Similarity-based error prediction approach for real-time inflow forecasting

Mahmood Akbari and Abbas Afshar

ABSTRACT

Regardless of extensive researches on hydrologic forecasting models, the issue of updating the outputs from forecasting models has remained a main challenge. Most of the existing output updating methods are mainly based on the presence of persistence in the errors. This paper presents an alternative approach to updating the outputs from forecasting models in order to produce more accurate forecast results. The approach uses the concept of the similarity in errors for error prediction. The K nearest neighbor (KNN) algorithm is employed as a similarity-based error prediction model and improvements are made by new data, and two other forms of the KNN are developed in this study. The KNN models are applied for the error prediction of flow forecasting models in two catchments and the updated flows are compared to those of persistence-based methods such as autoregressive (AR) and artificial neural network (ANN) models. The results show that the similarity-based error prediction models can be recognized as an efficient alternative for real-time inflow forecasting, especially where the persistence in the error series of flow forecasting model is relatively low.

Key words | K nearest neighbor (KNN), new data, real-time inflow forecasting, similarity-based error prediction

Mahmood Akbari (corresponding author)
Department of Civil Engineering,
University of Kashan,
Kashan,
Iran
E-mail: *makbari@kashanu.ac.ir*

Abbas Afshar
Department of Civil Engineering,
Iran University of Science and Technology,
Tehran,
Iran

INTRODUCTION

Inflow forecasts are a fundamental requirement for managing water resources systems and the successful operation of river-reservoir systems. Increased computer capacity has led to the increased use of hydrological models in forecasting. Along with sophisticated models and longer time series, also the data acquisition systems that are used to collect real-time hydrologic data have improved and made it possible to use up-to-date information from basins in real-time forecasting.

The forecasting models that operate in real-time are often supported by observed inflow or water level at the time of forecasting. This feedback process of assimilating the measured data into the forecasting procedure to improve the performance of a real-time forecasting system is referred to as updating (Refsgaard 1997). The updating procedures may concentrate on input variables, state variables, model parameters, and output variables depending on the variables modified during the feedback process (World Meteorological Organization (WMO) 1992; Refsgaard 1997).

If forecasting interest is limited to only a few variables at some specific locations with a high degree of accuracy and for a considerably long forecast lead-time, a data assimilation scheme based on the updating of output variables may be the most suitable approach (Babovic *et al.* 2001). The key advantage of an output updating procedure is the simplicity of its application in a totally automated way to any flow forecasting model (hereafter called primary model), without any need to alter its structure and physical meaning, or its operational implementation (Brath *et al.* 2002). This method, which is often called error prediction, has been widely used in different fields of real-time forecasting as well as in inflow forecasting. In this method, an error prediction model is provided in the updating mode to estimate the errors likely to occur in the next

doi: 10.2166/nh.2013.098

time step without addressing the contribution of the different sources of error. The predicted error is then added as a correction to the corresponding forecast of the primary model (hereafter called simulated flow) to provide the updated forecast. The primary model, operating together (i.e., coupled) with its output forecast updating procedure, is known as a real-time forecasting model (Becker & Serban 1990). In real-time inflow forecasting, incorporating the knowledge of the prediction errors of the past forecast to the forecasting models of different horizons can greatly improve performance of models (Khu *et al.* 2001). As the corrections are made by the difference between the simulated and observed values (hereafter called simulation errors), the flow data have to be highly reliable (Lundberg 1982).

Error prediction methods can be applied to any type of forecasting model since they are based on the time series modeling of the forecast error series (Bell & Moore 1998).

The most widely used models in error prediction are linear models such as the autoregressive (AR) and autoregressive moving average (ARMA) models in which the error prediction at each time step is assumed to be a linear function of the errors obtained from previous time steps (Shamseldin & O'Connor 1999; Toth *et al.* 1999; Xiong & O'Connor 2002; Xiong *et al.* 2004; Goswami *et al.* 2005). Along with traditional linear stochastic models, nonlinear time series models such as artificial neural networks (ANNs) have been applied for error prediction (Shamseldin & O'Connor 2001; Brath *et al.* 2002; Xiong & O'Connor 2002; Abebe & Price 2003; Goswami *et al.* 2005). Genetic programming (GP) as a nonlinear prediction method has also been used for real-time inflow forecasting (Khu *et al.* 2001, 2004). In addition to error prediction method, other types of output updating procedures have been used in which previously observed data along with forecasted inflows are used as inputs for a linear or nonlinear prediction model to produce directly updated inflow forecasts (Xiong *et al.* 2004; Goswami *et al.* 2005).

Output updating processes in these models are based on the presence of persistence (hereafter called persistence-based models) in the error series of forecasting models. These approaches may result in reliable forecasts if the structure of the simulation-mode forecast error time series exhibits high persistence (Xiong & O'Connor 2002).

As outlined, in the persistence-based models, the simulation errors in latest time steps are an indisputable part of these models and are used as the inputs to the error prediction models. In comparison with the persistence-based methods, this study takes advantage of the concept of similarity in the errors and develops similarity-based models for error prediction. The similarity-based error prediction models, as addressed in this paper, estimate the current error based on the most similar errors to the state of the current time step. Conceptually speaking, the similarity-based error prediction models assume that the current error will be close to the values of the similar simulation errors. The similar error cases for current time step can be found from historical error cases set according to the similarity degree between them and the historical error cases. The premise behind the similarity-based models for error prediction is that they can mimic what has happened in the past under situations similar to the current time. In fact, in these models, it is expected that the similar input variables to the flow forecasting model produce more or less similar outputs and hence relatively similar errors. In this regard, the K nearest neighbor (KNN) algorithm, which is one of the most popular algorithms in pattern recognition, can be a proper choice to be used as a similarity-based error prediction model. KNN is based on the similarity assumption and prediction is made by the finite number of similar neighbors. In previous studies, KNN has been frequently applied for the non-updating mode of flow forecasting (Karlsson & Yakowitz 1987; Yakowitz 1987; Galeati 1990; Shamseldin & O'Connor 1996; Solomatine *et al.* 2008; Wu *et al.* 2009; Akbari *et al.* 2011).

