Check for updates

River flow forecasting and estimation using different artificial neural network techniques

Özgür Kişi

ABSTRACT

This paper demonstrates the application of different artificial neural network (ANN) techniques for the estimation of monthly streamflows. In the first part of the study, three different ANN techniques, namely, feed forward neural networks (FFNN), generalized regression neural networks (GRNN) and radial basis ANN (RBF) are used in one-month ahead streamflow forecasting and the results are evaluated. Monthly flow data from two stations, Gerdelli Station on Canakdere River and Isakoy Station on Goksudere River, in the Eastern Black Sea region of Turkey are used in the study. Based on the results, the GRNN was found to be better than the other ANN techniques in monthly flow forecasting. The effect of periodicity on the model's forecasting performance was also investigated. In the second part of the study, the performance of the ANN techniques was tested for river flow estimation using data from the nearby river.

Key words | estimation, forecasting, neural computing techniques, river flow

Özgür Kişi (corresponding author)
Erciyes University, Engineering Faculty,
Civil Eng. Dept., 38039, Kayseri,
Turkey
E-mail: *kisi@erciyes.edu.tr*

INTRODUCTION

Many of the activities associated with planning and operation of the components of a water resource system require forecasts of future events. For the hydrologic component, there is a need for both short term and long term forecasts of streamflow events in order to optimize the system or to plan for future expansion or reduction.

Traditionally, the class of autoregressive moving average (ARMA) models has been the statistical method most widely used for modeling water resources time-series (Maier & Dandy 1996). In streamflow forecasting, time-series models are used to describe the stochastic structure of the time sequence of streamflows and precipitation values measured over time. The limitation of univariate time-series methods in streamflow forecasting is that the only information they incorporate is that which is present in past flows. Many of the available techniques for time series analysis assume linear relationships among variables. In the real world, however, temporal variations in data do not exhibit simple regularities and are difficult to analyze and predict. It seems necessary that nonlinear models such as ANN, which are suited to complex nonlinear problems, be used for the analysis of real world temporal data (Kisi 2005).

ANNs have been successfully applied in a number of diverse fields including water resources. In order to optimally fit an ARMA-type model to a time-series, the data must be stationary and follow a normal distribution (Hipel 1986). When developing ANN models, the statistical distribution of the data need not be known and non-stationarities in the data, such as trends and seasonal variations, are implicitly accounted for by the internal structure of the ANNs (Maier & Dandy 1996). ANNs differ from the traditional approaches in synthetic hydrology in the sense that they belong to a class of data-driven approaches. Data-driven approaches are suited to complex problems. They have the ability to determine which model inputs are critical, so that there is no need for prior knowledge about the relationships amongst the variables being modeled. ANNs are relatively insensitive to noisy data, unlike ARMA-type models, as they have the ability to determine the underlying relationship between model inputs and outputs, resulting in good generalization capability.

doi: 10.2166/nh.2008.026

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

In the hydrological forecasting context, recent experiments have reported that artificial neural networks (ANN) may offer a promising alternative for rainfall–runoff modeling, streamflow prediction and reservoir inflow forecasting. Smith & Eli (1995) used a feed forward ANN (FFNN) to predict the peak discharge and the time of peak resulting from a single rainfall pattern. Shamseldin (1997) compared the FFNN models with the simple linear model, the seasonally based linear perturbation model and the nearest neighbour linear perturbation model in rainfall-runoff modelling. He suggested that the neural network shows considerable promise in the context of rainfall-runoff modelling. Tokar & Johnson (1999) employed a FFNN methodology to forecast daily runoff as a function of daily precipitation, temperature, and snowmelt for the Little Patuxent River watershed in Maryland. They compared FFNN rainfall-runoff model with results obtained using the statistical regression and a simple conceptual model and found that the FFNN performs better than the others. Sudheer *et al.* (2002*a*) proposed a data-driven algorithm for designing the radial basis function ANN (RBF) structure in rainfall-runoff modelling. They validated their methodology using the data for a river basin in India.

Karunanithi *et al.* (1994) used a FFNN model to predict river flows. They compared the accuracy of FFNN with that of the analytic nonlinear power model and found that the ANN model shows better accuracy than the latter. Zealand *et al.* (1999) investigated the utility of FFNN model for short term streamflow forecasting. They explored the capabilities of FFNN and compared the performance of this tool to conventional approaches used to forecast streamflow. They concluded that the ANNs developed consistently outperformed a conventional model during the verification (testing) phase. Chang & Chen (2001) used a fuzzy-neural network model for real time streamflow prediction. They compared the model results with those of the autoregressive moving average exogenous variables model (ARMAX) and found that the fuzzy-neural network model could offer a higher degree of reliability and accuracy than the ARMAX in streamflow forecasting. Sivakumar *et al.* (2002) used FFNN models for 1-day and 7-day-ahead forecasts of daily river flows. Sudheer & Jain (2003) compared the FFNN and RBF in daily streamflow estimation. They found that the FFNN results were similar to those of the RBF. Cigizoglu (2003) investigated the accuracy of FFNN in estimation, forecasting and extrapolation of daily streamflow data. Cigizoglu & Kisi (2005) developed FFNN models for 1-day ahead forecasting of daily streamflows. They used k-fold partitioning, a statistical method, in the ANN training stage. Kisi (2004) used FFNN models in monthly streamflow forecasting. He compared the ANN results with the autoregressive models (AR) and found that the ANN model performed better than the AR. Sahoo & Ray (2006) investigated the accuracy of FFNN and RBF models in forecasting of daily flows of a Hawaii stream. To the knowledge of the author, there is no published work which compares the input–output mapping capability of feed forward neural networks, radial basis function ANN and generalized regression neural networks (GRNN) in prediction of monthly streamflows.

There are many parameters (precipitation, evapotranspiration, ground water, initial moisture content of soil etc.) which affect the next day runoff. Although it is possible to identify sophisticated models taking into consideration the hydrological and hydro-meteorological variables such as precipitation, runoff, temperature and evaporation, it is economically preferable that a model which simulates the flow variations on the basis of past discharge records be available to the decision maker, whether administrator, local authority or technical operator. Therefore, only the past discharge records were used as inputs in the present study. The present study describes the utilization of the input–output mapping capabilities of different neural network techniques in monthly streamflow prediction. Below, three different ANN techniques, namely, FFNN, GRNN and RBF are used in one-month ahead streamflow forecasting and results are evaluated. The effect of periodicity on the forecasting performance of the models is investigated. The ANN techniques are used to estimate monthly flows of Canakdere River using the data of Goksudere River. The term forecasting is used for the model applications with input and output data belonging to the same river station in the study. The term estimation is preferred when input and output data corresponds to different stations.

