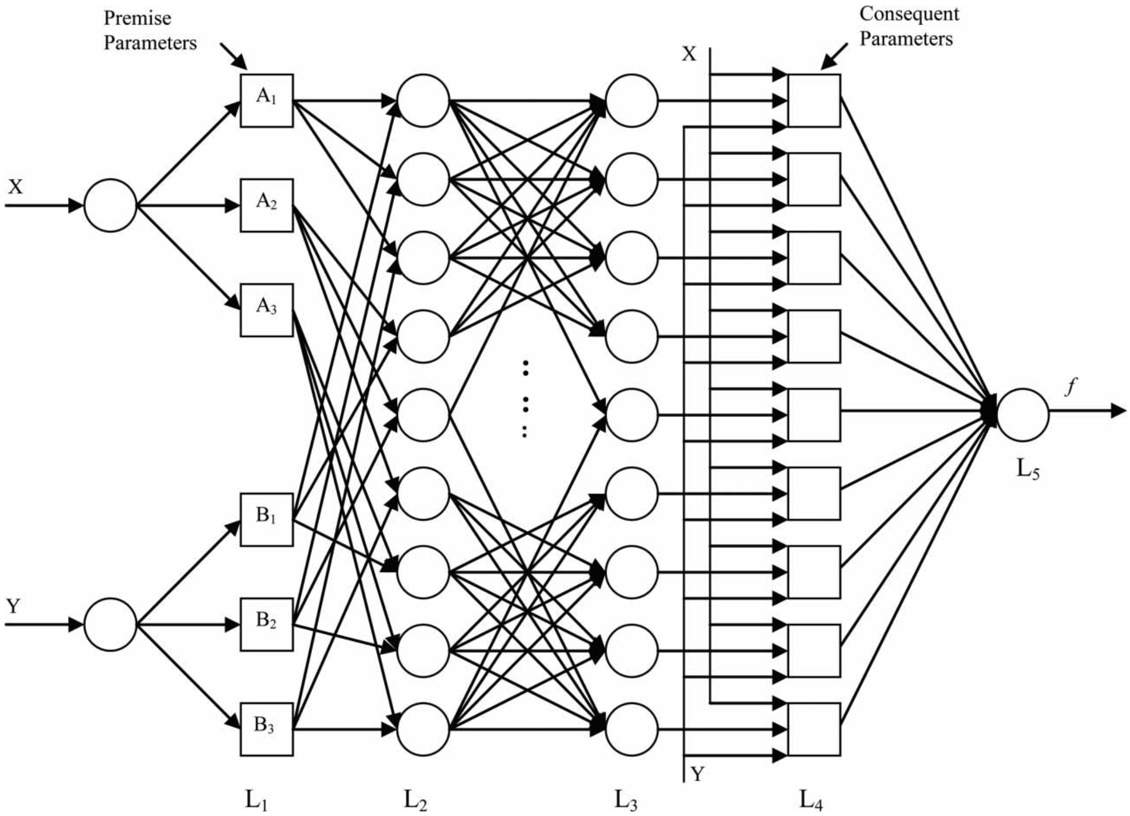

Figure 1 ANFIS architecture.

output signal corresponding to the new level. Each neuron in a specific layer is connected to many other neurons via weighted connections. The weights determine the strength of the connections between interconnected neurons.

A major concern in the development of a neural network is determining an appropriate set of weights that make it perform the desired function. There are many ways that this can be done; the most popular class of these algorithms is based on supervised training. Supervised training starts with a network comprising an arbitrary number of hidden neurons, a fixed topology of connections, and randomly selected values for weights. The network is then presented with a set of training patterns, each comprising an example of the problem to be solved (the inputs) and its corresponding solution (the targeted output). Each problem is input into the network in turn, and the resultant output is compared to the targeted solution providing a measure of total error in the network for the set of training patterns. Properly trained back propagation network give reasonable results when presented with new input during validation. In the process of model development, several network architectures with different number of input neurons in input layer with varying number of hidden neurons are considered to select the optimal architecture of the network. A trial and error procedure based on the minimum error during validation is used to select the best network architecture.

The Levenberg–Marquardt algorithm is a modification of the classic Newton algorithm for finding an optimum solution to a minimization problem. It is designed to approach second-order training speed and accuracy without having to compute the Hessian matrix. The Hessian matrix contains second derivative of network errors with respect to network weights and Jacobeans matrix contains a first derivative of the network error matrix with respect to weights. When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximated as $H = J^{\mathrm{T}}J$ and the gradient can be computed as $g = J^{\mathrm{T}}\,e$, where J is the Jacobean matrix and e is a matrix of network errors. The Jacobean matrix can be computed through a standard back propagation technique that is much less complex than computing the Hessian matrix. So, in the Levenberg–Marquardt Algorithm, the Hessian matrix is approximated in terms of the Jacobean matrix following a Newton-like update in the following ways.

$$x_{k+1} = x_k - [J^{\mathrm{T}}J + \mu I]^{-1}J^{\mathrm{T}}e, \tag{5}$$

$$J^{\mathrm{T}}J = H, \tag{6}$$

$$J = \nabla E, \tag{7}$$

$$J^{\mathrm{T}}e = g, \tag{8}$$

where x is the weights of neural network and μ is the learning parameter that controls the learning process. When μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small-step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Overall, second-order nonlinear optimization techniques are usually faster and more reliable. Therefore, in this study, the Levenberg–Marquardt algorithm is used for multilayer perceptron training.

4 Study area and data availability

In the present study, the Narmada River basin up to the Mandla gauging site forms the study area. The rainfall and runoff data of Narmada River basin have been used to develop the soft computing models (FIS, ANFIS and ANN) for real-time flood forecasting at the Mandla gauging site.

