Artificial neural network model for river flow forecasting in a developing country

Asaad Y. Shamseldin

ABSTRACT

The present paper deals with exploring the use of Artificial Neural Networks (ANN) for forecasting the Blue Nile river flows in Sudan. Four ANN rainfall–runoff models based on the structure of the well-known multi-layer perceptron are developed. These models use the rainfall index as a common external input, with the rainfall index being a weighted sum of the recent and current rainfall. These models differ in terms of the additional external inputs being used by the model. The additional inputs are basically the seasonal expectations of both the rainfall index and the observed discharge. The results show that the model, which uses both the seasonal expectation of the observed discharge and the rainfall index as additional inputs, has the best performance. The estimated discharges of this model are further updated using a non-linear Auto-Regressive Exogenous-input model (NARXM)-ANN river flow forecasting output-updating procedure. In this way, a real-time river flow forecasting model is developed. The results show that the forecast updating has significantly enhanced the quality of the discharge forecasts. The results also indicate that the ANN has considerable potential to be used for river flow forecasting in developing countries.

Key words | Blue Nile, data-driven modelling, floods, neural network, river flow forecasting, Sudan

Asaad Y. Shamseldin
Department of Civil and Environmental
    Engineering,
University of Auckland,
Private Bag 92019, Auckland,
New Zealand
E-mail: *a.shamseldin@auckland.ac.nz*

INTRODUCTION

The current paper presents a case study concerned with the use of a hydroinformatics data-driven modelling tool (artificial neural network (ANN)) in river flow forecasting in a developing country (Sudan). This paper demonstrates how the hydroinformatics modelling technology can be applied in solving challenging water resources problems in developing countries. However, hydroinformatics is not only about modelling technology, albeit being a key component of hydroinformatics. It is a socio-technology occupying "the middle ground between water related physical and natural sciences, information and communication technology and the social context" (Price & Jemberie 2006). A review of recent advances as well as *emergent directions for future developments* in hydroinformatics can be found in Coulibaly *et al.* (2009).

River flow forecasting models are usually used as components in flood forecasting systems which provide flood warnings. Floods are among the natural disasters which are most damaging in terms of economic and life losses. They account for about one-third of the damages and one-half of the fatalities attributed to natural disasters (Berz 2000). Between 1973 and 1997, UNESCO (2005) stated that around *66 million people* suffered flood damage worldwide. In the case of poor and densely populated countries, Berz (2000) noted that floods continue to cause the largest numbers of deaths relative to other natural disasters. In developing countries, floods can have significant impacts on the economy. In general, natural disasters have disproportionate impacts on the Gross Domestic Product (GDP) of developing countries as compared with that of developed countries (UN 2009).

Flood mitigation can be achieved by adopting structural and non-structural measures. Implementing large-scale structural engineering solutions such as river training and construction of dams and levees can prevent flooding. Implementing these structural engineering solutions is usually beyond the means and abilities of many developing countries. Furthermore, in many countries, the construction of large scale embankments is regarded as *economically infeasible in addition to being environmentally unsound*. Reduction of flood losses can also be accomplished by adopting several non-structural measures such as regulating land use in the flood plains, improving the design of buildings, river flow forecasting systems, public education, emergency planning and flood insurance schemes. The cost of some of these non-structural measures can be cheaper relative to the large-scale structural engineering solutions and hence they have a lot of potential use in developing countries. Although flood insurance protection schemes would help individuals and communities to recover from the devastating effects of flooding, such schemes are virtually nonexistent in many developing countries. IIASA (2009) reported that "In the developing world, less than 2% of the costs of catastrophes are absorbed by any form of insurance, compared to 50% of the costs of catastrophes covered by insurance in the United States".