FEED FORWARD NEURAL NETWORKS (FFNN)

A FFNN distinguishes itself by the presence of one or more hidden layers, whose computation nodes are

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

correspondingly called hidden neurons of hidden units. FFNN is massively parallel system composed of many processing elements connected by links of variable weights. Of the many ANN paradigms, the backpropagation network is by far the most popular (Haykin 1994). The network consists of layers of parallel processing elements, called neurons, with each layer being fully connected to the proceeding layer by interconnection fully connected to the proceeding layer by interconnection strengths, or weights, W. Figure 1 illustrates a three-layer FFNN consisting of layers i, j, and k, with the interconnection weights W_{ij} and W_{jk} between layers of neurons. Initial estimated weight values are progressively corrected during a training process which compares predicted outputs to known outputs, and backpropagates any errors (from right to left in Figure 1) to determine the appropriate weight adjustments necessary to minimize the errors.

Each neuron in layers j and k receives the weighted sum of outputs NET from the previous layer as input. As an example, NET for layer j is given by

$$NET_{pj} = \sum_{i=1}^{L} W_{ij} O_{pi} + \theta_j \tag{1}$$

where θ_j = a bias, or threshold value, for neuron j, that is trainable by treating it as a weight W_j in product with an output O_p set equal to 1. Each neuron in layers j and k produces its output $f(NET)$ by passing its value of NET through a nonlinear activation function. A commonly used functional form is the logistic activation function:

$$f(NET) = \frac{1}{1 + e^{-NET}} \tag{2}$$

The connection weights (W_{ij}, W_{jk}) of the network are learned through a process called training in which large numbers of input-output pattern pairs are presented to the network in a repetitive fashion designed to provide iterative corrections to the weights. Each iteration (called an "epoch") is a single pass through all training pattern pairs. The process consists of presenting an input pattern p to the network, making predictions as to the output O_{pk}, and then comparing this predicted output to the input pattern's actual output Tpk. The total error E_p, based on the squared difference between predicted and actual outputs for pattern p, is computed as

$$E_p = \sum_{k=1}^{N} \left(T_{pk} - O_{pk}\right)^2 \tag{3}$$

The goal of the training process is to present a sufficient number P of unique input-output pattern pairs, which when coupled with a suitable methodology for iterative correction of the interconnection weights, produces a final set of weights that minimizes the global error, E, defined as

$$E = \sum_{p=1}^{p} E_p \tag{4}$$

The Levenberg–Marquardt (LM) optimization technique is used here for adjusting the weights. This optimization technique is more powerful than the conventional gradient descent technique (Hagan & Menhaj 1994; El-Bakyr 2003; Cigizoglu & Kisi 2005). Throughout all the FFNN simulations an adaptive learning rate was used to increase the ratio of convergence. A difficult task with FFNN involves choosing the number of hidden nodes. Here, we use the FFNN with one hidden layer and the common trial and error method to select the number of hidden nodes. The sigmoid function is used for the hidden and output node's activation functions. The ANN networks training were stopped after 100 epochs since the variation of error was too small after this epoch. The error graph for an ANN model during training is shown in Figure 2.

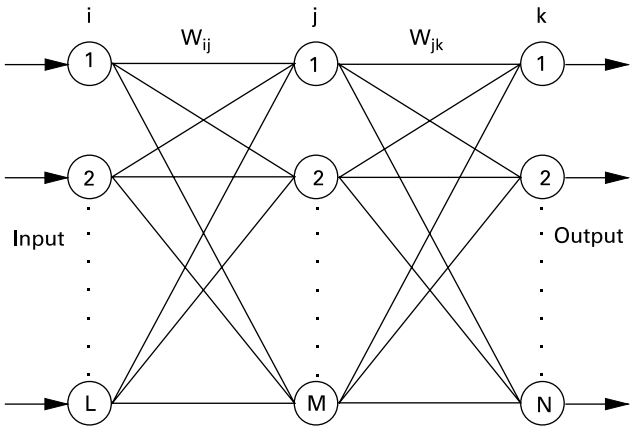

Figure 1 | Schematic diagram of a FFNN.

GENERALIZED REGRESSION NEURAL NETWORKS (GRNN)

A schematic of the GRNN is shown in Figure 3. Tsoukalas and Uhrig (1997) describe the theory of the GRNN. The

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

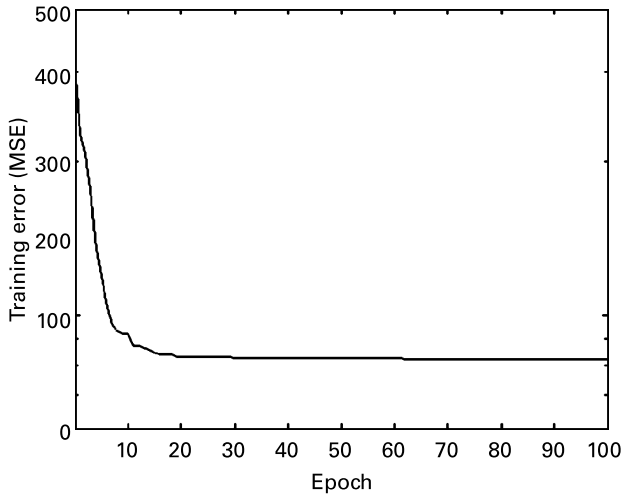

Figure 2 | The training error graph for the FFNN model.

GRNN consists of four layers: input layer, pattern layer, summation layer, and output layer. The number of input units in the first layer is equal to the total number of parameters, here this includes from one to three previous monthly flows. The first layer is fully connected to the second, pattern layer, where each unit represents a training pattern and its output is a measure of the distance of the input from the stored patterns. Each pattern layer unit is connected to the two neurons in the summation layer: S-summation neuron and D-summation neuron. The S-summation neuron computes the sum of the weighted outputs of the pattern layer while the D-summation neuron calculates the unweighted outputs of the pattern neurons. The connection weight between the i^{th} neuron in the pattern layer and the S-summation neuron is y_i; the target output value corresponding to the i^{th} input pattern. For D-summation neuron, the connection weight is unity. The output layer merely divides the output of each S-summation neuron by that of each D-summation neuron, yielding the predicted value to an unknown input vector x as

$$\widehat{y}_i(x) = \frac{\sum_{i=1}^{n} y_i \exp[-D(x, x_i)]}{\sum_{i=1}^{n} \exp[-D(x, x_i)]} \tag{5}$$

where n indicates the number of training patterns and the Gaussian D function in Equation (5) is defined as

$$D(x, x_i) = \sum_{j=1}^{p} \left(\frac{x_j - x_{ij}}{\zeta} \right)^2 \tag{6}$$

where p indicates the number of elements of an input vector. The x_j and x_{ij} represent the j^{th} element of x and x_i; respectively. The ζ is generally referred to as the spread factor, whose optimal value is often determined experimentally (Kim *et al.* 2003). A large spread corresponds to a smooth approximation function. Too large a spread means a lot of neurons will be required to fit a fast changing function. Too small a spread means many neurons will be required to fit a smooth function, and the network may not generalize well. In this study, different spreads were tried to find the best value for the given problem. The GRNN does