The Narmada River emanates at Amarkantak in the Shahdol district of Madhya Pradesh in Central India at an elevation of 1057 m above the mean sea level. The Narmada basin extends over an area of 98,796 km^2 and lies between longitudes 72° 32′ E–81° 45′ E and latitudes 21° 20′ N–23° 45′ N. In the present study, the upper Narmada basin up to the Mandla site, covering a catchment area of 13,120 km^2 has been selected. A location map of the study area in the Narmada River basin and the upstream gauging stations are shown in Figure 2. The subbasin of the Narmada River up to the Manot gauging site lies between East longitudes 80° 24′–81° 70′ and North latitudes 22° 26′–23° 18′ with most of the part lying in the Mandla district and some part in Shahdol district of Madhya Pradesh in India. The basin occupies an area of 4980 km^2 and length of the river is about 269 km. The topographical and hydrometeorological details are reported in Nayak and Sudheer (2008). The rainfall and runoff data available for the monsoon season (June–October) during the years 1989–1993 on an hourly interval have been used in the study. The rainfall data are available in the form of aerial averages for the entire basin. Thus input data for a developed model consists of: (i) hourly runoff and hourly rainfall data at the Mandla gauging site and (ii) hourly runoff data at 4 upstream gauging sites: Manot, Mohgaon, Hridyanagar and Dindori.

5 Input vector selection and model evaluation

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 analysing these correlogram plots, the significant lags of independent variables that are potentially influencing the output (dependant variable) can be identified. The identified input vector according to Sudheer *et al*. (2002) included a total number of 16 variables (Nayak *et al*. 2005a).

Sudheer *et al*. (2003) suggest that, by following the guidelines used in traditional statistical modelling, the model performance can be improved in the case of soft computing-based models. 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, then suitable transformations to normality have to be applied. Data transformations often are used to simplify the structure of the data so that they follow a convenient statistical model (Sudheer *et al*. 2003). In the current study, log-normal transformation is used and the deterministic component in the runoff and rainfall series was removed prior to modelling. The variables are scaled to fall between 0 and 1 as the activation function warrants. The total available data have been divided into two sets, i.e. a calibration set and a validation set; the parameters of the model are identified using the calibration data set and the model is tested for its performance on the validation data set. The resulting hydrograph was analysed statistically using various performance indices. The goodness of fit statistics considered are the Root mean square error (RMSE), between the computed and observed runoff, the coefficient of correlation (CORR), the model efficiency (EFF), mean absolute error (MAE) and mean absolute relative error (MARE). In addition to these indices, a certain event-specific evaluation measures are also employed to evaluate the model performance such as the percentage error in peak flow estimation and the time difference to peak flow. Finally, the error distribution at different threshold levels for all the models is also compared.

6 Results and discussions

This section deals with the results obtained from FIS, ANFIS and ANN models for real-time flood forecasting at the Mandla gauging site of the Narmada River basin in Madhya Pradesh. The same input variables have been used for FIS, ANFIS and ANN as discussed in the previous chapter. The results obtained are in accordance with the data used and methodology presented in the previous chapters.

6.1 Optimal model structure

The three models (FIS, ANFIS and ANN) are developed for predicting the hourly runoff at the Mandla gauging station (for 1–6 hour lead time forecast) as a function of rainfall and runoff data at the Mandla gauging site and other gauging stations described in the previous section. Various network configurations with different network parameters are trained and tested using these models. The error parameters are computed for each model architecture. The best model architecture is selected on the basis of the overall performance of the model during validation.

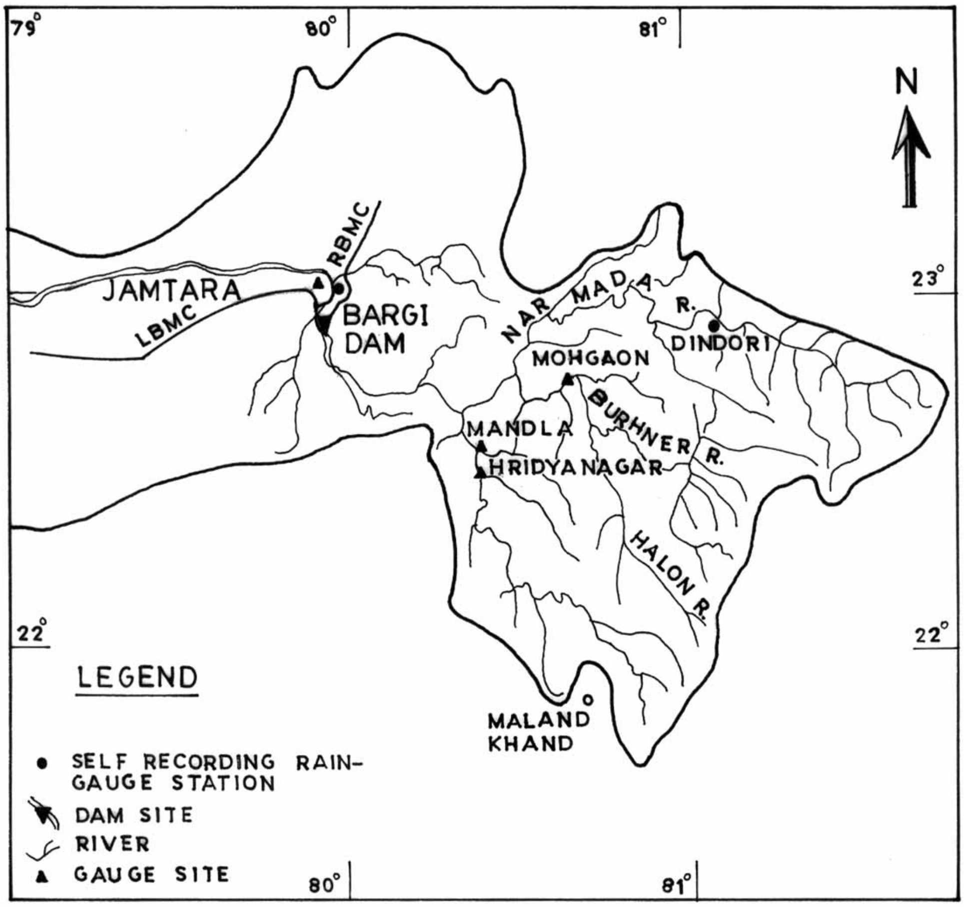

Figure 2 Basin map of river Narmada up to Bargi reservoir.

In the FIS development, the input data are partitioned using SC. In the analysis, the optimal cluster radius is different for different lead times of forecasting. According to the rule formation and the error calculation, the final cluster radius is selected by trial and error basis. The performance of the model at different cluster radii is given in Table 1. From Table 1, it is observed that at 0.45 cluster radius, the training shows better results and at 0.35 cluster radius, the testing result is better. If we compare both radii, then there is no large difference in error between the two cluster radii for training. Hence, a 0.35 cluster radius is selected for 1 hour lead time to forecast. In this way, the cluster radius is selected for a higher forecast horizon also.