River flow forecasting is usually regarded a valuable and cost-effective flood mitigation measure either alone or in combination with other structural or non-structural flood mitigation measures. River flow forecasting is also considered as one of the key components of Flood, Forecasting, Warning and Response (FFWR) systems. These systems can be viewed as complex socio-technical systems requiring the tuning of their technical dimensions to suit the requirements of at-risk population (customers/users) (Khatibi & Cluckie 2006). The key components of a typical FFWR system are: (i) detection of the likelihood of floods forming using hydro-meterological data, (ii) river flow forecasting using observed hydro-meteorological data, (iii) warning, (iv) dissemination and (v) response (cf. Khatibi & Cluckie 2006; Todini 2006). While the river flow forecasting is important, the "social performance" of the flood warning dissemination and communication channels is also important. The social performance (i.e. effectiveness) of the FFWR systems involves complex relationships between

"the technologies selected, the social characteristics of the warning recipients and local barriers to warnings i.e. the local circumstances of the community/individual" (DEFRA/EA 2005). WMO (2005) provided an excellent summary about the common problems facing early warning systems in developing and least developed countries These problems include: "(i) linkages between the national meteorological and hydrological services, the emergency management authorities and the media, (ii) alert mechanisms from national authorities to local level, (iii) availability of national to local emergency policy and preparedness plans and (iv) education and public outreach programmes and other capacities that would enable the public to know how to respond to warnings".

The focus of the present paper is on the use of non-structural flood mitigation measures, namely real-time river flow forecasting in the context of developing countries. The present paper deals with developing a river flow forecasting model (i.e. a rainfall–runoff model with an add-on forecast updating procedure) for the Blue Nile River in Sudan. The model developed in this paper can be used as the basis of issuing flood warnings, thereby helping in reducing flood damages. The Blue Nile is a major tributary of the Nile River contributing about 60% of its total annual flow. Over the last three decades floods arising from the Nile River have become more *frequent and* more *extreme*. Major flood events occurred in 1988, 1994, 1996, 1998, 1999, 2001 and 2003. These more frequent and more extreme floods may be attributed to El Niño (Eltahir 1996; Said 1999) or may be the result of global warming. These floods cause considerable damage to communities living along the river. For example, the 1988 flood caused the death of 58 persons and left nearly 2 million people homeless (Chicago Sun-Times 1988). It also resulted in the outbreak of diseases such as malaria and diarrhoea with vulnerable groups such as children and elderly being at high risk.

There are three rainfall–runoff model categories which can be used as components in river flow forecasting systems (Wheater *et al.* 1993; Senbeta *et al.* 1999; Shamseldin 2006): (i) metric models, which are also known as empirical black-box models, (ii) conceptual models, which are also called explicit soil moisture accounting models, and (iii) mechanistic models, which are also referred to as physically based models. In the context of developing countries and many of

the developed countries, the first two categories are usually used in operational river flow forecasting systems. At present, models of the third category remain mainly as heuristic research tools due to their intense data requirements. It is often difficult to fulfil these requirements due to the paucity of the data and the associated costs needed to obtain the data.

Artificial Neural Network (ANN) models, which are the focus of the present paper, belong to the first model category. They are inspired by research into the biological neural networks. However, in pure systems terminology, they are basically non-linear data-driven models which provide powerful solutions to many complex modelling problems. Similar to models of the first category they require synchronous input–output data for their calibration. Many studies have demonstrated that the ANN models are very successful in simulating river flows (e.g. Shamseldin 1997; Coulibaly *et al.* 2000; Chang & Chen 2001; Dibike & Solomatine 2001; Shamseldin *et al.* 2002, 2007; Rajurkar *et al.* 2004; Goswami *et al.* 2005; Dawson *et al.* 2006; Abrahart *et al.* 2007; Boucher *et al.* 2009; Fernando & Shamseldin 2009; Pramanik & Panda 2009) and hence worthy of investigation with regard to river flow forecasting on the Blue Nile River. There is a limited number of studies dealing with river flow forecasting on the Blue Nile. Examples of these studies can be found in Grijsen *et al.* (1992), Elmahi & O'Connor (1995), Shamseldin *et al.* (1999), Shamseldin & O'Connor (2003) and Antar *et al.* (2005). Thus, this paper will shed more light on potential data-driven models which can be used for flood forecasting on the Blue Nile.