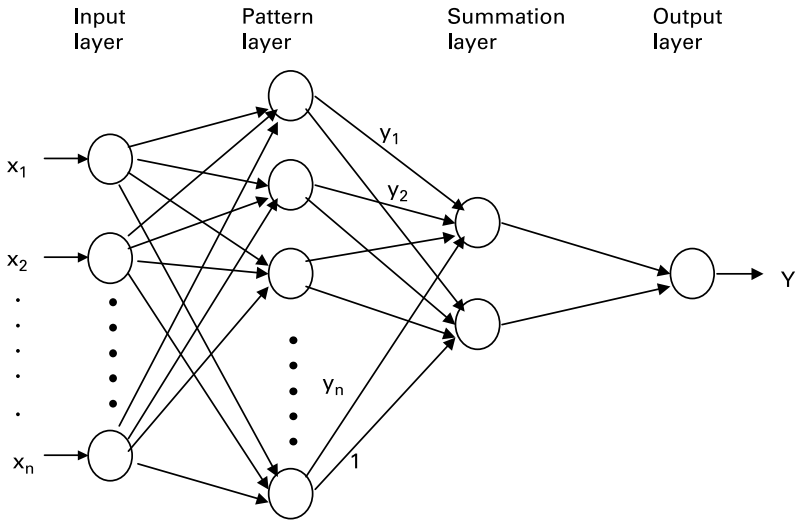

Figure 3 | Schematic diagram of a GRNN.

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

not require an iterative training procedure as in the backpropagation method (Specht 1991).

RADIAL BASIS NEURAL NETWORKS (RBF)

RBF networks were introduced into the neural network literature by Broomhead & Lowe (1988). The RBF network model is motivated by the locally tuned response observed in biological neurons. Neurons with a locally tuned response characteristic can be found in several parts of the nervous system, for example, cells in the visual cortex sensitive to bars oriented in a certain direction or other visual features within a small region of the visual field (Poggio & Girosi 1990). These locally tuned neurons show response characteristics bounded to a small range of the input space. The theoretical basis of the RBF approach lies in the field of interpolation of multivariate functions. The objective of interpolating a set of tuples $(x^s, y^s)_{s=1}^N$ with $x^s \in R^d$ is to find a function F: $R^d \rightarrow R$ with $F(x^s) = y^s$ for all $s = 1, \ldots, N$ where F is a function of a linear space. In the RBF approach the interpolating function F is a linear combination of basis functions

$$F(x) = \sum_{s=1}^{N} w_s \phi(\|x - x^s\|) + p(x) \quad (7)$$

where $\|\cdot\|$ denotes Euclidean norm, $w_1, \ldots, w_N$ are real numbers, ϕ a real valued function, and $p \in \prod_n^d$ a polynomial of degree at most n (fixed in advance) in d variables. The interpolation problem is to determine the real coefficients $w_1, \ldots, w_N$ and the polynomial term $p = \sum_{l=1}^{D} a_1 p_j$ where $p_l, \ldots, p_D$ is the standard basis of $\prod_n^d$ and $a_1, \ldots, a_D$ are real coefficients. The interpolation conditions are

$$F(x^s) = y^s, \quad s = 1, \ldots, N \quad (8)$$

and

$$\sum_{s=1}^{N} w_s p_j(x^s) = 0, \quad j = 1, \ldots, D \quad (9)$$

The function ϕ is called a radial basis function if the interpolation problem has a unique solution for any choice of data points. The most popular and widely used radial basis function is the Gaussian basis function

$$\phi(\|x - c\|) = e^{-(\|x-c\|/2\sigma^2)} \quad (10)$$

with peak at center $c \in R^d$ and deceasing as the distance from the center increases.

The solution of the exact interpolating RBF mapping passes through every data point (x^s, y^s). In the presence of noise, the exact solution of the interpolation problem is typically a function oscillating between the given data points. An additional problem with the exact interpolation procedure is that the number of basis functions is equal to the number of data points and so calculating the inverse of the NxN matrix ϕ becomes intractable in practice. The interpretation of the RBF method as an artificial neural network consists of three layers: a layer of input neurons feeding the feature vectors into the network; a hidden layer of RBF neurons, calculating the outcome of the basis functions; and a layer of output neurons, calculating a linear combination of the basis functions (Taurino *et al.* 2003). Different numbers of hidden layer neurons were tried in the study.

CASE STUDIES

In this study, the monthly flow data which from Gerdelli Station on Canakdere River and Isakoy Station on Goksudere River in the Eastern Black Sea region of Turkey are used. The locations of the Gerdelli and Isakoy Stations are shown in Figure 4. The drainage areas at these sites are 285 km^2 for Gerdelli and 395 km^2 for Isakoy. The observed data is 39 years (468 months) long with an observation period between 1961 and 1999 for both stations. The observed data is for hydrologic years, *i.e.* the first month of the year is October and the last month of the year is September.

Three different program codes, including neural network toolbox (Demuth & Beale 2004), are written in Matlab® language for the FFNN, RBF and GRNN simulations. The ANN models are written in two parts: the first part trains the network and the second part produces an output (a prediction) for a given input pattern. In each of the investigations which follow; two sets of data are required. The first data set is used to train the network, and is referred to as the training set. The second data set is used to determine how well the trained network performed (testing data set). In the applications, the first 29-years of flow data (348 months, 75% of the whole data

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

Figure 4 | The Gerdelli and Isakoy Station on rivers Goksudere and Canakdere.

set) are used for training and the remaining 10-year (120 months, 25% of the whole data set) are used for testing.

The data sets' monthly flow statistics are presented in Table 1 for the Gerdelli and Isakoy stations. The observed monthly flows show high positive skewness ($c_{sx} = 1.31$ and 1.74). The auto-correlations are quite low showing low persistence ($r_1 = 0.53$, $r_2 = 0.33$, $r_3 = 0.01$). The testing data set extremes ($x_{min} = 0.04\,m^3/s$, $x_{max} = 58.1\,m^3/s$) fall within the training data set limits ($x_{min} = 0.00\,m^3/s, x_{max} = 68.4\,m^3/s$) for the Gerdelli Station. This condition is also the same at the Isakoy Station. This means that the trained neural networks do not face difficulties in making an extrapolation.