In the ANFIS model development, the premise parameters (the shape of the member functions (MFs)) are optimized using the back propagation (gradient descent) algorithm and consequent parameters are optimized using the least-square technique. Gaussian MFs are selected for the model and input space partitioning is carried out using the SC algorithm. ANFIS makes use of a mixture of back propagation (to learn the premise parameters) and the least mean square estimation (to determine the consequent parameters). The ANFIS model structure has been identified by changing the cluster radius from 0.1 to 1 at an increment of 0.05 (similar to FIS) and a model performance is estimated. After several trails, the cluster radius is fixed as 0.35 for a 1 hour lead forecasting and optimal number of rules corresponding to 0.35 is 7.

As discussed, an ANN model which uses the LM training algorithm has been developed for the same input data used for the development of FIS and ANFIS models. Single hidden layer with log-sigmoid function nodes is used in the ANN. The linear activation function is considered in the output layer. As the log-sigmoid transfer function has been used in the model, the input–output data have been scaled appropriately to fall within the function limit. The trial and error procedure started with two hidden neurons initially, and the number of hidden neurons was increased up to 10 with a step size of 1 in each trial. For each set of hidden neurons, the network was trained in batch mode to minimize the mean square error at the output layer. The training was stopped when there was no significant improvement in the efficiency, and the model was then tested for its generalization properties. The parsimonious structure that resulted in minimum error and maximum efficiency during training as well as testing was selected as the final form of the ANN model. The learning rate and momentum factor are fixed as 0.01 and 0.1, respectively, after several experiments. The final structure of the ANN model is: 16 input neurons, 9 hidden neurons and 1 output neuron.

Table 1 Performance indices for FIS for different cluster radius for 1 hour lead forecasting.

		Training				Testing			
Cluster radius	Rules	CORR	EFF (%)	RMSE (m^3/s)	MAE (m^3/s)	CORR	EFF (%)	RMSE (m^3/s)	MAE (m^3/s)
0.5	4	0.998	99.74	38.45	7.18	0.997	99.47	31.59	6.45
0.45	5	0.998	99.77	35.74	7.19	0.997	99.44	32.43	6.79
0.4	5	0.998	99.69	41.61	7.29	0.996	99.22	38.28	7.07
0.35	8	0.998	99.72	39.55	7.18	0.997	99.47	31.35	6.58

6.2 *Evaluation of selected models*

The values of the performance indices for the 1 hour ahead forecast of all three models are presented in Table 2. It is observed from the table that correlation statistics, which evaluate the linear correlation between the observed and computed runoff, is consistent for all models during calibration as well as the validation period. It is seen from the table that during testing, the correlation coefficient is 0.996 for the ANN model and 0.997 for the other two models. The model efficiency that evaluates the capability of the model in predicting runoff values away from the mean is found to be more than 99% during the calibration and validation periods for all models. The RMSE statistic, which indicates a measure of the model error in units of the variable, is good for all models as is evident from the low values. During model testing, ANFIS predicts 31.35 m^3/s RMSE and FIS prediction is also very close to ANFIS (31.81m^3/s). Overall, for a 1-hour ahead forecast, the performance of all the models is comparable, but the RMSE estimated by the ANN model is slightly higher than the other two models. From Table 2, it is found that ANFIS estimated the lowest error (MAE and MARE indices) compared to ANN and FIS models during the training and testing period. It may be noted that ANFIS has seven rules and FIS has eight rules for a 1 hour lead forecasting. Though there is reduction in the number of rules, ANFIS results in similar performance as that of FIS.

6.3 *Forecasting at higher lead time (>1 hour)*

The values of global evaluation measures during the calibration and validation period for three models along the different forecast horizon are summarized in Table 3. The predicted value from the FIS model up to 6 hour lead time is presented in Table 3, and it is observed that the model efficiency is more than 90%. It is also observed that the RMSE value increases as the forecast hour increases. The RMSE value increases from 31.81 to 150.50 m^3/s for the validation period. In Table 3, a number of rules generated by the clustering method is also presented. It is found that the RMSE value for the ANFIS model ranges from 31.35 to 148.81 m^3/s. The rules generated in the ANFIS models for a higher lead hour are quite less as compared with the FIS model due to the hybrid algorithm used in the optimization of flood forecasting. It is observed that for a 1-hour ahead forecast, the performance of all the models is comparable. However, it is worth mentioning that as the lead time increases, the ANFIS model has the lowest RMSE compared to the FIS and ANN model. From Table 3, it is found that during training, the performance indices of all the three models declines as lead time increases. The RMSE obtained during training lies in the range of 39.45–213.70 m^3/s for FIS. FIS shows the highest RMSE for a 6 hour lead time during the training period (213.70 m^3/s).

ANN shows comparatively a less error during training because of the vigorous learning of the training pattern. From Table 3, it is observed that ANN predicts 332.29 m^3/s RMSE at a 6 hour lead time during the validation period. The performance of ANFIS and FIS are quite close to each other and it is hard to decide which model is the best one for the flood forecasting. Again, from the efficiency statistics, it is observed that the ANN model performance is very poor (40.91%) at a 6 hour lead time as compared to the other two models.

In order to reinforce the observation and findings, MARE indices are estimated for a 1–6 hour lead time forecasting for three developed models and results are presented in Table 3. It

Table 2 Statistical indices for different models for 1 hour lead forecasting.

Performance indices	Training			Testing		
	FIS	ANFIS	ANN	FIS	ANFIS	ANN
CORR	0.998	0.998	0.998	0.997	0.997	0.996
EFF (%)	99.72	99.72	99.63	99.46	99.47	99.07
RMSE (m^3/s)	39.45	39.54	45.32	31.81	31.35	41.78
MAE (m^3/s)	7.18	6.23	22.53	6.58	6.54	20.33
MARE	0.017	0.015	0.079	0.026	0.026	0.084

Table 3 Performance indices for higher lead forecast.