The traditional KNN selected as the similarity-based error prediction model suffers from a main drawback when compared with the persistence-based error prediction models. While the persistence-based models use new information in the form of the latest forecast errors, the traditional KNN model is based only on historical calibration data where search is done for neighbors. In other words, the traditional KNN model does not take any advantage from the new observations. To overcome this deficiency, two other forms of KNN model are developed and investigated in this study in which the new observations are imported into the search space. The role of the new data to be utilized in data-driven models has previously been

investigated in other models like the ANN and adaptive neural fuzzy inference system models (Xiong *et al.* 2004; Akbari *et al.* 2009). Application of the proposed approach to two large catchments in Iran shows that the similarity-based error prediction models may be addressed as an efficient alternative for real-time inflow forecasting, especially where the error persistence in flow forecasting model is relatively low. The rest of the paper is organized as follows: in the first section the structure of real-time inflow forecasting is expressed. The study employs an ANN model as the primary model in the simulation mode for flow forecasting, and thus the ANN model is then described. Different error prediction models including earlier persistence-based and developed similarity-based models in the updating mode are also explained. In the next section, case studies are introduced and then the ANN model is used to simulate inflow series and, subsequently, all updating models are employed to update the flow estimates of the ANN model. Finally, discussions and conclusions are made on the basis of comparing the results from the different updating models.

METHODOLOGY AND MODELS

Real-time inflow forecasting

In order to develop a model for real-time inflow forecasting, a flow forecasting model is needed as a primary model and an error prediction model in the updating mode. The procedure of integrating the primary model and error prediction model can be expressed as follows and shown in Figure 1. First, the simulation errors of the primary model are obtained as:

$$e(t) = Q(t) - \hat{Q}(t) \tag{1}$$

where $e(t)$ denotes the simulation error of the selected primary model, $Q(t)$ and $\hat{Q}(t)$ are the observed and simulated flows, respectively.

The updating model at the current time step is then developed to predict $e(t)$ and the updated forecast is calculated as:

$$\hat{Q}(t) = \hat{Q}(t) + \hat{e}(t) \tag{2}$$

where $\hat{Q}(t)$ is the updated flow and $\hat{e}(t)$ is the estimate of $e(t)$ at time t (i.e., simulation error).

In the updating mode, the forecasts resulting from the primary model are subsequently modified, or updated, in accordance with the errors observed in the previous forecasts or the ones available in the calibration data set. This study focuses on the methodology how to forecast the simulation error of the primary model.

ANN model: primary model

The ANN model can capture the nonlinear relationships involved in the rainfall–runoff process, an extremely complex physical process, which is not clearly understood (Zhang & Govindaraju 2000). Examples of the applications can be found in such references as Karunanidhi *et al.* (1994), Hsu *et al.* (1995), Minns & Hall (1996), Shamseldin (1997), Jain & Indurthy (2003), Rajurkar *et al.* (2004) and Chau *et al.* (2005).

Figure 2 shows the general structure of the multi-layer feed-forward neural network used in the present study. Hornik *et al.* (1989) proved that a single hidden layer network containing a sufficiently large number of neurons can be used to approximate any measurable functional relationship between the input variables and the output variable to any desired accuracy. Thus, the use of a single hidden layer is generally recommended and this recommendation has been adopted in the present study as well as in many

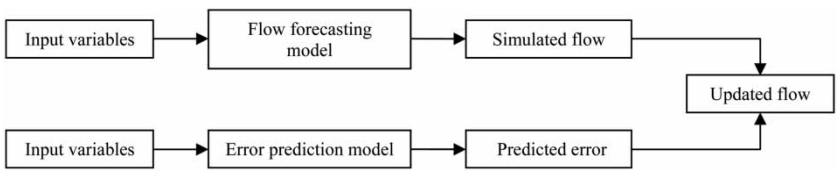

Figure 1 | A schematic diagram of the real-time flow forecasting.

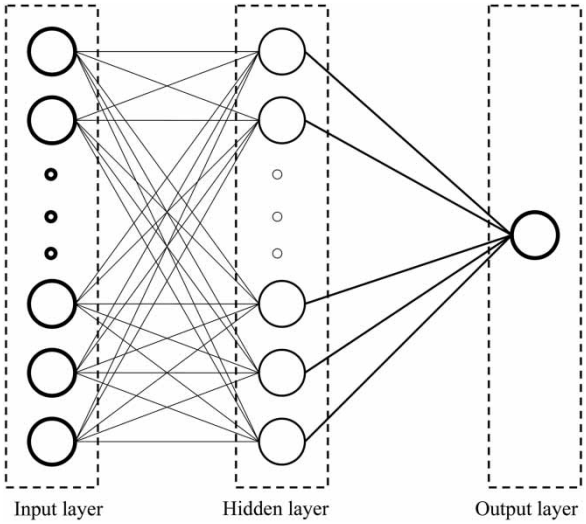

Figure 2 | A schematic diagram for the multi-layer feed-forward neural network.

of the applications of the multi-layer feed-forward networks to the field of hydrology and water resources. The main differences among the various types of ANN involve network architecture and the method for determining the weights and the activation functions for inputs and neurons (training) (Caudill & Butler 1992). In the present work, a scaled conjugate gradient algorithm (Moller 1993) is used for training. In addition, one of the most widely used nonlinear activation functions, namely log-sigmoidal function, is chosen for the hidden nodes and a linear activation function is employed for the output layer. It is necessary to mention that because the log-sigmoidal function can take on values ranging in the (0,1) domain, a normalization of the values of the input and output variables needs to be done. In this study, the variables are scaled to the range (0,1) by using the original data range of the corresponding variable as a scalar. In addition, this transformation is done to ensure that all variables receive equal attention during the training process (Bowden *et al.* 2003) and to avoid the numerical difficulties during the calculation (Lin *et al.* 2006).

The determination of the number of neurons in input and hidden layers is an essential task for each ANN model. In the process of model development, several network architectures with a different number of input neurons in input layer with a varying number of hidden neurons are usually considered to select an appropriate architecture of the network. If p is the number of inputs and q is the number of neurons used in the hidden layer of a three layer ANN model with an output node then the number of the weights optimized in the ANN model is equal to $(p+1)q+q+1$.

In this study, apart from the fact that the ANN model is employed as the primary model for flow forecasting in the simulation mode, the ANN is used as one of the persistence-based error prediction models in the updating mode.

Persistence-based approach for error prediction

The persistence-based approach uses two of the most commonly used models for error prediction, namely the AR model and ANN model.

The classic model of the persistence-based structure of a time series is the linear univariate AR model. This technique takes advantage of the dependence of model errors by characterizing this dependency through a weighted combination of the most recent prediction errors. Mathematically, this updating model is expressed as:

$$\hat{e}(t) = a_1 e(t-1) + a_2 e(t-2) + \ldots + a_p e(t-p) \quad (3)$$

in which a_i for $i = 1$ to p are the parameters of the AR model and p is the order of the AR model for error prediction. The order of persistence (p) can be investigated by examining the autocorrelation (AC) function of a prediction error time series generated from calibration data and finally is determined in the trial and error tests.