APPLICATION

Comparison of neural network techniques in 1-month ahead streamflow prediction

Before applying the ANN algorithms, it is necessary to normalize the data. The input and output data are normalized dividing by the maximum value so as to fall in the range 0 to 1. These normalized data are used to train (calibrate) each ANN model.

Three input combinations based on preceding monthly flows are evaluated. The affect of periodicity is investigated by adding a component α into each input combination. α

Table 1 | The monthly statistical parameters of data set for Gerdelli and Isakoy Station (x_{mean}, Sx, Csx, x_{min}, x_{max}, r_1, r_2, r_3 denote the overall mean, standard deviation, skewness, lag-1, lag-2 and lag-3 auto-correlation coefficients, respectively)

Station	Data set	x_{mean} (m^3/s)	Sx (m^3/s)	Csx (m^3/s)	x_{min} (m^3/s)	x_{max} (m^3/s)	r_1	r_2	r_3
Gerdelli	Training data	11.57	13.03	1.35	0.00	64.8	0.573	0.336	0.020
	Test data	12.29	13.20	1.17	0.04	58.1	0.421	0.329	−0.004
	Entire data	11.67	0.60	1.31	0.00	64.8	0.535	0.333	0.011
Isakoy	Training data	14.75	18.0	1.85	0.03	128	0.567	0.340	0.011
	Test data	17.86	19.3	1.45	0.69	88.6	0.386	0.291	0.017
	Entire data	15.44	18.3	1.74	0.03	128	0.520	0.327	0.014

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

takes values between 1 and 12 according to the month of the year to be forecast. Q_t represents the discharge at time t. The input combinations evaluated in the study are; (i) Q_{t-1}, (ii) Q_{t-1} and α, (iii) Q_{t-1}, Q_{t-2}, (iv) Q_{t-1}, Q_{t-2} and α, (v) Q_{t-1}, Q_{t-2}, Q_{t-3}, (vi) Q_{t-1}, Q_{t-2}, Q_{t-3} and α. Here α denotes the periodicity component. In all cases, the output layer has only one neuron, the discharge Q_t for the current month.

For each input combination, three different neural networks are trained. After training is over, the weights are saved and used to test (validate) each network performance on test data. The ANN results are transformed back to the original domain by multiplying by the maximum value, and the mean square errors (MSE) are computed for the test data. The MSE is expressed as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_{observed} - Y_{predicted}\right)^2 \quad (11)$$

where N denotes the number of testing data set.

Multi-linear regression (MLR) is also included in the comparison. For the Gerdelli Station, the MSE statistics of the ANN models in test period are given in Table 2. All the models have the lowest MSE for the input combination iv. The performance of the GRNN seems to be slightly better than those of the FFNN and RBF. The critical issue in training a FFNN is avoiding overfitting as it reduces its capacity of generalization. If too many neurons are used, the network has too many parameters and may overfit the data. In contrast, if too few neurons are included in the network, it might not be possible to fully detect the signal and variance of a complex data set. Here, the hidden nodes number of the FFNN were determined using the trial and error method. As an example, the MSE-hidden nodes number graph for the FFNN model (input combination iv) is shown in Figure 5. The FFNN model with 7 hidden nodes seems to perform better than the others. The number of hidden nodes for the other FFNN models was also found in a same manner. For the Gerdelli Station, optimum hidden nodes numbers of the FFNN models were found to vary between 2 and 9. The best way to avoid overfitting is to use lots of training data. For noise-free data, if we have at least 5 times as many training cases as there are weights in the network, we are unlikely to suffer from overfitting. The other way to avoid the overfitting problem is to use of different FFNN structures (Sudheer *et al.* 2002*b*). In this study, different FFNN structures were tried and 29-years of flow data (348 months) were used for training of the ANN models. The 36 weights were used for the most complex FFNN(3,9,1) model comprising 3 inputs, 9 hidden and 1 output nodes (input combination iv). The training data seem to be enough to avoid overfitting. Different spreads were tried for the GRNN models. The MSE-spread constant graph for the GRNN model (input combination iv) is shown in Figure 6. The optimum spreads were found to vary between 0.05 and 0.16 for this station. Different numbers of hidden layer neurons were tried for the RBF models in the study. Simple trial-error showed that hidden layer neurons' numbers between 8 and 13 gave the minimum mean square errors (MSE) for the Gerdelli Station. As an example, the MSE-hidden nodes number graph for the RBF model

Table 2 | The MSE statistics of different ANN applications for Gerdelli Station (test period)

		MSE (m^6/s^2)			
Model input		**FFNN**	**GRNN**	**RBF**	**MLR**
(i)	Q_{t-1}	133.9	132.3	130.19	170.2
(ii)	Q_{t-1} and α	98.20	94.94	96.76	165.6
(iii)	Q_{t-1} and Q_{t-2}	132.0	126.2	126.1	**155.1**
(iv)	Q_{t-1}, Q_{t-2} and α	**92.00**	**85.59**	**90.59**	155.2
(v)	Q_{t-1}, Q_{t-2} and Q_{t-3}	129.6	124.3	120.5	152.4
(vi)	Q_{t-1}, Q_{t-2}, Q_{t-3} and α	105.9	93.20	99.03	151.4

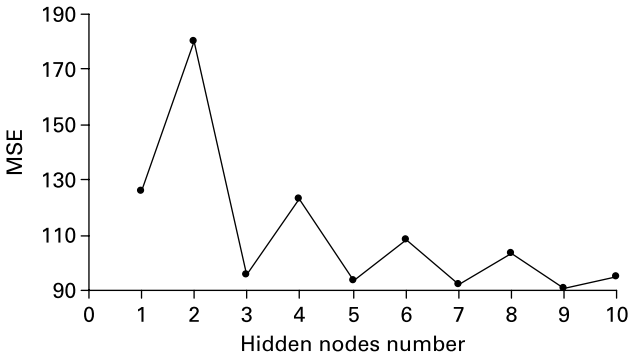

Figure 5 | The MSE-hidden nodes number graph for the FFNN model (input combination iv).

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

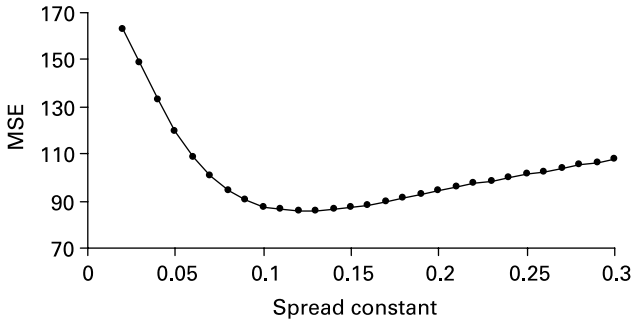

Figure 6 | The MSE-spread constant graph for the GRNN model (input combination iv).