The ANN river flow forecasting models have many features which make them attractive for use in developing countries. These features include:

- Rapid development: ANN models are easy to develop, as they do not require very detailed knowledge about the physical functioning of the catchment. The key to their success is the determination of the appropriate external inputs to the model.
- Rapid execution time: once the ANN models are calibrated they are fast to run, requiring very little execution time on a modest PC.
- Parsimony in terms of their data requirements compared to the other traditional models. In many of the developing countries, the hydrological data is very sparse.

- Availability of open source codes either free or at a very cheap rate. For example, the source code of the Stuttgart Neural Network Simulator is freely available on the Web (see http://www-ra.informatik.uni-tuebingen.de/SNNS/). The licence of the Trajan neural network software package costs around $3000 and it has facilities for code generation (see www.trajan-software.demon.co.uk). The availability of open source code would help the in-house development of river forecasting software and hence reducing the development cost. The *in-house* development would also help in empowering local institutions and strengthening their technical capacity.

It is worth noting the above features are not exclusive to ANN models but are also shared by other metric and conceptual models. Thus, there are other rival models to the ANN models which are also appropriate for river flow forecasting in developing countries.

When implementing a river flow forecasting system in developing countries careful consideration should be given to the sustainability of its operation. The whole life cost analysis of the system would help in this regard. The technology adopted in the forecasting system should be appropriate to the prevailing conditions in developing countries to ensure sustainability. What you will find in many developing countries is that, in a response to major flood events, off-the-shelf river flow forecasting systems are purchased from the developed countries and put into operation, usually with foreign technical and financial help. Some of these employed systems are technologically advanced requiring the purchase and acquisition of data as well as technical support from third parties. With the on-going process of "*brain drain*" of skilled workers from developing (poor) countries to the developed (rich) countries and the lack of adequate funds for purchasing the relevant data and technical support, the sustainability of such systems is at very high risk.

This paper first develops four ANN rainfall–runoff models operating in the simulation design mode without any feedback information from the most recently observed discharge data. These models use different external inputs reflecting the recent rainfall history and the seasonality of the catchment. Having identified an appropriate ANN rainfall–runoff model, the simulation mode discharges of

this model are updated using the Non-linear Auto-Regressive Exogenous-Input Model (NARXM-ANN) output updating procedure developed by Shamseldin & O'Connor (2001). The operation of this procedure is based on the real-time external modification of simulation mode discharges of the ANN rainfall–runoff model without interfering with its operation. In this way, a real-time river flow forecasting model is developed providing river flow forecasts for different lead-times. This real-time forecasting model is basically the substantive ANN simulation model together with the add-on NARXM-ANN output updating procedure. The updating procedure enables the use of feedback information in the form of the most recently observed discharge data in order to enhance the discharge estimates of the substantive model, which differ from the observed discharges. In broad terms, the addition of an updating component to the simulation mode results of the rainfall–runoff model would significantly improve the reliability of the real-time river flow forecasts. However, as the lead-time increases, the reliability of the real-time river flow forecasts would be more dependent on the reliability of the design mode simulated river flows rather than the flexibility of the updating procedure. Thus, the approach adopted in this study for the development of the real-time forecasting model enables the evaluation of performance of ANN operating as a rainfall–runoff simulation model and also an efficient forecast updating procedure. This would enhance the chance for improving the river flow forecast reliability for short as well as when the lead-time increases.

The present paper is organized as follows; firstly a brief description of the study area is given. Secondly, the ANN rainfall–runoff models used in this study, the method used for determining their inputs and their calibration/training are described. Thirdly, the NARXM-ANN output updating procedure is described. Fourthly, the procedure used in evaluating model performance is described. Finally, the results and the conclusions of the study are discussed.

STUDY AREA

The Blue Nile is one of the main tributaries of the River Nile contributing approximately around 60% of its annual flow—the annual flow being 84 km^3. The Blue Nile and its

tributaries arise from the Ethiopian plateau in East Africa with an elevation range between 2,000–3,000 m. The Blue Nile basin has an area of $324,530 \text{ km}^2$, which covers most of Ethiopia west of longitude 40°E and between latitudes 9° and 12°N (Shahin 1985, p. 42). The climate in the Ethiopian plateau is regarded as temperate despite being situated in a tropical region (USBOR 1964, p. 29). In the upper Blue Nile in Ethiopia, Conway (1997) noted that annual mean potential evapotranspiration and rainfall range from 1,800 mm to 1,200 mm and 924 mm to 1,845 mm, respectively.