The second persistence-based error prediction model is an ANN model which uses the previous forecast errors (up to the time of making the new forecast) as the inputs to the ANN model as well as the AR error prediction model. The use of the ANN model is prompted to capture possible nonlinear relationships among the forecast error series of the primary model, hence improving the updating performance over the linear AR model (Xiong & O'Connor 2002).

Mathematically, this error prediction model is expressed as:

$$\hat{e}(t) = f_{\mathrm{ANN}}(e(t-1), e(t-2), \ldots, e(t-p)) \quad (4)$$

where f_{ANN} represents the nonlinear structure implied in the ANN model. Clearly, for the ANN updating model, the number of input nodes represent the order of persistence p and output node includes only the single output, $\hat{e}(t)$.

Similarity-based approach for error prediction

In this approach, the KNN algorithm is used for the error prediction of flow forecasting model, $\hat{e}_t$. To construct the error prediction model, a set of error calibration data is required. These data may be obtained from the error time series generated from the primary flow forecasting model. The error predicted by the KNN would be close to the error values of the nearby points from the error calibration data. The similarity or closeness of a query instance (error in this study) is estimated according to the closeness of its feature vector with those of data available in calibration data. For this purpose, some or all of the input variables to the primary flow forecasting model such as previous rainfall and flow variables can be considered as the feature vector elements of the error for determining the similarity.

The traditional KNN combines the target values of K selected neighbors from error calibration data, to predict the target value of the given error test pattern.

A weighted Euclidian norm is usually used to measure the closeness (similarity) of the feature vector of query error (X_t) and any feature vector of the calibration data set (X_i):

$$\mathrm{d}(X_t, X_i) = \sum_{j=1}^{m} w_j^a (x_{ij} - x_{tj})^2 \tag{5}$$

where i, j are the indices for data (error values) and attributes, respectively, w_j^a is the weight of each attribute, m is the number of attributes, and x_{ij} is the normalized value of jth attribute of ith error value in the calibration data set. It should be noted that each attribute is normalized to minimize the scale difference. Given the output values of neighbors (e_i), the predicted output for the test pattern ($\hat{e}_t$) is calculated as follows:

$$\hat{e}_t = \sum_{i=1}^{K} w_i^N e_i / \sum_{i=1}^{K} w_i^N \tag{6}$$

where w_i^N is the weight of each neighbor and K is the number of neighbors. In the simple form of KNN, w_i^N is employed to be $1/K$ and the estimate is the mean value of KNNs. In the modified form of KNN, however, each neighbor is usually given a weight based on the distance between the neighbor and the test pattern. A farther neighbor receives a smaller weight, which reduces its effect on the prediction results compared to other closer neighbors. In this way, some monotonically decreasing kernel functions (i.e., linear, inversion, square inverse, exponential, Gaussian) have been used (Wand & Schucany 1990; Aha & Goldstone 1992; Ruprecht & Müller 1994; Solomatine *et al.* 2008), which are defined as follows:

$$\begin{aligned}
&\text{(a) Linear } w_i^N = 1 - \mathrm{d}(X_q, X_i)\\
&\text{(b) Inversion } w_i^N = (\mathrm{d}(X_q, X_i))^{-1}\\
&\text{(c) Square inverse } w_i^N = ((\mathrm{d}(X_q, X_i))^2)^{-1}\\
&\text{(d) Exponential } w_i^N = \exp(-\mathrm{d}(X_q, X_i))\\
&\text{(e) Gaussian } w_i^N = \exp(-(\mathrm{d}(X_q, X_i))^2)
\end{aligned} \tag{7}$$

Atkeson *et al.* (1997) claimed there is no clear evidence that a specific kernel function is always superior to others; however, some outperformed others on some data sets (Solomatine *et al.* 2008). The number of nearest neighbors which should be considered in estimation is a challenging issue itself. Indeed, no well-established methods exist for selecting an optimal K for using the KNN. Wu *et al.* (2009) adopted K as $m + 1$ in a monthly streamflow prediction model where the attributes were all the lagged monthly streamflows. However, in practice, the number of neighbors is often chosen empirically by cross-validation or domain experts (Kang & Cho 2008). In this study, the KNN is traditionally calibrated off-line using the calibration data set which results in the best kernel function, the optimal number of nearest neighbors, and the weights of attributes (w_j^a).

The similarity-based approach for error prediction employs the KNN algorithm in three different models. The difference in the three proposed KNN models is in the database where the search is done for the neighbors. As done traditionally, the first model, (KNN-1), chooses the nearest neighbors among the historical calibration data. The

traditional form of KNN does not take advantage of new observations unlike the persistence-based models which use the latest forecasts. In the real-time inflow forecasting system, when new data become available, they can be considered to be utilized in the KNN model as well as the calibration data set. The second and third models, namely KNN-2 and KNN-3, use the new observations where the search is made for the neighbors. Specifically, in KNN-2, the newly observed data are also supposed to be available for searching the neighbors as well as historical calibration data. The role of new data in this model is twofold. First, they can expand the size of data set used in the KNN model as a data-driven model which can be valuable. Second, it has greater worth, especially if there are some changes in the physical system of a catchment; in this case, KNN-2 can partially reflect these changes by searching the nearest neighbors among the latest observed data in addition to the historical data.

The last model from the similarity-based category (KNN-3) is a KNN model in which only errors in the most recent time steps are used as the neighbors for the error prediction. In fact, this model, as persistence-based models, assumes that persistence is available in the forecast errors and error prediction is made only by the errors in the latest forecasts up to the time of making the new forecast. However, the contribution of the errors in the previous time steps for error prediction, unlike the persistence-based models, is no longer fixed and changes for different time steps in accordance with the closeness of the feature vector of error at the current time step with the ones in the previous time steps.

Thus, this model benefits from both persistence in the errors and the similarities in the feature vectors of the errors. The number of neighbors in this model also represents the order of the persistence in the errors (i.e., $K = p$).