(input combination iv) is shown in Figure 7. The MLR results are not good since the linear relationship between the flow values is low (see auto-correlations in Table 1). The variation of observed and predicted flows is shown in Figure 8. All ANN forecasts are parallel to the observed hydrographs. There are underestimations of some peaks. Availability of a small number of training patterns for peak flows can be a reason behind this. The other reason behind this may be the normalization technique used in this study. However, it is found that FFNN simulations have a drawback. After each FFNN simulation different forecasts were obtained with the same network structure resulting in different performance criteria. This is mainly due to the different initial random weight assignment in the beginning of each training simulation. The FFNN network is trapped by different local error minima each time and the global minimum is not attained. The FFNN forecasting results presented in the study are for the best simulations selected among several others. Accordingly the total duration required for FFNN training is quite long compared with the GRNN and RBF. This total duration of FFNN simulations is longer than the unique GRNN and RBF applications. GRNN algorithm requires the second longest training duration. The other drawback of the FFNN is that it gives negative values for the low flows (see Figure 8). However, the GRNN and RBF forecasts were all positive. Because the forecasts of the GRNN and RBF are bounded by the minimum and maximum of the observed series, preventing the networks from providing forecasts which are not physically possible. The residuals of the models are depicted in Figure 9. The underestimations of the peaks are obviously seen since the models have high residuals for the peak values.

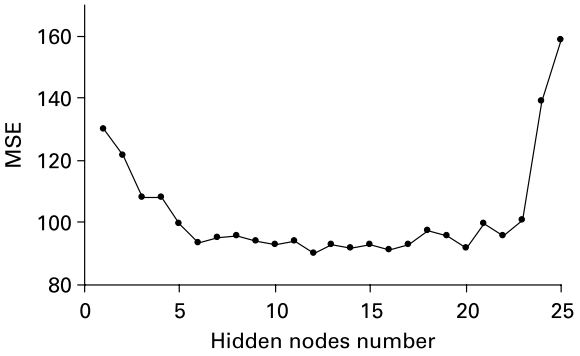

Figure 7 | The MSE-hidden nodes number graph for the RBF model (input combination iv).

The means of the FFNN, GRNN, RBF and MLR flow forecasts are 2.1, 5.3, 3.1 and 17 percent lower than the observed one, respectively. For the water resources applications, where the mean flow is required rather than the extremes, this result can carry significance.

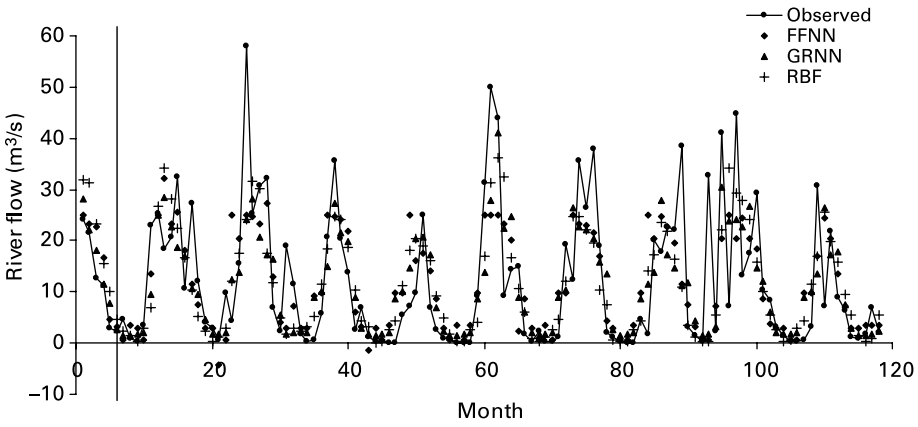

Figure 8 | The observed and forecasted monthly river flows – Gerdelli Station.

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

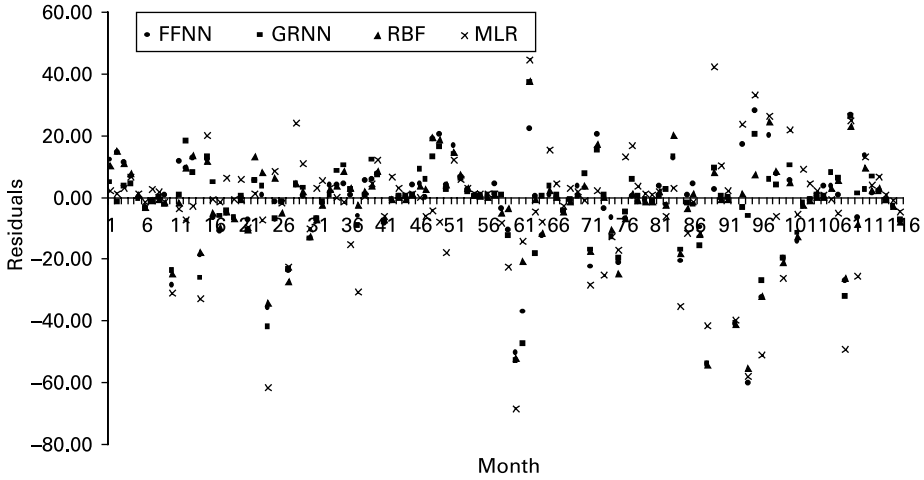

Figure 9 | The residuals of the models in monthly flow forecasting – Gerdelli Station.

The MSE statistics of each model for Isakoy Station are given in Table 3. Here also the lowest MSE (201.5 m^6/s^2) belongs to the GRNN for the input combination (vi). For this station, the RBF has lower MSE than the FFNN as found in the preceding application. For the Gerdelli Station, optimum hidden nodes numbers of the FFNN models were found to vary between 2 and 6. The optimum spreads were found to vary between 0.03 and 0.17 for the GRNN models. The hidden layer neurons' numbers between 8 and 13 gave the minimum mean square errors (MSE) for the RBF models. The forecasting performances of the FFNN, GRNN and RBF are compared in Figure 10 in the form of hydrograph. The underestimations of the peaks are also seen for this station. Figure 11 shows the residuals of models. High residuals can be obviously seen for the peak values as found for the Gerdelli Station. For the Isakoy Station, the means of the FFNN, GRNN, RBF and MLR flow forecasts are 13.2%, 13.0%, 13.0% and 19.5% lower than the observed one, respectively.