The Blue Nile River is considered as a trans-boundary flowing through Ethiopia and Sudan. It starts at Lake Tana in Ethiopia and flows for 900 km to the Sudanese–Ethiopian border. In Sudan, it joins the White Nile (one of the main tributaries of the Nile River) at Khartoum (the capital of Sudan) to form the Nile river. Figure 1 shows the Blue Nile catchment upstream at Eldiem near the Sudanese–Ethiopian border.
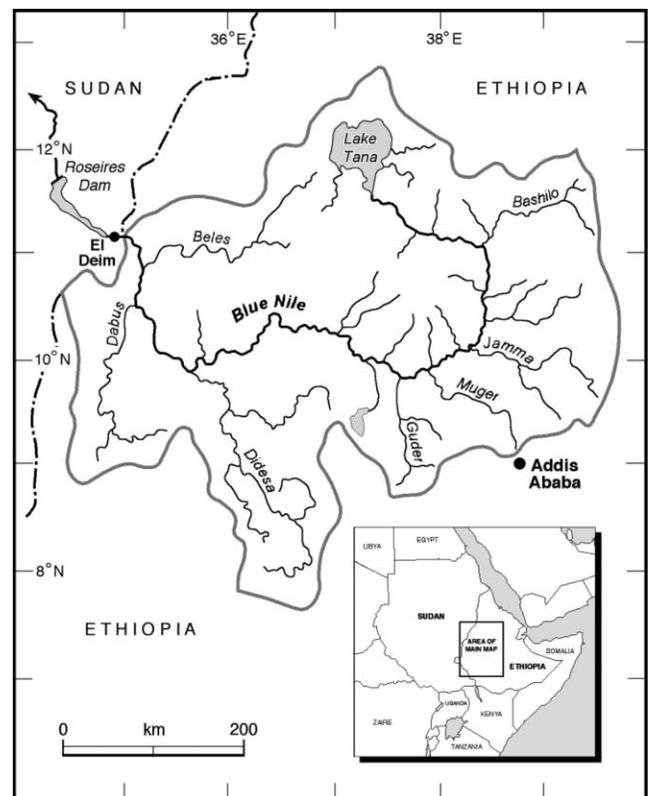

Figure 1 | The Blue Nile catchment upstream of Eldeim.

The Blue Nile is a very seasonal river with 80% of its annual rainfall occurring during the months of June–September with the peak flow occurring in late August (see Figure 2). Flood peaks usually occur in late August and the maximum daily flow can reach a value of $10,000 \, m^3/s$. The mean annual flow of the Blue Nile River at Eldeim is about $50 \, km^3$; with the annual flow varying between $70 \, km^3$ during flood years and $30 \, km^3$ during drought years. The lag time of this catchment (i.e. the time difference between the peak of the rainfall and the peak of the discharge hydrograph) varies over the range of 10–20 days, depending on the catchment wetness.

In this study, four years of daily flow values of the Blue Nile River measured at Eldeim near the Sudanese–Ethiopian border and the areally averaged rainfall data for the period 1992–1995 are used. The first three years are used for model calibration/training while the remaining year is used for model verification/validation. The flow data contains a mixture of annual flood peaks of different magnitudes. The areal average rainfall was obtained using the data of six rainfall stations (El Sebai 1998). This is a very coarse rainfall data resolution. However, remote sensing technology offers tremendous opportunities for improving the spatial and temporal resolution of the rainfall data in this catchment (cf. El Sebai 1998; Antar et al. 2005).