				Training				Testing			
Models	Lead time	Cluster radius	Number of rules	CORR	EFF (%)	RMSE (m^3/s)	MARE	CORR	EFF (%)	RMSE (m^3/s)	MARE
FIS	1	0.3	8	0.999	99.72	39.45	0.017	0.997	99.46	31.81	0.026
	2	0.3	8	0.995	99.02	74.28	0.030	0.993	98.47	53.48	0.050
	3	0.3	8	0.991	98.01	105.79	0.043	0.986	97.08	73.86	0.073
	4	0.5	4	0.982	96.08	148.3	0.058	0.975	94.91	97.53	0.097
	5	0.45	5	0.974	94.69	172.68	0.072	0.96	91.91	122.92	0.123
	6	0.3	7	0.961	91.86	213.7	0.087	0.94	87.88	150.5	0.156
ANFIS	1	0.35	7	0.999	99.72	39.55	0.015	0.997	99.47	31.35	0.026
	2	0.5	4	0.996	99.16	68.81	0.027	0.993	98.55	51.98	0.049
	3	0.4	5	0.992	98.41	94.5	0.039	0.986	96.91	75.93	0.072
	4	0.4	5	0.988	97.5	118.5	0.053	0.975	94.75	99.02	0.089
	5	0.35	5	0.977	95.19	164.29	0.062	0.967	93.29	111.97	0.118
	6	0.4	5	0.971	94.19	180.58	0.067	0.942	88.15	148.81	0.152
ANN	1			0.998	99.63	45.32	0.079	0.996	99.07	41.78	0.084
	2			0.995	98.48	92.32	0.050	0.991	98.08	59.96	0.065
	3			0.99	97.77	111.84	0.045	0.981	96.32	82.89	0.116
	4			0.995	98.98	75.52	0.049	0.959	91.32	127.35	0.153
	5			0.983	96.68	136.55	0.059	0.921	78.94	198.36	0.167
	6			0.994	96.68	186.14	0.074	0.827	40.91	332.29	0.176

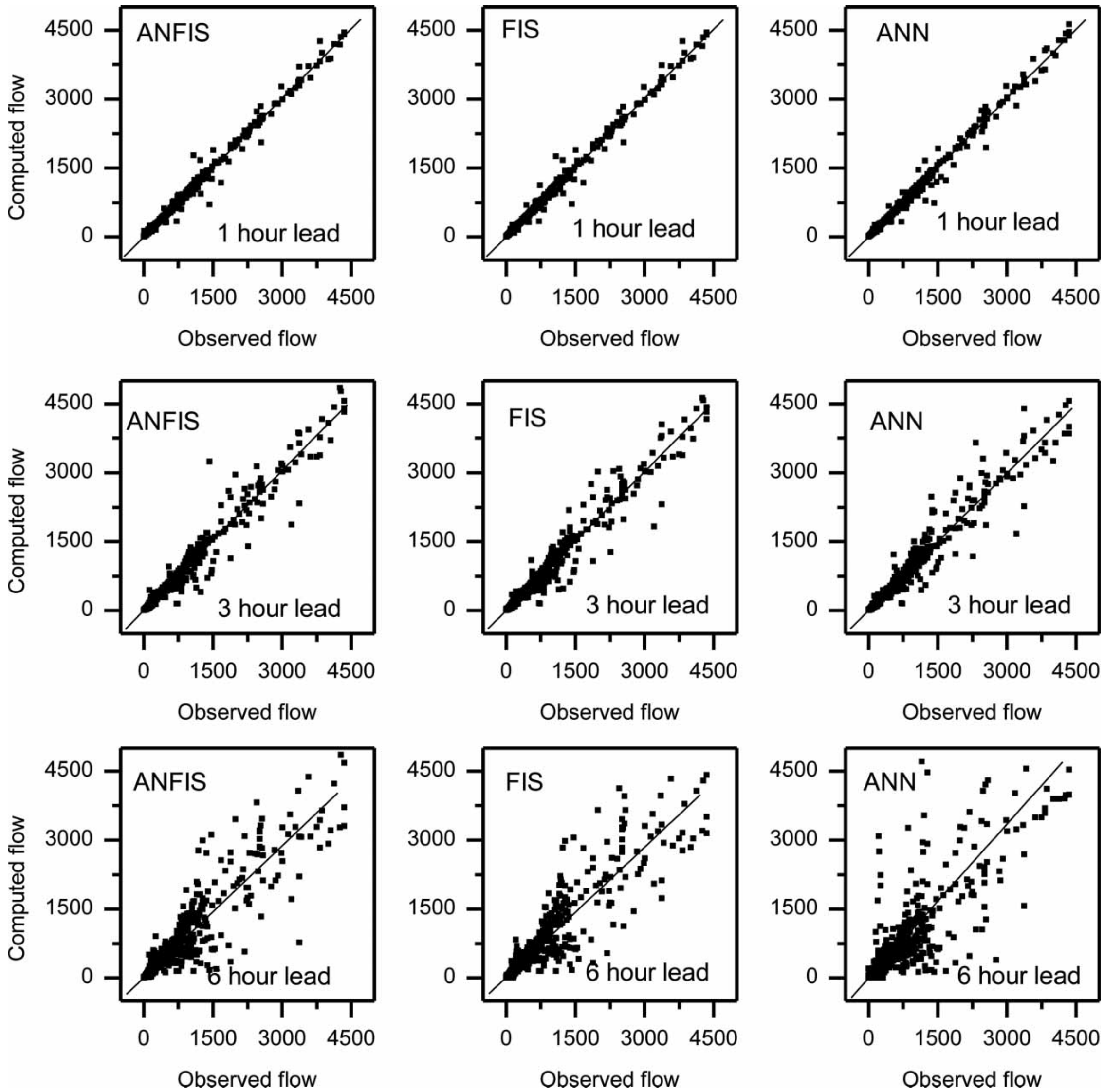

Figure 3 Scatter plots for 1-, 3- and 6-hour lead forecast for ANFIS, FIS and ANN models during validation period.

is interesting to note that for a 1-hour ahead forecast, the performance of all the models is equal, but when the lead time increases, the ANFIS performance is superior compared to the other two models. It may be noted that ANFIS estimated MARE as 0.67 whereas ANN and FIS estimated 0.074 and 0.87, respectively.