Table 3 | The MSE statistics of different ANN applications for Isaköy Station (test period)

		MSE (m^6/s^2)			
Model input		**FFNN**	**GRNN**	**RBF**	**MLR**
(i)	Q_{t-1}	292.2	289.5	285.5	380.0
(ii)	Q_{t-1} and α	**224.3**	213.8	**212.4**	368.2
(iii)	Q_{t-1} and Q_{t-2}	322.4	314.5	309.2	346.6
(iv)	Q_{t-1}, Q_{t-2} and α	236.2	218.8	219.8	**346.3**
(v)	Q_{t-1}, Q_{t-2} and Q_{t-3}	293.9	274.9	241.5	350.3
(vi)	Q_{t-1}, Q_{t-2}, Q_{t-3} and α	244.7	**201.5**	226.9	349.3

The effect of periodicity component on model's forecasting performance

To see the effect of periodicity in monthly flow forecasting, one more input (α) was added into the input combinations (i), (iii) and (v). The GRNN model that uses the periodicity component was called as GRNN-periodic. It has been obviously seen from the Tables 2–3 that the periodicity considerably decreases the MSE for each model. Since the extreme flows, i.e. the minimum and maximum values, are important for certain water resources problems such as dam spillway and sluiceway operations, a detailed analysis of their prediction performance was undertaken. The GRNN

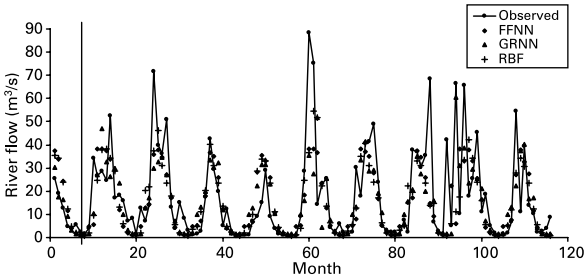

Figure 10 | The observed and forecasted monthly river flows – Isakoy Station.

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

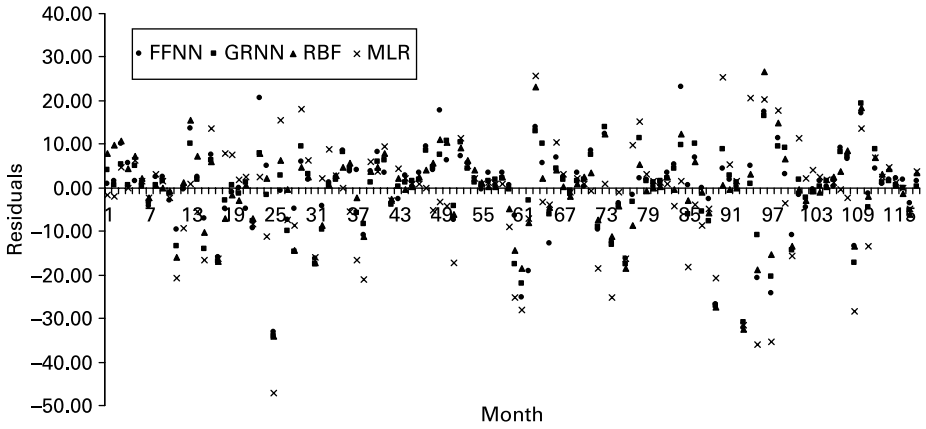

Figure 11 | The residuals of the models in monthly flow forecasting – Isakoy Station.

and GRNN-periodic are compared in peak flow estimation for the Gerdelli (in case of input combinations (iii) and (iv)) and Isakoy Station (in case of input combinations (v) and (vi)) in Tables 4–5. It can be seen from the tables that the peak forecasts of the GRNN-periodic are mostly better than those of the GRNN for both stations. The comparison of model estimates of low flows is presented in Tables 6–7. The GRNN-periodic shows much better performance than the GRNN in low flows prediction. The variation of observed and predicted flows by the GRNN and GRNN-periodic is shown in Figures 12–13. It can be seen from these figures that the GRNN-periodic forecasts closer to the corresponding observed values than those of the GRNN, especially for the peak and low flows.

River flow estimation using nearby river data

The flow data estimation using the nearby river is an important issue since the downstream or upstream data are missing for many rivers. In this case the flow data from the

Table 4 | The comparison of model estimates of hydrograph peaks in test period – Gerdelli Station

Observed peaks	GRNN	GRNN-periodic	Relative error (%)	
(>35 m^3/s)	estimate	estimate	GRNN	GRNN-periodic
58.10	16.91	24.01	− 71	− 59
50.00	21.77	27.97	− 56	− 44
44.70	15.17	24.23	− 65	− 46
44.00	43.62	41.10	− 1	− 7
41.10	8.34	30.31	− 80	− 26
38.40	15.56	11.19	− 59	− 71
38.00	19.56	19.99	− 49	− 47
35.70	22.92	22.69	− 36	− 36
35.70	12.93	20.50	− 63	− 46

Table 5 | The comparison of model estimates of hydrograph peaks in test period – Isakoy Station

Observed peaks	GRNN	GRNN-periodic	Relative error (%)	
(>50 m^3/s)	estimate	estimate	GRNN	GRNN-periodic
88.60	32.03	35.52	− 64	− 60
75.20	26.61	27.47	− 65	− 63
71.70	13.32	29.75	− 81	− 59
68.60	17.00	14.50	− 75	− 79
66.40	28.42	60.48	− 57	− 9
65.70	19.95	38.69	− 70	− 41
54.40	9.60	22.35	− 82	− 59
52.50	16.81	26.46	− 68	− 50
50.80	28.77	27.15	− 43	− 47

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

Table 6 | The comparison of model estimates of hydrograph low flows in test period – Gerdelli Station

Observed flows	GRNN	GRNN-periodic	Relative error (%)	
(<0.15 m^3/s)	estimate	estimate	GRNN	GRNN-periodic
0.14	4.76	2.10	3297	1401
0.14	4.74	8.78	3287	6172
0.14	4.75	1.78	3291	1173
0.13	4.83	1.52	3615	1066
0.10	4.72	1.78	4618	1679
0.10	4.74	2.10	4641	2000
0.04	4.75	2.10	11762	5151

Table 7 | The comparison of model estimates of hydrograph low flows in test period – Isakoy Station

Observed flows	GRNN	GRNN-periodic	Relative error (%)	
(<1.2 m^3/s)	estimate	estimate	GRNN	GRNN-periodic
1.18	7.60	1.19	544	0.6
1.13	6.85	0.65	281	−42
1.03	7.08	9.04	588	778
0.89	3.62	0.80	305	−11
0.76	6.91	0.78	814	3.1
0.73	6.70	1.08	813	48
0.69	6.93	0.65	910	−5.6

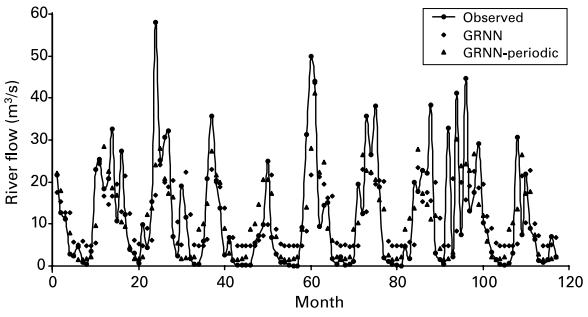

Figure 12 | The observed and forecasted monthly river flows – Gerdelli Station.