APPLICATION

Study catchments

The data used in this study were taken from the Karoon and Dez catchments located in the southwest of Iran with sizes of about 25,000 and 22,000 km^2 and times of concentration about 36 and 23 hours, respectively. There exist two main reservoirs at these catchments, namely the Karoon1 and Dez reservoirs, mainly designed for power generation. Daily reservoir inflow predictions are essential to the operational planning and scheduling of these hydroelectric power systems. In this study, the Soosan and Talezang gauging stations located at the upstream of the Karoon1 and Dez reservoirs, respectively, were taken as examples, and real-time inflow forecasting at these stations for the lead time of 1 day was considered (Figure 3). The rainfall and flow data from some synoptic and hydrometric gauging stations distributed in the catchments are available and can be used as

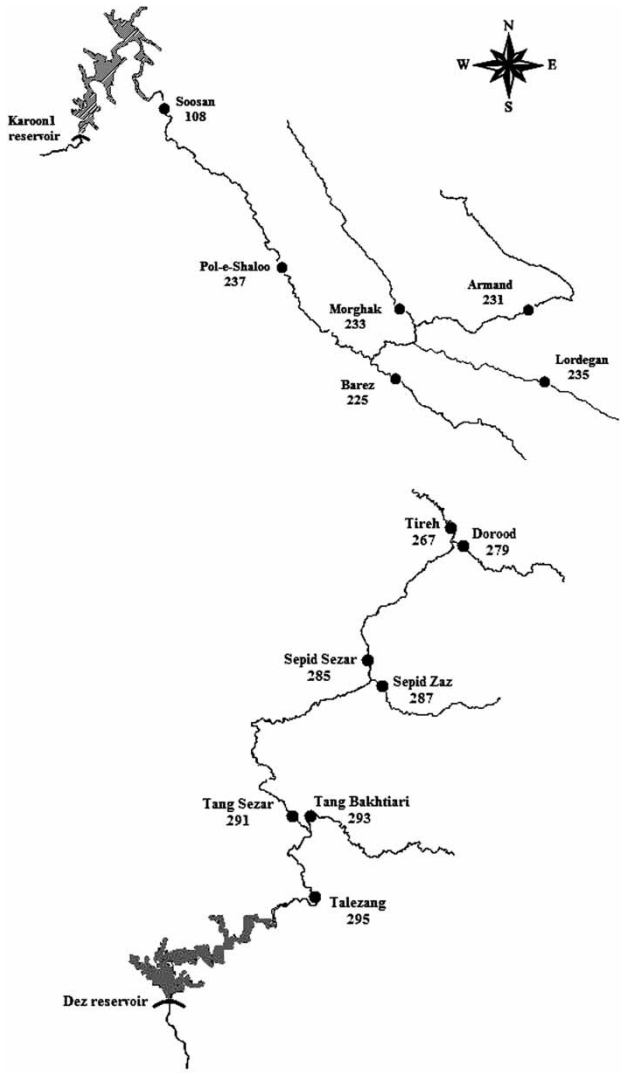

Figure 3 | The Karoon1 and Dez reservoirs and the upstream gauging stations. (Numbers represent the national code number of each gauging station in Iran's telemetry system.).

Table 1 | The statistical parameters for the data sets of the Soosan and Talezang gauging stations

	Calibration					Verification				
Gauging station	Period	X_{min}	X_{mean}	X_{max}	S_x	Period	X_{min}	X_{mean}	X_{max}	S_x
Soosan	1994–2000	79.8	285.5	2,479	223.9	2001–2003	69.6	272.8	2,233	217.9
Talezang	1982–1987	0.9	252.8	2,416	215.1	1988–1990	0.2	253.2	2,689	221.2

X: flow (m^3/s).

inputs for the forecasting models. Table 1 shows the statistical parameters (i.e., minimum value: X_{min}, maximum value: X_{max}, mean: X_{mean}, standard deviation: S_x) for the daily flow data sets of the Soosan and Talezang gauging stations. As can be seen from the table, the splitting of the data into calibration and verification data sets has been done so that both data sets have relatively similar statistics. In addition, the periods of verification data for both cases are selected after the periods of calibration data. This is consistent with the role of the new data in the KNN-2 model, as forecasting proceeds in a real system, the new data are obtained. The primary model and the updating procedures are calibrated and verified using the same calibration and verification periods in order to assess improvement in the forecast accuracy for each data set separately.

Performance criteria: model evaluation

Numerous statistical goodness-of-fit criteria have been proposed in the literature for evaluating hydrologic modeling results. We consider the Nash–Sutcliffe coefficient of efficiency (NSE) (Equation (8)) (Nash & Sutcliffe 1970) and average absolute relative error (AARE) (Equation (9)) as more common criteria. The Nash–Sutcliffe coefficient of efficiency which provides a measure of the ability of a model to predict values that are different from the mean has the following form:

$$\begin{aligned} \mathrm{NSE} &= \left(1 - \frac{F}{F_0}\right) \\ F &= \sum (Q_{\text{observed}} - Q_{\text{predicted}})^2 \\ F_0 &= \sum (Q_{\text{observed}} - Q_{\text{mean observed}})^2 \end{aligned} \tag{8}$$

The initial variance F_0 can be perceived as a measure of the performance of a no knowledge model having the mean of the observed flow time series as its flow forecast in all times. Thus, the NSE value is a global measure of comparing the predicted value with the overall mean value. In both the calibration and the verification periods, the initial variance is calculated using the mean flow of the calibration period (Goswami & O'Connor 2010). It should be recognized that although the NSE criterion has frequently been used in the literature for the model evaluation, there is no consensus on what NSE level guarantees a satisfactory performance. Nonetheless, this criterion can be used to compare the relative merits of the models in which the closer the Nash index is to 1, the better the performance of the model.

The AARE is calculated as follows:

$$\mathrm{AARE} = \frac{1}{N}\sum \left|\frac{(Q_{\text{observed}} - Q_{\text{predicted}})}{Q_{\text{observed}}}\right| \times 100\% \tag{9}$$

where N is the number of data used for evaluation. The error statistics based on the percentage error in prediction with respect to observed value (such as AARE) may be better for performance evaluations as they give appropriate importance to all magnitude flows (low, medium, or high) (Jain & Kumar 2007). The coefficient of efficiency, however, tends to give greater attention to the high magnitude flows due to the involvement of the square of the difference between the observed and predicted flows. Therefore, in this evaluation, the errors in estimating low magnitude flows are dominated by the errors in estimating high magnitude flows.