Scatter plots between observed and computed flows for 1, 3 and 6 hour lead forecasting are presented in Figure 3. From Figure 3, it is observed that ANFIS and FIS show a close match between observed and predicted values for a 1 hour lead forecasting. As the lead time increases, the model performance decreases, but ANFIS estimates the full range of flow very accurately in different forecast horizons and most of the flow tend to fall close to the 45° line compared to other two models, thus showing a good agreement between observed and forecasted flows. ANN predicted values deviate significantly from the observed values. Large numbers of points are away from the linear fit line which shows a poor match between observed and predicted values.

The distribution of relative errors by all models during the validation period for a one hour lead forecasting is presented in Figure 4 from which it is evident that the ANFIS performs better than the other two models.

6.4 Performance of models in peak prediction

The global error statistics provide relevant information on the overall performance of the models, but do not provide specific information about model performance at high flow, which is of critical importance in a flood forecasting context. Hence, the three peak events are randomly selected from the validation data sets for further analysis. The percentage error in the predicted peak flow and the time gap between the predicted and observed peak flows are calculated.

The percentage error in peak for the three most severe flood events predicted by the three models is presented in Table 4. Table 4 reveals that for each scenario, all the three models overestimated the peak flow. The highest peak that occurred within

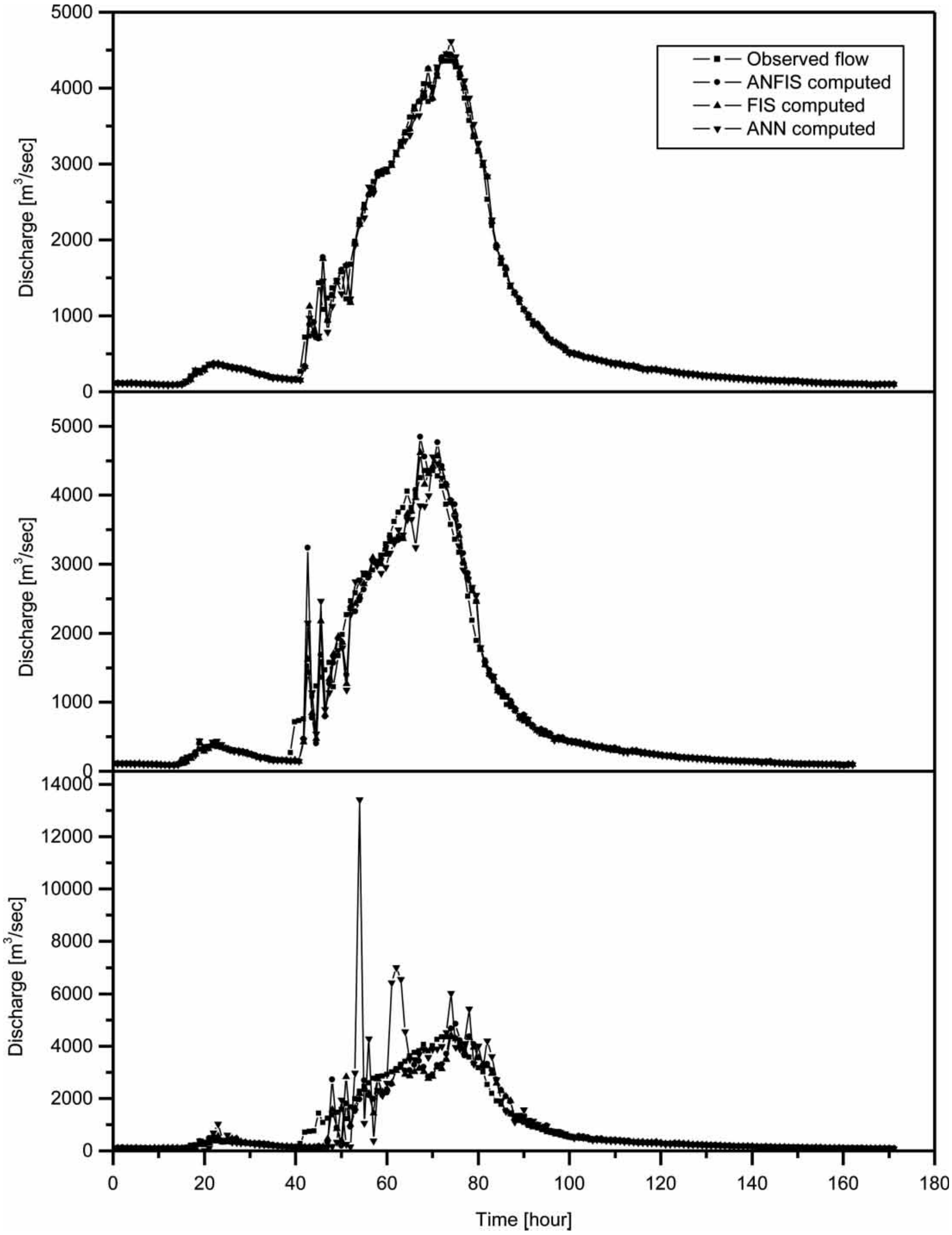

Figure 4 Observed and computed plot for peak flow event during 1 hour lead forecasting.

Table 4 Percentage error in peak for three most severe peak flood (validation year 1992–1993).

Forecast hour	Models	4354.5 (m^3/s)	3379.3 (m^3/s)	2569.39 (m^3/s)
1-hour	FIS	2.02	10.36	1.68
	ANFIS	1.14	9.42	2.37
	ANN	14.43	3.55	9.36
3-hour	FIS	6.1	19.87	18.1
	ANFIS	11.3	14.01	7.89
	ANN	5.16	26.32	12.10
6-hour	FIS	3.83	21.82	52.84
	ANFIS	11.34	12.86	34.48
	ANN	89.23	94.34	72.67

the validation period, i.e. 4354.5 m^3/s was estimated with less than 3% error by ANFIS and FIS at 1 hour lead time. For a 6 hour lead time, FIS and ANFIS predicted with an error less than 5−12%, respectively, but ANN performance in both lead hours is inferior to ANFIS and FIS. It can be observed that ANN can predict higher peak flow values with less error for lower forecast lead hours.

Figure 5 shows the forecasted residuals for a full range of flow at a 1 hour lead time for the three models. It is observed that the major errors for all the models are near the peak flow region of the hydrograph. It is found that the accuracy decreases for the ANN and FIS models; however, ANFIS estimated the peak flow characteristics better than other two models as the magnitude of error is very less compared to ANN and FIS models.