THE MULTI-LAYER PERCEPTRON (MLP)

The ANN rainfall models and the output updating procedure developed in this study are based on the structure of the multi-layer perceptron (MLP). It is one of the most popular neural network types which has been extensively used in hydrological modelling (cf. Maier & Dandy 2000;
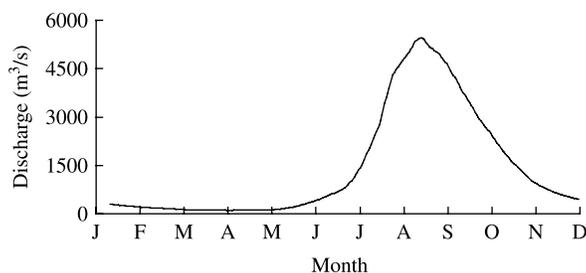

Figure 2 │ Average annual discharge hydrograph of the Blue Nile River at Eldeim.
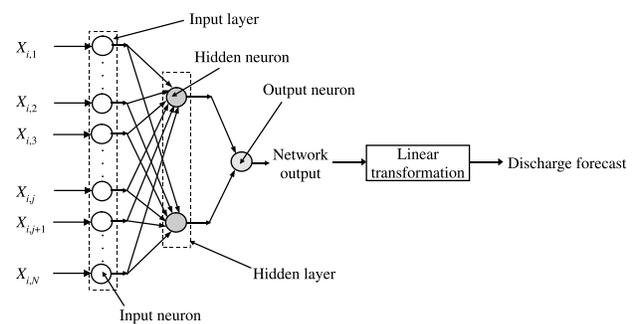

Figure 3 │ Schematic diagram of the Multi-Layer Perceptron.

Dawson & Wilby 2001). The MLP is simply a nonlinear input–output model. The structure of the MLP consists of a network of interconnected neurons (computational units) linked together by connection pathways. The neurons are arranged in a cascade of layers, each layer performing a unique function in the overall operation of the network (see Figure 3).

The simplest form of the MLP consists of three layers; input layer, hidden layer and output layer. The more complex forms have more than one hidden layer which are very rarely used in hydrological applications as the use of more than one is hardly ever beneficial (Masters 1993). Figure 3 shows a schematic diagram of the simple form of the MLP. As shown in the figure, adjacent layers are connected by links governing the flow of information. In the MLP, the information flows only in the feedforward direction without any feedback links.

The first layer in the MLP is the input layer which receives the external input vector $\mathbf{X}_i$ to the network at each discrete time period i:

$$\mathbf{X}_i = (X_{i,1}, X_{i,2}, \dots X_{i,j}, \dots, X_{i,N})^T \qquad (1)$$

where $X_{i,j}$ is the jth external input for the i-th time period, N is the number of external inputs and where T denotes the vector transpose. As each element of the external input vector is allocated to one of the input neurons, the number of neurons in this layer is equal to the number of external inputs N. The input neurons transmit the external input array into the network without any modifications.

The hidden layer is an intermediate layer between the input and the output layer. It is called hidden because it has no direct connections to the external inputs and outputs.

The hidden layer enhances the capability of the network to deal robustly and efficiently with inherently complex non-linear modelling problems. The number of neurons (M) in this layer is usually unknown *a priori* and it is estimated by a trial-and-error procedure. Each neuron in the hidden layer receives the same input vector of N elements from the neurons of the input layer, as defined by Equation (1). The input–output transformation in the k-th hidden neuron is achieved by a mathematical nonlinear transfer function which can be expressed as

$$Y_{i,k} = f(y_{i,k}) = f\left(\sum_{j=1}^{N} w_{j,k}Q_{i,j} + w_{o,k}\right) \tag{2}$$

where $Y_{i,k}$ is the output of the kth hidden neuron for the ith time period, f( ) is the nonlinear transfer function, $w_{j,k}$ is the connection weight, which is assigned to the connection pathway between the kth hidden neuron and the jth neuron in the previous (input) layer and $w_{o,k}$ is the threshold value of the kth hidden neuron.

The output layer is the last layer in the MLP and its main role is to produce the final network output. The number of neurons in the output layer is equal to the number of elements in the external output array of the network, this number being unity in the single-output case of the present study. This single-output neuron receives an input array $\mathbf{Y}_i = (Y_{i,1}, Y_{i,2}, \ldots, Y_{i,M})^T$ from the previous hidden layer, the elements of which are the M outputs of the hidden neurons. The input–output transformation of the output neuron is similar to that of the hidden neuron. The final network output Z_i for the ith time period is given by

$$Z_i = f\left(\sum_{j=1}^{M} W_j Y_{i,j} + W_o\right) \tag{3}$$

where W_j is the connection weight between the jth neuron in the hidden layer and the single output neuron and W_o is the neuron threshold value.