ANN model for simulation-mode flow forecasting

As stated earlier, the ANN model is the flow forecasting model selected to estimate flow for each catchment for non-updating simulation mode. In a data-driven model, an important step is the choice of the input variables

representing the system to be modeled. The rainfall and flow data from the upstream gauging stations are among the most important variables affecting the inflow to a reservoir and thus may efficiently be used in the data-driven models. Bowden *et al.* (2005) reviewed different approaches for choosing the input variables for the data-driven models in general with emphasis on the ANN ones. This study employs cross-correlation and mutual information analyses to identify the suitable flow and rainfall variables from upstream gauging stations to be used in the ANN model to form the input patterns and define an appropriate range of the model structures to be investigated. As stated earlier, the networks formed by only one hidden layer are tested. Finally, a trial and error procedure, recommended as the best strategy by Shamseldin (1997), is used to determine the best combination of input variables and also the number of nodes in the single hidden layer for the optimal configuration of the ANN model. Table 2 presents the best configuration of the ANN models for each catchment and the performances of forecasting over calibration and verification data sets. The table shows the predicted discharge of day t at the Soosan gauging station, namely $Q_{108}(t)$, will depend on the following variables: the discharges of days t–1 and t–2 at the Soosan gauging station ($Q_{108}(t–1)$ and $Q_{108}(t–2)$, respectively), the discharges of day t–1 at the Pol-e-Shaloo, Armand, and Barez gauging stations ($Q_{237}(t–1)$, $Q_{231}(t–1)$, and $Q_{225}(t–1)$, respectively), the rainfall depths of day t–1 at the Barez and Lordegan gauging stations ($R_{225}(t–1)$ and $R_{235}(t–1)$, respectively). Also, it is observed that the predicted discharge of day t at the Talezang gauging station, namely $Q_{295}(t)$, is related to these variables: the discharges of days t–1 and t–2 at the Talezang gauging station ($Q_{295}(t–1)$ and $Q_{295}(t–2)$, respectively), the discharges of day t–1 at the Tang Bakhtiari, Tang Sezar, Sepid Zaz, and Sepid Sezar gauging stations ($Q_{293}(t–1)$, $Q_{291}(t–1)$, $Q_{287}(t–1)$, and $Q_{285}(t–1)$, respectively), the rainfall depths of day t–1 at the Talezang and Sepid Sezar gauging stations ($R_{295}(t–1)$ and $R_{285}(t–1)$, respectively). Note that in the case of the Dez catchment, the predicted flow in the current day t for the Talezang gauging station, namely $Q_{295}(t)$, depends on the unseen rainfall on the day ($R_{295}(t)$). In this study, the historical predictions of rainfall were not available for model development; hence, historical observations are used for both the model calibration and verification. Use of the perfect foresight of rainfall input over the lead time of the output forecast eliminates the effects of the errors in the meteorological forecasts when they are used as the inputs to the primary model. Therefore, in the updating mode, the relative merits of the selected forecast updating procedures can be assessed more objectively (Shamseldin & O'Connor 2001).

Forecast errors analysis

Having determined the forecasts by the ANN model, an analysis of the errors of the simulated flows over calibration data for each catchment was done and the plots of autocorrelation functions of the flow forecast error series are presented in Figure 4.

Table 2 | The performances of the ANN models for the test catchments

		NSE		AARE (%)	
Catchment	**Optimal configuration**	**Calibration**	**Verification**	**Calibration**	**Verification**
Karoon	Number of input nodes = 7; including Q_{108} (t–1), Q_{108} (t–2), Q_{237} (t–1), Q_{231} (t–1), Q_{225} (t–1), R_{225} (t–1), R_{235} (t–1)	0.866	0.814	16.12	15.58
	Number of hidden nodes = 9				
	Number of output nodes = 1; including Q_{108} (t)				
Dez	Number of input nodes = 8; including Q_{295} (t–1), Q_{295} (t–2), Q_{293} (t–1), Q_{291} (t–1), Q_{287} (t–1), Q_{285} (t–1), R_{295} (t), R_{285} (t–1)	0.830	0.789	17.73	16.29
	Number of hidden nodes = 11				
	Number of output nodes = 1; including Q_{295} (t)				

Q: flow; R: rainfall.
Subscripts represent the national code number of each gauging station in Iran's telemetry system (see Figure 3).

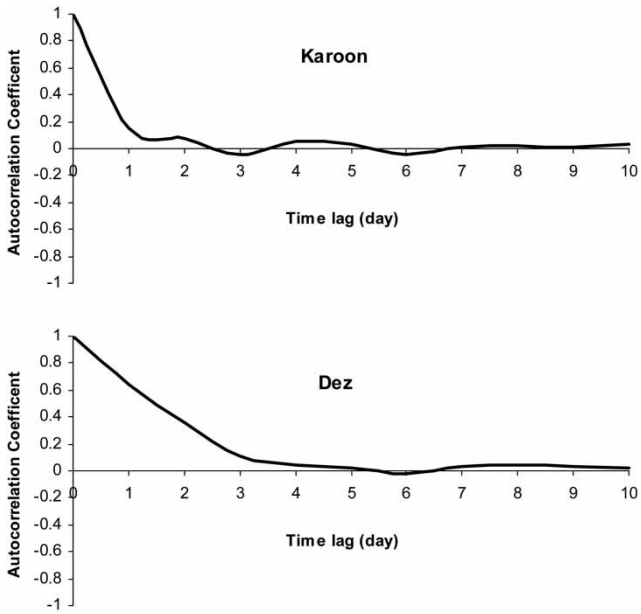

Figure 4 | Autocorrelation coefficients of the flow forecast error series.

Analysis of Figure 4 indicates that, in the case of the Karoon catchment, the maximum autocorrelation of the error time series is 0.15 for a 1 day lag and it is close to zero for other lags. These values strongly reject the hypothesis of the presence of the persistence in the prediction error in the Karoon catchment. However, in the case of the Dez catchment, the simulation-mode error series show higher persistence as compared to the Karoon catchment, since the corresponding autocorrelation coefficient values are significantly different from zero. Therefore, one may conclude that the persistence-based updating procedures may improve the forecasts in the Dez catchment, while they may not be so efficient for the Karoon catchment, since these techniques rely upon the presence of the persistence in the prediction errors. In contrast, the similarity-based updating procedures can be expected to perform better in updating than the persistence-based updating procedures, especially for catchments such as the Karoon, where the presence of persistence is not observed.

Implementation of the updating procedures

This section applies the persistence-based and similarity-based procedures for forecasting the simulation error on flows, $e(t)$. As stated earlier, the order of persistence (p) for the AR and ANN updating models is first investigated using AC function of a prediction error time series generated from the calibration data and finally determined by trial and error tests. Based on the results of the AC functions shown in Figure 4, the maximum number of the persistence was limited to three and five in the cases of the Karoon and Dez catchments, respectively, both for the AR and ANN models, in trial and error tests. The comparisons of the efficiencies of the discharges updated with varying degrees of the persistence showed that AR(2) and AR(3) were the best configurations of the AR models for the Karoon and Dez catchments, respectively. Also for the ANN models, the more parsimonious networks among the better performing ones were preferred as the error prediction models for the catchments. As a consequence, the ANN with two input nodes and three nodes at the hidden layer was chosen as the error prediction model at the Soosan gauging station at the Karoon catchment. In addition, the results indicated the ANN with four input nodes and seven nodes at the hidden layer is the optimal configuration for the Talezang gauging station at the Dez catchment.