For the comparison of performance among the three models, the distribution of errors for different ranges of flow is calculated. The total validation period of flow is divided into low, medium and high magnitude flows. The partition of data sets is carried out as (a) low flow ($x < \mu$), (b) medium flow ($\mu \leq x \leq \mu + 2\sigma$) and c) high flow ($x > \mu + 2\sigma$), where $x =$ the number of data points, $\mu =$ average of the observed discharge and $\sigma =$ standard deviation of the observed discharge.

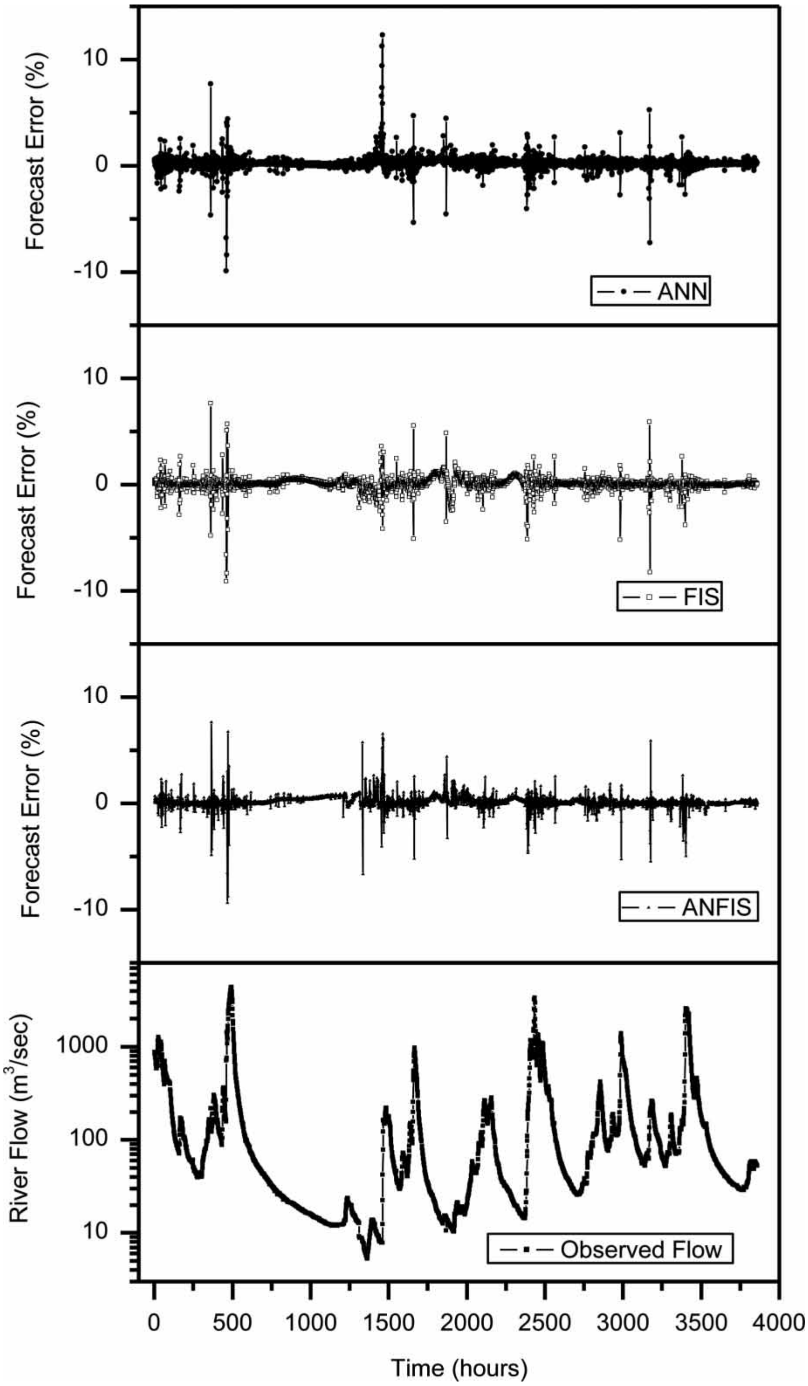

Figure 5 Forecast error plot for 1 hour lead forecast during validation period.

Table 5 Threshold statistics for different models during validation period (1992–1993).

	1 hour lead			3 hour lead			6 hour lead		
	FIS	ANFIS	ANN	FIS	ANFIS	ANN	FIS	ANFIS	ANN
Low flow									
TS1	38.87	42.54	16.42	14.41	21.18	14.75	11.62	9.36	8.76
TS5	84.77	84.56	79.86	49.18	75.21	69.44	53.48	50.87	37.13
TS10	95.88	93.54	81.22	67.88	89.67	88.94	78.11	78.55	65.28
TS20	99.39	96.64	96.77	92.67	97.34	96.99	93.55	92.38	79.60
Medium flow									
TS1	28.84	27.68	18.21	8.83	9.54	8.53	4.92	5.91	3.22
TS5	83.47	82.89	67.28	43.29	37.99	48.81	25.76	23.99	20.51
TS10	94.94	95.13	89.04	70.63	66.27	73.38	49.66	48.82	38.13
TS20	97.98	97.82	97.3	89.47	89.27	89.59	74.58	72.47	66.10
High flow									
TS1	27.94	27.41	7.35	10.14	8.03	6.85	2.96	4.17	4.12
TS5	75.74	76.3	43.38	32.61	30.66	30.14	17.04	13.04	14.43
TS10	88.97	90.37	76.7	57.97	54.74	47.26	26.67	26.81	28.86
TS20	96.32	97.04	96.9	83.33	81.75	78.77	55.56	60.87	41.24

Table 6 Statistical indices for 1 hour lead forecasting without rainfall.

	Training			Testing		
Performance indices	FIS	ANFIS	ANN	FIS	ANFIS	ANN
CORR	0.9986	0.9988	0.9989	0.9974	0.9978	0.9972
EFF (%)	99.71	99.75	99.73	99.47	99.54	99.40
RMSE (m^3/s)	40.36	36.98	38.77	32.33	30.02	34.26

Number of data points in low, medium and high flow is 3207, 596 and 147, respectively. The threshold statistic for a level of x% is a measure of the consistency in forecasting errors from a particular model. The threshold statistics 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 expressed in x% level (TL) as, $TS_x = (Y_x/n)^* 100$, where Y_x is the number of computed stream flows (out of n total computed) for which the absolute relative error is less than x% for the model. Threshold statistics for different ranges of flow, computed from all the three models are presented in Table 5 for comparison.