The neuron threshold values and the connection weights between adjacent layers are effectively the parameters of the network which are to be estimated by training (i.e. calibration).

The same nonlinear mathematical transformation function is generally used for all of the hidden and output neurons. The most widely used nonlinear transfer function in neural network applications, which is also used in this study, is the sigmoid function (Blum 1992, p. 39). This function has an S shape and its range varies between 0 and 1.

As the actual external outputs of the network are generally outside the bounded range of the neuron transfer function, then it is necessary to rescale or transform the actual (i.e. observed) external outputs in such a way as to be within the bounded output range in order to facilitate the calibration of the MLP and to make direct comparisons between the network estimated outputs and the external rescaled actual outputs. In the present work, the observed discharge series Q_i is rescaled according to the following linear transformation:

$$Qs_i = 0.1 + 0.75\left(\frac{Q_i}{Q_{\max}}\right) \tag{4}$$

where Qs_i is the rescaled observed discharge series and $Q_{\max}$ is the maximum observed discharge in the calibration period. In this case, the effective bounded range of the rescaled discharge series Qs_i in the calibration period varies between 0.1 and 0.85. The use of this effective rescaling range is vital in order to facilitate the calibration process, in particular when derivative-based optimisation techniques are used for calibration of the network. This is important to ensure that the MLP can forecast discharge values that are greater than those occurring in the calibration period.

DETERMINATION OF THE EXTERNAL INPUTS TO THE ANN RAINFALL–RUNOFF MODEL

The determination of appropriate external input types to the ANN model is one of the fundamental keys which enables the MLP to provide effective solutions to complex modelling problems. Similar to other hydrological studies, the approach used in this study for determining the external input information is based on the utilisation of prior hydrological knowledge about the catchment and the use of a trial-and-error procedure (Shamseldin 1997; Thirumalaiah & Deo 2000; Bowden *et al.* 2005; Parent *et al.* 2008).

The first input information type used in this study is the rainfall in the form of the most recent rainfall values over the memory length (m) of the catchment. These values are

basically the elements of the rainfall vector $\mathbf{Y}_i$ defined by

$$\mathbf{Y}_i = (R_i, R_{i-1}, R_{i-2}, \dots, R_{i-m+1})^T \tag{5}$$

where R_i is the measured rainfall for the ith time period. Each of the elements can be assigned to one of the neuron in the input layer. However, if the memory length of the catchment is large then this will require a similar large number of input neurons. The use of a large number of inputs will result in a complex non-parsimonious network with a large number of parameters to be estimated by calibration/training and such estimation may not be an easy task.

In the case of the Blue Nile basin the memory length is large and hence there is a need for reducing the number of inputs to the network to obtain a parsimonious ANN model. In this study, the input reduction is achieved through the pre-processing of the external inputs using a linear transformation. As a result of the pre-processing a rainfall index RI_i is considered as the external input factor to the network instead of the m rainfall values. The rainfall index is simply a weighted sum of the m rainfall values obtained in the same manner as the simple linear model. The rainfall index RI_i can be expressed mathematically by the following equation:

$$RI_i = \sum_{j=i-m+1}^{i} R_j h_{i-j+1} \tag{6}$$

where h_j is the jth weight which in the context of the simple linear model can be viewed as the discrete pulse response ordinate. The rainfall index (RI) reflects the recent history of rainfall incidents and it can also be viewed as a crude index for soil moisture conditions/wetness of the catchment. The estimation of the rainfall index (RI) requires a knowledge of the numerical values of the ordinates of the pulse response function. In this study, these ordinates are obtained in the parametric form, in a similar fashion to those of the simple linear model using the two-parameter gamma distribution model proposed by Nash (1957). The impulse response of the gamma distribution model, $h(t)$, is given by

$$h(t) = \frac{1}{K\Gamma(n)} \left(\frac{t}{K}\right)^{n-1} e^{-t/K} \tag{7}$$

where $\Gamma(n)$ is the gamma function of the variable n. In the previous equation, n and K are the parameters of the gamma distribution model of the h_j series. However, in real applications, n and the product nK are usually considered to be the parameters of the gamma distribution model. This would facilitate the calibration process as n and nK are less dependent on each other than n and K (cf. Kachroo & Liang 1992).