In the KNN-1 model, cross-correlation and mutual information analyses were done to study the relation between different input variables to the ANN flow forecasting model and the model errors for each catchment. The resulting information was used as a guide in a trial and error process, to select the most related attributes at which they can be best applied in the KNN-1 error prediction model. Consequently, the KNN-1 models were configured with the attributes of $Q_{108}(t-1)$, $Q_{231}(t-1)$, $Q_{225}(t-1)$, $R_{225}(t-1)$, and $R_{235}(t-1)$ for the Karoon catchment and the attributes of $Q_{295}(t-1)$, $Q_{293}(t-1)$, $Q_{291}(t-1)$, $Q_{285}(t-1)$, $R_{295}(t-1)$, and $R_{285}(t-1)$ for the Dez catchment. The same attributes used in the KNN-1 model for each catchment are maintained for the KNN-2 and KNN-3 models. Updated forecast is obtained by adding the predicted errors to the simulated flow forecasts.

RESULTS AND DISCUSSION

Tables 3 and 4 show the performance of the flow forecasts updated with the best performing configurations identified

Table 3 | The performances of different updating procedures on the forecasts of the ANN model for the Karoon catchment

Updating model	Optimal configuration	NSE		AARE (%)	
		Calibration	Verification	Calibration	Verification
AR	Autoregressive order, $p = 2$	0.870	0.827	15.72	14.08
ANN	Number of input nodes = 2 ($p = 2$) Number of hidden nodes = 3 Number of output nodes = 1; including $e(t)$	0.872	0.838	14.85	13.76
KNN-1	Number of neighbors, $K = 5$ Number of attributes, $m = 5$; including Q_{108} (t–1), Q_{231} (t–1), Q_{225} (t–1), R_{225} (t–1), R_{235} (t–1) Gausian weight function	**0.894**	0.879	**11.14**	10.54
KNN-2	Number of neighbors, $K = 5$ Number of attributes, $m = 5$; including Q_{108} (t–1), Q_{231} (t–1), Q_{225} (t–1), R_{225} (t–1), R_{235} (t–1) Gausian weight function	Not applicable	**0.902**	Not applicable	**9.63**
KNN-3	Number of neighbors, $K = 3$ Number of attributes, $m = 5$; including Q_{108} (t–1), Q_{231} (t–1), Q_{225} (t–1), R_{225} (t–1), R_{235} (t–1) Gausian weight function	0.871	0.837	13.64	12.28
No updating		0.866	0.814	16.12	15.58

Table 4 | The performances of different updating procedures on the forecasts of the ANN model for the Dez catchment

Updating model	Optimal configuration	NSE		AARE (%)	
		Calibration	Verification	Calibration	Verification
AR	Autoregressive order, $p = 3$	0.862	0.860	14.39	13.72
ANN	Number of input nodes = 4 ($p = 4$) Number of hidden nodes = 7 Number of output nodes = 1; including $e(t)$	**0.884**	**0.887**	12.66	11.96
KNN-1	Number of neighbors, $K = 4$ Number of attributes, $m = 6$; including Q_{295} (t–1), Q_{293} (t–1), Q_{291} (t–1), Q_{285} (t–1), R_{295} (t), R_{285} (t–1) Gausian weight function	0.846	0.844	12.93	12.81
KNN-2	Number of neighbors, $K = 4$ Number of attributes, $m = 6$; including Q_{295} (t–1), Q_{293} (t–1), Q_{291} (t–1), Q_{285} (t–1), R_{295} (t), R_{285} (t–1) Gausian weight function	Not applicable	0.863	Not applicable	11.19
KNN-3	Number of neighbors, $K = 5$ Number of attributes, $m = 6$; including Q_{295} (t–1), Q_{293} (t–1), Q_{291} (t–1), Q_{285} (t–1), R_{295} (t), R_{285} (t–1) Gausian weight function	0.878	0.862	**10.54**	**10.07**
No updating		0.830	0.789	17.73	16.29

for each updating procedure for the Karoon and Dez catchments, respectively.

Examination of Tables 3 and 4 reveals that all updating procedures can efficiently improve the simulation mode forecasts by means of the non-updated forecast errors of the primary model. However, it may be seen from the same tables that the efficiency of the updating procedures varies for different catchments. In the Dez catchment, the ANN performs better than other updating procedures in terms of NSE and the KNN-3 is the best in terms of AARE for both the calibration and the verification periods. In the case of the Karoon catchment, the KNN-1 performs better than the others for the calibration period while the KNN-2 updating procedure has the highest performance measure for the verification period. It should be noted that the KNN-2 is not applicable for the calibration data, since there is no additional knowledge from new data for this case and only the information related to the calibration data is available for this period. In the case of the verification data, the new data for any query instance comprise the data available in the verification period up to the time of making a prediction for it. Therefore, as forecasting proceeds in this case, more new data become available.

In the case of the Karoon catchment, as expected, the persistence-based updating procedures, namely the AR and ANN, are not so successful, since these models are based on the presence of the error persistence while the autocorrelation in the corresponding error series is weak. In contrast, the different types of the KNN models which rely on the similarity in the errors are more efficient to improve the accuracy of the forecasts. Better performance of these models may be attributed to the fact that they mimic what has happened in the past under a similar situation of the current time for error prediction. In this study, the input variables to the flow forecasting model are assumed to be reliable, so the main source of uncertainty in the model output is due to the inappropriate model structure. Therefore, one may assume relatively similar errors for the outputs of the similar input variables to the flow forecasting model.

Further investigation showed the contributions of new data to finally selected neighbors for the KNN-2 models were 29 and 24% for the Karoon and Dez catchments, respectively. The results shown in Tables 3 and 4 efficiently demonstrate the worth of the new data as used in the KNN-2 model. As stated earlier, apart from the fact that the new data can extend the information quantity included in the data, they can reflect the time-varying characteristics of the hydrological processes and, hence, the better performance for the KNN-2 over the KNN-1 is expected.

It was also found that among the persistence-based updating procedures, the ANN model is more effective than the AR model for both catchments. This superiority may be attributed to the nonlinear relations involved between the forecast errors of the primary flow forecasting models which can be captured better by the ANN model.