6.4.1 Low flow

It can be observed from Table 5 that the three models show comparable performance in the low flow range. At a 1 hour lead forecast, ANFIS and FIS models show more than 80% of low flow are within a 5% relative error. However, the distribution of error is seen to be better for the ANFIS model in the low flow range as 42.54% of the low flow counts are predicted within a 1% relative error compared to 38.87% for FIS. At a 6 hour lead forecasting, it is observed that FIS counts 11.62% of low flow within a 1% relative error. During a 3 hour forecast, the ANN seems to be better than FIS as 69.44% of the low flow counts are within a 5% relative error.

6.4.2 Medium flow

In case of medium flow, FIS and ANFIS models are found to be more efficient than ANN. For a 1 hour lead forecasting within a 1% error, FIS counts 28.84% of medium flow whereas ANFIS and ANN estimated 27.68 and 18.21%, respectively. For a higher lead hour (6 hour), the FIS estimated medium flow is 25.76% within a 5% relative error whereas ANFIS estimated 23.99%. At a 3 hour lead forecasting, ANN counts 48.81% of medium flow within a 5% relative error, which seems to be better as compared to ANFIS and FIS.

6.4.3 High flow

In case of high flow, again FIS and ANFIS are observed to be better than ANN. At a 1 hour lead forecasting, ANFIS and FIS counts 27.41% and 27.94% of flow, respectively, within a 1% relative error, whereas ANN measures only 7.35% of flow. At a 6 hour forecasting, it is difficult to predict which model is better as ANFIS counts 4.17% of high flow within a 1% relative error and FIS and ANN count 17.04% within a 5% relative error and 28.86% within a 10% relative error, respectively. From this analysis, it is difficult to suggest that any specific model is superior to the other.

A sensitivity analysis was carried out by removing the rainfall values from the input vector, and the model parameters are re-estimated. This analysis is performed with the heuristic that as the upstream runoff values are incorporated into the input vector, the prediction may not be sensitive to rainfall information. Model parameters are again estimated for different lead times as described earlier. The values of the performance indices for the 1 hour lead forecast of all three models are presented in Table 6.

It is observed from the table that correlation statistics is consistent for all models during the calibration as well as the validation period. Correlation coefficient is more than 0.99 by three models. The model efficiency is more than 99% during the calibration and validation periods for all models. ANFIS predicts 30.02 m^3/s RMSE and FIS predicts 32.33 m^3/s during the validation. Overall, for a 1-hour lead forecast, the performance of all the models is comparable. It is observed that RMSE estimated by all the models is slightly less compared to earlier developed models where rainfall information was incorporated into the models (see Table 1). It is worth mentioning that the RMSE estimated by the ANFIS model is very less and the analysis indicates that the downstream prediction is not sensitive to rainfall if upstream discharge values are considered in the input. The consequent parameters are presented in Table 7 and it is evident from the table that weight associated with the rainfall is zero. When the rainfall is removed from the input vector, the model complexity is reduced (in terms of parameters) and the model performance also improves.

7 Summary and conclusions

This paper addresses the problem of generating if–then rules from hierarchical fuzzy models where the number of influential variables is high. In the current study, there are 19 input variables, hence grid-partitioning techniques cannot be applied to generate a parsimonious rule set. In the current investigation, a SC algorithm was used for input space partitioning in fuzzy and neurofuzzy models, and the developed models were

Table 7 Values of consequent parameters for ANFIS model for 1 hour lead.

Rainfall			Manot			Mohegaon			H. Nagar	Dindori			Mandla						Const.
0	0	0	0.30	0.30	0.30	0.32	0.32	0.32	0.32	0.09	0.09	0.09	0.21	0.21	0.21	0.21	0.21	0.21	0.21
0	0	0	0.46	0.45	0.45	0.54	0.53	0.53	0.51	0.20	0.20	0.20	0.42	0.41	0.41	0.41	0.41	0.41	0.41
0	0	0	0.40	0.40	0.40	0.43	0.43	0.43	0.40	0.16	0.16	0.16	0.31	0.31	0.31	0.31	0.31	0.30	0.31
0	0	0	0.51	0.51	0.51	0.59	0.59	0.58	0.55	0.28	0.27	0.27	0.48	0.48	0.47	0.47	0.47	0.47	0.46
0	0	0	0.35	0.35	0.35	0.38	0.38	0.38	0.34	0.13	0.13	0.13	0.26	0.26	0.26	0.26	0.26	0.26	0.26
0	0	0	0.38	0.38	0.38	0.47	0.48	0.48	0.51	0.15	0.15	0.15	0.37	0.37	0.37	0.37	0.37	0.36	0.36
0	0	0	0.59	0.58	0.58	0.64	0.64	0.64	0.58	0.31	0.31	0.31	0.54	0.53	0.53	0.53	0.53	0.53	0.53

applied for river flow forecasting using data for the Narmada River basin in India. An ANN model which uses an LM backpropagation training algorithm has been developed for the same basin. Five years monsoon season rainfall and discharge data (1989–1993) were used for the analysis. The relative performance of these models was comprehensively evaluated using various statistical indices. It is observed that the neurofuzzy model performs better compared to the ANN and FIS models in estimating the hydrograph characteristics, especially at longer forecast time horizons. It is found that the use of a less number of if–then rules in the neurofuzzy model can accurately model the rainfall–runoff dynamics compared to FIS model which uses the same partitioning algorithm for the rule generation. The ANFIS model is able to capture the inherent nonlinearity in the rainfall runoff process better than the other two models, and is able to forecast flows satisfactorily up to 6 hours in advance. A very close fit is obtained between computed and observed flows up to 1 hour in advance for all models, but the ANFIS and FIS models tend to preserve this performance at longer lead times. The neurofuzzy models were able to predict hydrograph characteristics such as peak flow and time to peak better than the ANN and fuzzy models. It was found that the highest peak flow (4354.5 m^3/s) was estimated with less then 3% error by the ANFIS and FIS models at 1 hour lead time, but the ANN predicted 14.43% error in the peak flow. At a 6 hour lead time, the FIS and ANFIS predicted flows with an error less than 5–12%, respectively, but the ANN performance was not comparable. It was also found that ANFIS outperformed the other models in estimating MARE indices at longer lead time. This study also suggests that simpler models with only runoff values in the input vector for continuous river-flow simulation can perform better than their complex counterparts that use a number of influencing variables as input. A sensitivity analysis was carried out by removing the rainfall from input vector and it was observed that there were no large differences in the model performance.