Figure 2 shows that the Blue Nile basin has a very remarkable seasonality and therefore incorporation of input information about this seasonality may further improve the performance of the ANN model. In the present study, the seasonality is incorporated in the ANN model by using the seasonal expectation of the discharge $_sQ_i$ and the corresponding seasonal expectation rainfall index $_sRI_i$ as external inputs to the MLP.

According to the forgoing discussion, the MLP can have a maximum of three external inputs, namely, the rainfall index, the seasonal expectation rainfall index $_sRI_i$ and the seasonal expectation of the discharge $_sQ_i$. In the present study, four different ANN models are developed which use a combination of these three external inputs. These four different models are referred to in this paper as ANN1 ANN2, ANN3 and ANN4, respectively. Table 1 provides a summary of the external inputs used by each of these four models. In this study, the ANN1 model, which only utilises the rainfall index as input information, is regarded as a benchmark model against which the performance of the other complex ANN models which use more input information can be compared. Thus, it is not the intention to use the ANN1 model as a serious rainfall–runoff model. Furthermore, the development of these four models would help in determining the effects of input information on model performance.

Table 1 │ Summary of the external inputs used by the different ANN models

Model	External input data		
	RI	$_sRI_i$	$_sQ_i$
ANN1	√		
ANN2	√	√	
ANN3	√		√
ANN4	√	√	√

CALIBRATION OF THE ANN RAINFALL–RUNOFF MODELS

The operation of the four ANN models developed in this study depends on the prior estimation of the numerical values of the external inputs to the network. As shown in Table 1 each of the four models uses the rainfall index RI_i as a common input. The calculation of the rainfall index requires a knowledge of the parameter values of the gamma distribution model (n and nK). Once the corresponding external inputs to a particular ANN model are calculated, its operation with a given number of hidden neurons requires the estimation of the parameter values of the network. Hence, the calibration of each of the four ANN models involves the determinations of the parameter values of the gamma distribution model (n and nK) as well as the network parameter values. The procedure used to calibrate the ANN models is based on an iterative calibration procedure which was developed by Shamseldin (1997). The steps involved in this iterative procedure are:

1. Choose suitable initial estimates for the values of the parameters of the gamma distribution model in order to calculate the external inputs to the ANN model (i.e. the corresponding combination of the rainfall index RI_i, the seasonal expectation of the rainfall index $_sRI_i$ and the seasonal expectation of the discharge $_sQ_i$).

2. Choose suitable initial values of the parameters of the network (i.e. the connection weights and the neuron threshold values for the hidden and output neurons) and, using the conjugate gradient algorithm (see Press et al. 1989), find estimates of the parameter values of the network by minimising the least-squares objective function which is the sum of the squares of the differences between the ANN outputs and the rescaled observed discharges.

3. Using the ANN model parameter estimates of step 2, use the simplex method (Nelder & Mead 1965) to refine the initial estimated values of the parameters of the gamma distribution model by minimising the same least-squares objective function.

4. Using the refined estimated values of n and nK obtained in step 3 as new initial values, return to step 1.

5. Repeat steps 1–4 until there is no improvement in the overall performance of the neural network as measured in terms of the least-squares objective function.

THE NON-LINEAR AUTO-REGRESSIVE EXOGENOUS-INPUT MODEL (NARXM)-ARTIFICIAL NEURAL NETWORK (ANN) UPDATING PROCEDURE

The essence of the NARXM-ANN updating procedure is that the simulation mode discharges of the substantive rainfall–runoff model are used as exogenous inputs to the NARXM-ANN procedure which is also based on the MLP. The structure of the MLP used in the NARXM-ANN procedure is similar to that used when developing the four ANN rainfall–runoff models. In the case of the NARXM-ANN model, the MLP has one hidden layer and the sigmoid function is used as a transfer function for the hidden and the output neurons. The linear scaling function given by Equation (4) is also used for rescaling the external observed discharges to facilitate the calibration of the MLP and the comparisons between the observed and the estimated discharges.