In the case of the Dez catchment, where the error series have a strong time persistence structure, the role of the persistence is more pronounced than the role of similarity and the ANN updating procedure performs better than the others in terms of efficiency index NSE. This shows that the earlier persistence-based methods are still superior to the KNN algorithms applied in this study, where persistence in the error series is relatively high. The efficiency of these models, however, varies from one catchment to another catchment, depending on the degree of the persistence in the structure of the error series of the primary model. Considering the improvement of the updating models on the AARE, one can see the best updating procedure in the Dez catchment is the KNN-3. This model which tracks the error values of the latest time steps as well as the persistence-based procedures, takes the advantages from the similarities in the feature vectors of the errors to determine the contribution of the previous errors for the error prediction. A remarkable deficiency of this method, however, is that in no case an error value higher than the error values of the latest time steps can be predicted. This drawback can especially be more crucial for the higher error values which unusually correspond to the higher flows. In other words, the KNN-3 cannot predict the high errors and, hence, the performance indices based on the square errors like the NSE decrease intensively while the AARE index is affected moderately.

In order to illustrate this drawback, the efficiency of the KNN-3 model as compared to the KNN-1 and KNN-2 models, on the different ranges of flow, was investigated. For this purpose, the magnitude of flow at the verification period for both catchments was divided into low

($X \leq X_{mean}$), medium ($X_{mean} < X \leq X_{mean} + 2S_x$) and high magnitude ($X > X_{mean} + 2S_x$) flows.

Table 5 compares the results of the models in terms of AARE over the verification data set on the different ranges of flow. The results show that although the KNN-3 is relatively efficient at forecasting low and medium flows, it is not successful at forecasting the high flows as compared with the KNN-1 and KNN-2 models for both catchments.

In order to illustrate the details of the forecast matching, the graphical comparisons of the observed, simulated, and updated hydrographs of the two catchments for the highest flood in the verification period are plotted in Figure 5. The updated forecasts are presented using only the best updating procedures in terms of NSE value for each catchment.

Table 5 | Comparison of the performance of the different types of the KNN models in terms of AARE for the different ranges of flow during the verification period for both catchments

Catchment	Karoon			Dez		
Range of flow	Low range	Medium range	High range	Low range	Medium range	High range
KNN-1	11.79	7.71	14.71	13.45	11.34	15.11
KNN-2	11.07	6.40	14.14	12.01	9.33	13.96
KNN-3	13.96	7.67	25.37	12.27	4.53	22.12

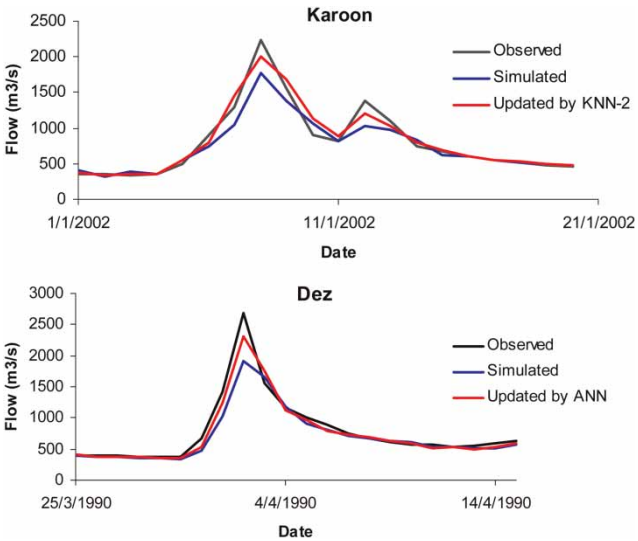

Figure 5 | Comparisons of the observed, simulated, and updated hydrographs of the catchments.

The similarity-based error prediction models described in this paper involved just one-step lead time updating by estimating the simulation error of the primary model. In order to make multi-step ahead predictions (i.e., the predictions with a lead time larger than one time step), two procedures may be employed: (1) use of a separate multi-step error prediction model for each lead time in which the feature vector includes the most recent data available at the forecast origin; and/or (2) use of a single one-step error prediction model for all lead times in which the exact value of some or all of the components of the feature vector may not be available for the lead times larger than one time step. The first procedure uses only the original data while the latter may use the estimated values of the components over the forecast lead time. In the latter case, recursive schemes may be used to obtain the feature vector estimations for successive lead time one after another. Since the estimation errors of the feature vector are propagated into subsequent estimations, a strong deterioration is evidenced for increasing lead times. Application of these methods and their efficiency in a multi-step ahead real-time inflow forecasting problem need to be investigated in the future.

CONCLUSION

This study introduced a similarity-based approach for error prediction of real-time inflow forecasting. The KNN algorithm was used as the main platform to implement the proposed similarity-based approach. It was shown that the modified KNN, which benefits from the new data in the search space, is superior to the traditional KNN.

The combination of similarity and persistence in error prediction was used to present another prediction model (KNN-3) in which the search space for neighbors is limited to the latest forecast errors. It was illustrated that this model is relatively efficient for the medium and low range of inflows.

Application of the proposed models revealed that using the similarities in the errors into the similarity-based KNN models may substantially improve inflow forecasting. If the similarities in errors are prevailing, the proposed models can successfully be applied in any flow forecasting model

for catchments of various sizes and lead times. Their performance and achievement in other hydrological settings with different persistence and similarities in errors need to be tested.

The results of this study indicated that the different forms of the KNN algorithm may be more successful than the persistence-based models where the persistence in the error series of the primary model is relatively weak. Although the proposed similarity-based updating procedures applied in this study can be utilized as efficient alternatives for real-time inflow forecasting, the persistence-based updating procedures are still effective tools for error prediction where the degree of persistence in the errors of the flow simulations is considerably high.

To extend the similarity-based error prediction models for multi-step ahead real-time flow forecasting, two different approaches were proposed and their performances need to be investigated.

ACKNOWLEDGEMENTS

The authors would like to thank the helpful comments of the three anonymous reviewers and Mr M. Fazel.

REFERENCES

Abebe, A. J. & Price, R. K. 2003 Managing uncertainty in hydrological models using complementary models. *Hydrol. Sci. J.* **48** (5), 679–692.

Aha, D. W. & Goldstone, R. L. 1992 Concept learning and flexible weighting. *Proceedings of 14th Annual Conference of the Cognitive Science Society*, Mahwaj, NJ, USA, 534–539.

Akbari, M., Afshar, A. & Rezaei, M. 2009 Fuzzy rule based models modification by new data: application to flood flow forecasting. *Water Resour. Manage.* **23**, 2491–2504.

Akbari, M., Van Overloop, P. J. & Afshar, A. 2011 Clustered K nearest neighbor algorithm for daily inflow forecasting. *Water Resour. Manage.* **25**, 1341–1357.

Atkeson, C. G., Moore, A. W. & Schaal, S. 1997 Locally weighted learning. *Artif. Intell. Rev.* **11** (5), 11–73.