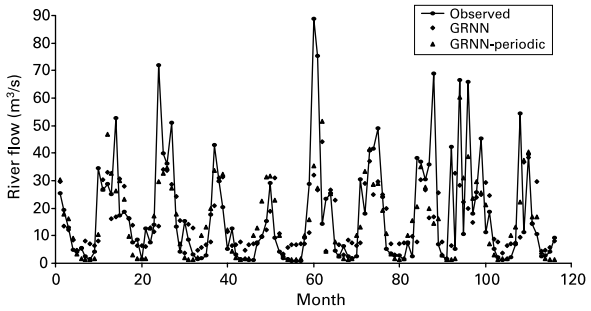

Figure 13 | The observed and forecasted monthly river flows – Isakoy Station.

nearby rivers can be used to estimate the missing flows. To solve this problem, the regression techniques are frequently used (Cigizoglu 2003). Recently, ANNs have been successfully applied in cross-station applications. Cigizoglu (2004) used a FFNN model to estimate daily sediment data of downstream station using the upstream data and vice versa and he compared the FFNN results with those of the MLR. Cigizoglu (2003) investigated the FFNN technique in daily river flow estimation using nearby river data and compared the ANN estimates with those of the MLR. Cigizoglu & Kisi (2005) used a GRNN model to estimate upstream daily intermittent river flows using downstream data. This part of the study focused on the investigation of different ANN techniques' performance in cross-station application. The dependent variables are the flows of the stations Isakoy and Gerdelli having drainage basins with similar hydrological characteristics. The data of the Isakoy Station were used to estimate monthly flows of the Gerdelli Station. In this

Table 8 | The mean square errors (MSE) for different ANN applications in flow estimation of Canakdere River using Goksudere River flows (test period)

		MSE (m^6/s^2)			
Model input		FFNN	GRNN	RBF	MLR
(i)	Q_t	13.60	**15.28**	13.88	**14.54**
(ii)	Q_t and α	14.43	16.85	13.94	14.78
(iii)	Q_t and Q_{t-1}	**13.30**	17.57	**13.20**	14.54
(iv)	Q_t, Q_{t-1} and α	14.22	22.68	13.27	14.84
(v)	Q_t, Q_{t-1} and Q_{t-2}	15.33	16.58	13.98	14.97
(vi)	Q_t, Q_{t-1}, Q_{t-2} and α	14.95	20.25	13.46	15.66

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

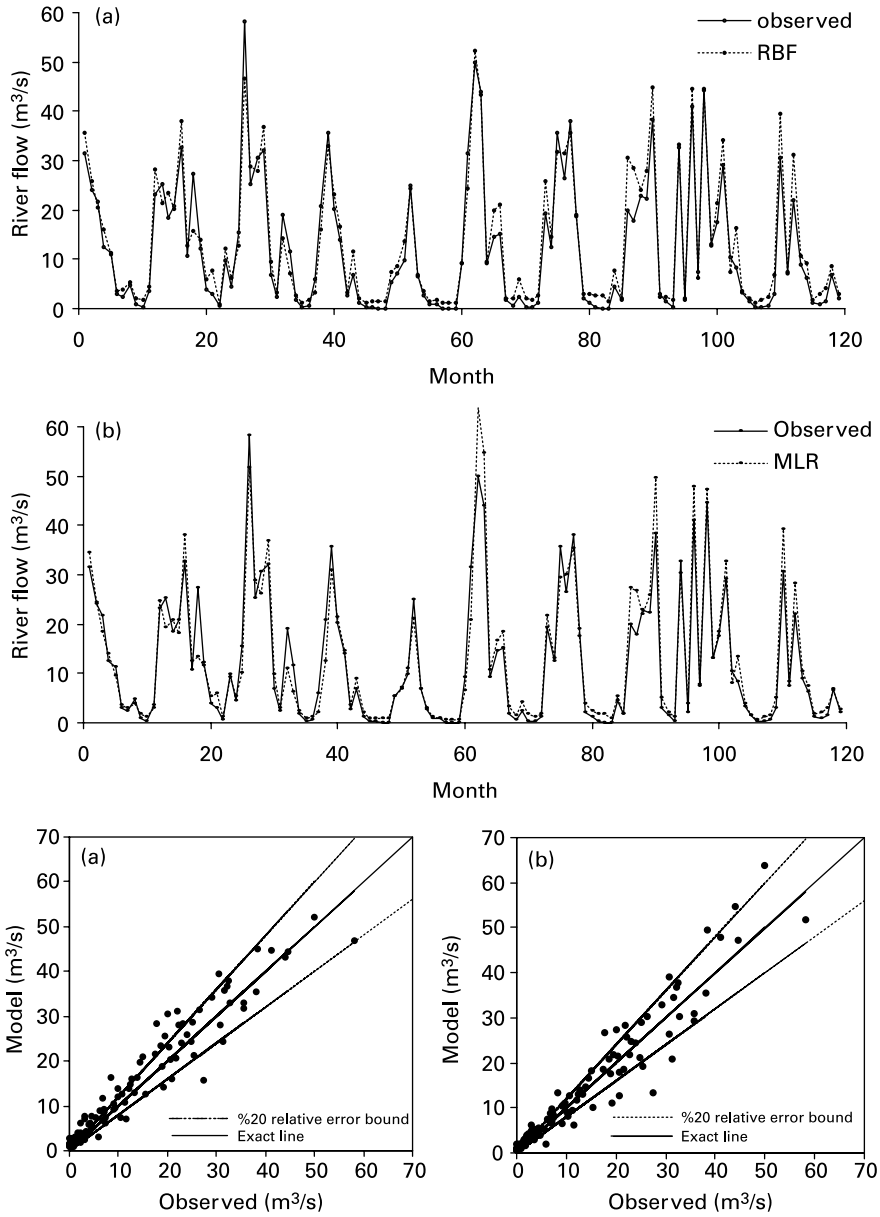

Figure 14 | Plotting of estimation performances for the test period using (a) RBF and (b) MLR.