References

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a. Artificial neural networks in hydrology I: Preliminary concepts. *Journal of Hydrologic Engineering, ASCE*, 5 (2), 115–123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000b. Artificial neural networks in hydrology II: Hydrologic applications. *Journal of Hydrologic Engineering, ASCE*, 5 (2), 124–137.

Atiya, A.F., El-Shoura, S.M., Shaheen, S.I., and El-Sherif, M.S., 1999. A comparison between neural-network forecasting techniques. Case study: river flow forecasting. *IEEE Transactions on Neural Networks*, 10 (2), 402–409.

Babovic V. and Keijzer M., 1999. Forecasting river discharges in the presence of chaos and noise. *In*: J. Marsalek, W.E. Watt, E.

Zeman and F. Sieker, eds. *Flood issues in contemporary water management*. NATO Science Series. Dordrecht: Kluwer Academic Press, 405–419.

Chang, C.L., Lo, S.L., and Yu, S.L., 2005. Applying fuzzy theory and genetic algorithm to interpolate precipitation. *Journal of Hydrology*, 314, 92–104.

Chang, F., Chen, Y., and Hu, H.-F, 2001. Counterpropagation fuzzy–neural network for streamflow reconstruction. *Hydrological Processes*, 15, 219–232.

Chiu, S., 1994. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2 (3), 267–278.

Connor, J.T., Martin, R.D., and Atlas, L.E., 1994. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5 (2), 240–254.

Dawson, D.W. and Wilby, R., 2001. Hydrological modeling using artificial neural networks. *Progress in Physical Geography*, 25 (1), 80–108.

Hornik, K., Stinchcombe, M., and White, H., 1989. Multilayer feed forward networks are universal approximators. *Neural Networks*, 2, 359–366.

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

Lin, C.J. and Chen, C.H., 2006. A compensation- based recurrent fuzzy neural network for dynamic system identification. *European Journal of Operational Research*, 172 (2), 696–715.

Ludermir, T. and Valenga, M., 1998. Self organizing modeling in forecasting daily river flows. *In*: *Proceedings of the 5th Brazilian Symposium on Neural Networks*, 9–11 December 1998 Belo Horizonte, Brazil. Los Alamitos, Calif: IEEE Computer Society Press, 210–214.

Nayak, P.C., 2010. Explaining internal behavior in a fuzzy if–then rule-based flood-forecasting model. *Journal of Hydrologic Engineering*, 15 (1), 20–28, doi: 10.1061/(ASCE)HE.1943-5584.0000146.

Nayak, P.C. and Sudheer, K.P., 2008. Fuzzy model identification based on cluster estimation for reservoir inflow forecasting. *Hydrological Processes*, 22, 827–841, doi:10.1002/hyp.6644.

Nayak, P.C., Sudheer, K.P., and Ramasastri, K.S., 2005a. Fuzzy computing based rainfall–runoff model for real time flood forecasting. *Hydrological Processes*, 19, 955–968.

Nayak, P.C., *et al.*, 2004. A neuro-fuzzy computing technique for modeling hydrological time series. *Journal of Hydrology*, 291 (1–2), 52–66.

Nayak, P.C., *et al.*, 2005b. Short-term flood forecasting with a neurofuzzy model. *Water Resources Research*, 41 (4), W04004, doi:10.1029/2004WR003562.

NCIWRD, 1999. *Integrated water resource development: a plan for action*. Report of the National Commission for Integrated Water Resource Development (NCIWRD), Vol. 1. Ministry of Water Resources, Government of India.

Ozelkan, E.C. and Duckstein, L., 2001. Fuzzy conceptual rainfall–runoff models. *Journal of Hydrology*, 253, 41–68.

See, L. and Openshaw, S., 1999. Applying soft computing approaches to river level forecasting. *Hydrological Sciences Journal*, 44 (5), 763–779.

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

Sen, Z. and Altunkaynak, A., 2003. Fuzzy awakening in rainfall–runoff modeling. *Nordic Hydrology*, 35 (1), 31–43.

Setnes, M., 2000. Supervised fuzzy clustering for rule extraction. *IEEE Transactions on Fuzzy Systems*, 8, 416–424.

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

Sudheer, K.P., Nayak, P.C., and Ramasastri, K.S., 2003. Improving peak flow estimates in artificial neural network river flow models. *Hydrological Processes*, 17 (1), 671–686.

Takagi, T. and 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.

Tayfur, G., 2006. Fuzzy, ANN, and regression models to predict longitudinal dispersion coefficient in natural streams. *Nordic Hydrology*, 37 (2), 143–164.

Tayfur, G. and Singh, V.P., 2006. ANN and fuzzy logic models for simulating event-based rainfall–runoff. *Journal of Hydraulic Engineering*, 132 (12), 1321–1330.

Tayfur, G. and Singh, V.P., 2011. Predicting mean and bankfull discharge from channel cross-sectional area by expert and regression methods. *Water Resources Management*, 25 (5), 1253–1267.

Tayfur, G., Moramarco, T., and Singh, V.P., 2007. Predicting and forecasting flow discharge at sites receiving significant lateral inflow. *Hydrological Processes*, 21, 1848–1859.

Ulke, A., Tayfur, G., and Ozkul, S., 2009. Predicting suspended sediment loads and missing data for gediz river, Turkey. *Journal of Hydrologic Engineering*, 14 (9), 954–965.

Vernieuwe, H., *et al.*, 2005. Comparison of data-driven Takagi–Sugeno models of rainfall–discharge dynamics. *Journal of Hydrology*, 302, 173–186.

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

Xiong, L.H. and O'Connor, K.M., 2002. Comparison of four updating models for real-time river flow forecasting. *Hydrological Sciences Journal*, 47 (4), 621–639.

Yager, R. and Filev, D., 1994. Generation of fuzzy rules by Mountain clustering. *Journal of Intelligent & Fuzzy Systems*, 2 (3), 209–219.