Babovic, V., Kanizares, R., Jenson, H. R. & Klinting, A. 2001 Neural networks as routine for error updating of numerical models. *J. Hydraulic Engng. ASCE* **127** (3), 181–193.

Becker, A. & Serban, P. 1990 Hydrological models for water resources system design and operation. Operational Hydrology Report 34, WMO No. 740, Geneva, Switzerland.

Bell, V. A. & Moore, R. J. 1998 A grid-based distributed flood forecasting model for use with weather radar data: Part 2. Case studies. *Hydrol. Earth System. Sci.* **2**, 283–298.

Bowden, G. J., Dandy, G. C. & Maier, H. R. 2003 Data transformation for neural network models in water resources applications. *J. Hydroinformat.* **5** (4), 245–258.

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

Brath, A., Montanari, A. & Toth, E. 2002 Neural networks and non-parametric methods for improving real time flood forecasting through conceptual hydrological models. *Hydrol. Earth System. Sci.* **6** (4), 627–640.

Caudill, M. & Butler, C. 1992 *Understanding Neural Networks*. Basic Networks 1. MIT Press, Cambridge, MA, USA.

Chau, K. W., Wu, C. L. & Li, Y. S. 2005 Comparison of several flood forecasting models in Yangtze River. *J. Hydrol. Eng.* **10** (6), 485–491.

Galeati, G. 1990 A comparison of parametric and non-parametric methods for runoff forecasting. *Hydrol. Sci. J.* **35** (1), 79–94.

Goswami, M. & O'Connor, K. M. 2010 A 'monster' that made the SMAR conceptual model 'right for the wrong reasons'. *Hydrolog. Sci. J.* **55** (6), 913–927.

Goswami, M., O'Connor, K. M., Bhattarai, K. P. & Shamseldin, A. Y. 2005 Assessing the performance of eight real-time updating models and procedures for the Brosna River. *Hydrol. Earth System. Sci.* **9** (4), 394–411.

Hornik, K., Stinchcombe, M. & White, H. 1989 Multilayer feed-forward networks are universal approximators. *Neural Netw.* **2** (5), 359–366.

Hsu, K.-L., Gupta, V. & Sorooshian, S. 1995 Artificial neural network modeling of the rainfall-runoff process. *Water Resour. Res.* **31**, 2517–2530.

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

Jain, A. & Kumar, A. M. 2007 Hybrid neural network models for hydrologic time series forecasting. *Appl. Soft. Comput.* **7**, 585–592.

Kang, P. & Cho, S. 2008 Locally linear reconstruction for instance-based learning. *Pattern Recognition* **41**, 3507–3518.

Karlsson, M. & Yakowitz, S. 1987 Nearest neighbor methods for non parametric rainfall runoff forecasting. *Water Resour. Res.* **23** (7), 1308–1330.

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

Khu, S. T., Keedwell, E. C. & Pollard, O. 2004 An evolutionary-based real-time updating technique for an operational rainfall-runoff forecasting model. In: *Complexity and Integrated Resources Management, Trans. 2nd Biennial Meeting of the Int. Env. Modelling and Software Soc* (C. Pahl-Wostl, S. Schmidt, A. E. Rizzoli & A. J. Jakeman, eds). iEMSs, Manno, Switzerland, 1, pp. 141–146.

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

Lin, J. Y., Cheng, C. T. & Chau, K. W. 2006 Using support vector machines for long-term discharge prediction. *Hydrolog. Sci. J.* **51** (4), 599–612.

Lundberg, A. 1982 Combination of a conceptual model and an autoregressive error model for improving short time forecasting. *Nordic Hydrol.* **13**, 233–246.

Minns, A. W. & Hall, M. J. 1996 Artificial neural networks as rainfall-runoff models. *Hydrol. Sci. J.* **41** (3), 399–417.

Moller, M. F. 1993 A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* **6**, 523–533.

Nash, J. E. & Sutcliffe, J. V. 1970 River flow forecasting through conceptual models; part I – a discussion of principles. *J. Hydrol.* **10**, 282–290.

Rajurkar, M. P., Kothyari, U. C. & Chaube, U. C. 2004 Modeling of the daily rainfall-runoff relationship with artificial neural network. *J. Hydrol.* **285** (1–4), 96–113.

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

Ruprecht, D. & Muller, H. 1994 A framework for generalized scattered data interpolation. Technical Report no. 539, Universitat Dortmund, Fachbereich Informatik, D-44221 Dortmund, Germany.

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

Shamseldin, A. Y. & O'Connor, K. M. 1996 A nearest neighbor linear perturbation model for river flow forecasting. *J. Hydrol.* **179**, 353–375.

Shamseldin, A. Y. & O'Connor, K. M. 1999 A real-time combination method for the outputs of different rainfall-runoff models. *Hydrol. Sci. J.* **44** (6), 895–912.

Shamseldin, A. Y. & O'Connor, K. M. 2001 A non-linear neural network technique for updating river flow forecasts. *Hydrol. Earth System. Sci.* **5** (4), 577–597.

Solomatine, D. P., Maskey, M. & Shrestha, D. L. 2008 Instance-based learning compared to other data-driven methods in hydrological forecasting. *Hydrol. Process.* **22**, 275–287.

Toth, E., Brath, A. & Montanari, A. 1999 Real-time flood forecasting via combined use of conceptual and stochastic models. *Phys. Chem. Earth* **24** (7), 793–798.

Wand, M. P. & Schucany, W. R. 1990 Gaussian-based kernels. *Can. J. Stat.* **18** (3), 197–204.

World Meteorological Organization (WMO) 1992 Simulated real time intercomparison of hydrological models. Operational Hydrology Report No. 38, Geneva, Switzerland.

Wu, C. L., Chau, K. W. & Li, Y. S. 2009 Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. *Water Resour. Res.* **45** (8), 1–23.

Xiong, L. & O'Connor, K. M. 2002 Comparison of four updating models for real-time river flow forecasting. *Hydrol. Sci. J.* **47** (4), 631–629.

Xiong, L., O'Connor, K. M. & Guo, S. 2004 Comparison of three updating schemes using artificial neural network in flow forecasting. *Hydrol. Earth Syst. Sci.* **8** (2), 247–255.

Yakowitz, S. 1987 Nearest-neighbor methods for time series analysis. *J. Time Ser. Analysis* **8** (2), 235–247.

Zhang, B. & Govindaraju, S. 2000 Prediction of watershed runoff using Bayesian concepts and modular neural networks. *Water Resour. Res.* **36** (3), 753–762.

First received 7 June 2012; accepted in revised form 19 September 2013. Available online 5 November 2013