application also the first 348 monthly flow data were used for training and the remaining 120 months were used for testing. Cross correlation value between two stations was obtained as 0.94. The various input combinations were tried. The MSE statistics of different ANN applications in monthly flow estimation are given in Table 8. The best performance criteria ($\mathrm{MSE} = 13.20\,\mathrm{m}^6/\mathrm{s}^2$, $R^2 = 0.936$) were obtained for the RBF model whose inputs are the monthly flows at time *t-1* and *t*. The FFNN also gave the lowest MSE for the input combination (iii). However, for the GRNN and MLR the lowest MSE was obtained for the input combination (i). The MLR showed better performance than the GRNN. The observed and estimated flows by RBF and MLR are shown in Figure 14 in the form of hydrograph and scatter plot. It can be seen that the RBF estimates are in agreement with the observed hydrographs.

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

The 20% lower and upper relative error bounds on the scatter plot were obtained by computing 20% relative error of observed flows. None of the high flows ($>35\,m^3/s$) are outside the 20% relative error bounds. Some estimates of low and medium flows exceed the lower and upper error bounds slightly (scatter plot (a), $R^2 = 0.936$). The MLR provides a more scattered plot (scatter plot (b), $R^2 = 0.926$) with poorer performance criteria. There are some high flow estimations outside the 20% relative error bounds.

CONCLUSIONS

1. This study indicated that forecasting and estimation of the monthly streamflow could be possible through the use of ANN.
2. The GRNN gave better performance than the FFNN and RBF techniques in one month ahead streamflow forecasting
3. The RBF performed better than the FFNN in monthly flow forecasting.
4. The FFNN may give some negative forecasts that are meaningless physically for the low flows.
5. It was found that the periodicity component considerably increases the models' performances especially for the ANN techniques.
6. The RBF and FFNN were found to be better than the GRNN in monthly river flow estimation using nearby river data.
7. Unlike the flow forecasting, the periodicity component mostly decreases the performance of the models in monthly river flow estimation.

REFERENCES

Broomhead, D. & Lowe, D. 1988 Multivariable functional interpolation and adaptive networks. *Complex Syst.* **2**, 321–355.

Chang, F-J. & Chen, Y-C. 2001 A counterpropagation fuzzy-neural network modeling approach to real time streamflow prediction. *J. Hydrol.* **245**, 153–164.

Cigizoglu, H. K. 2003 Estimation, forecasting and extrapolation of flow data by artificial neural networks. *Hydrol. Sci. J.* **48**(3), 349–361.

Cigizoglu, H. K. 2004 Estimation and forecasting of daily suspended sediment data by multi layer perceptrons. *Adv. Wat. Res.* **27**, 185–195.

Cigizoglu, H. K. & Kisi, O. 2005 Flow prediction by three back propagation techniques Using k-fold partitioning of neural network training data. *Nordic Hydrol.* **36**(1), 49–64.

Demuth, H. & Beale, M. 2004 *Neural Network Toolbox: For use with MATLAB®, User's Guide*. The MathWorks, MA, USA.

El-Bakyr, M. Y. 2003 Feed forward neural networks modeling for K-P interactions, chaos, solutions and fractals. *Chaos, Solutions and Fractals* **18**(5), 995–1000.

Hagan, M. T. & Menhaj, M. B. 1994 Training feed forward networks with the Marquaradt algorithm. *IEEE Trans. Neural Networks* **6**, 861–867.

Haykin, S. 1994 *Neural Networks: a Comprehensive Foundation*. IEEE Press, New York.

Hipel, K. W. 1986 Time series analysis in perspective. *Wat. Res. Bull.* **21**(4), 609–623.

Karunanithi, N., Grenney, W. J., Whitley, D. & Bovee, K. 1994 Neural networks for river flow prediction. *ASCE J. Comp. Civil Eng.* **8**(2), 201–220.

Kim, B., Kim, S. & Kim, K. 2003 Modelling of plasma etching using a generalized regression neural network. *Vacuum* **71**, 497–503.

Kisi, O. 2004 River flow modeling using artificial neural networks. *ASCE J. Hydrol. Eng.* **9**(1), 60–63.

Kisi, O. 2005 Daily river flow forecasting using artificial neural networks and auto-regressive models. *Turk. J. Eng. Environ. Sci.* **29**, 9–20.

Maier, H. R. & Dandy, G. C. 1996 Use of artificial neural networks for prediction of water quality parameters. *Wat. Resour. Res.* **32**(4), 1013–1022.

Poggio, T. & Girosi, F. 1990 Regularization algorithms for learning that are equivalent to multilayer networks. *Science* **2247**, 978–982.

Sahoo, G. B. & Ray, C. 2006 Flow forecasting for a Hawaii stream using rating curves and neural networks. *J. Hydrol.* **317**, 63–80.

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

Sivakumar, B., Jayawardena, A. W. & Fernando, T. M. K. G. 2002 River flow forecasting: use of phase space reconstruction and artificial neural networks approaches. *J. Hydrol.* **265**, 225–245.

Smith, J. & Eli, R. N. 1995 Neural-network models of rainfall-runoff process. *ASCE J. Wat. Resour. Plan. Manag.* **121**(6), 499–508.

Specht, D. F. 1991 A general regression neural network. *IEEE Trans. Neural Networks* **2**(6), 568–576.

Sudheer, K. P. & Jain, S. K. 2003 Radial basis function neural network for modeling rating curves. *ASCE J. Hydrol. Eng.* **8**(3), 161–164.

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

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest

Sudheer, K. P., Gosain, A. K., Rangan, D. M. & Saheb, S. M. 2002*b* Modelling evaporation using an artificial neural network algorithm. *Hydrol. Processes* **16**, 3189–3202.

Taurino, A. M., Distante, C., Siciliano, P. & Vasanelli, L. 2003 Quantitative and qualitative analysis of VOCs mixtures by means of a microsensors array and different evaluation methods. *Sens. Actuators* **93**, 117–125.

Tokar, A. S. & Johnson, P. A. 1999 Rainfall-runoff modeling using artificial neural networks. *ASC J. Hydrol. Eng.* **4**(3), 232–239.

Tsoukalas, L. H. & Uhrig, R. E. 1997 *Fuzzy and Neural Approaches in Engineering*. Wiley, New York, USA.

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

First received 14 June 2005; accepted in revised form 31 May 2007

Downloaded from http://iwaponline.com/hr/article-pdf/39/1/27/364545/27.pdf
by guest