In the NARXM-ANN updating procedure, the 1-day lead-time updated discharge forecast $\hat{\hat{Q}}_{i+1|i}$ at time i is given by

$$\hat{\hat{Q}}_{i+1|i} = G(Q_i, Q_{i-1}, \ldots\ldots\ldots Q_{i+1-p}, \hat{Q}_{i+1}, \hat{Q}_i, \ldots\ldots,$$
$$\hat{Q}_{i+1-q}) + e_{i+1} \tag{8}$$

where G denotes a nonlinear functional relation, $\hat{Q}_{i+1}$ is the simulation mode discharges of the rainfall–runoff model at time $i + 1$, p and q are the orders of the auto-regressive and the exogenous input parts of the NARXM-ANN procedure and e_{i+1} is the residual error of the updated discharge forecast. Thus, for the 1-day lead time forecast the external input array $\mathbf{X}_i$ to the NARXM-ANN procedure is basically $\mathbf{X}_i = (Q_i, Q_{i-1}, \ldots\ldots, Q_{i-p}, \hat{Q}_{i+1}, \hat{Q}_i, \ldots\ldots, \hat{Q}_{i+1-q})^T$ consisting of the simulation mode discharges (the exogenous inputs) and the recently observed discharges.

In operational real-time river forecasting, the NARXM-ANN updating output procedure can be used on-line to provide updated discharge estimates for the required

forecast lead-time (i.e. for the period beyond the current time for the end of which the forecast is required). This would require a knowledge of the values of the observed discharges and the non-updated discharges of the substantive ANN rainfall–runoff model over the lead-time of the forecast. The non-updated discharges over the lead-time are obtained from the substantive rainfall–runoff model, using forecasts of the meteorological input information. However, in the present work similar to other heuristic research works on river flow forecasting (e.g. Kachroo & Liang 1992; WMO 1992), the scenario of *perfect input foresight over the forecast lead-time* is adopted in estimating the non-updated discharges of the substantive rainfall–runoff model. This choice of input scenario effectively eliminates the effects of errors in the meteorological forecasts, so that the performance of the real-time river flow forecasting model (i.e. the rainfall–runoff model together with the updating procedure) can be objectively evaluated. If the real-time forecasting model is tested using forecasts of the input variables over the lead-time and the model fails, then it may be quite difficult to attribute this failure to the model itself or to the poor specification of the input variables over the lead time. Thus, adopting the *perfect input foresight* scenario is a first step towards building reliable real-time forecasting models.

As the values of the observed discharges over the forecast lead-time not yet available, estimates of these observed discharge values are obtained by the recursive applications of Equation (8). Accordingly, the updated discharge forecast, $\hat{\hat{Q}}_{i+l|i}$, at time i, for a lead-time $l \geq 1$, is given by

$$\hat{\hat{Q}}^{NARXM}_{i+l|i} = G\big(\hat{\hat{Q}}_{i+l-1|i}, \hat{\hat{Q}}_{i+l-2|i}, \ldots, \hat{\hat{Q}}_{i+1|i}, Q_i, \ldots, Q_{i+l-p},$$

$$\hat{Q}_{i+l|i}, \hat{Q}_{i+l-1|i}, \ldots, \hat{Q}_{i+1|i}, \hat{Q}_i, \hat{Q}_{i-1}, \ldots, \hat{Q}_{i+l-q}\big) \quad (9)$$

The calibration of the NARXM-ANN updating procedure involves the estimation of the number of hidden neurons, the corresponding parameters of the MLP as well as the autoregressive (p) and the moving average (q) orders. In this study, the number of hidden neurons and the values of these orders are estimated by trial and error. For a fixed number of hidden neurons and given values of p and q the

NARXM-ANN model is calibrated using the conjugate gradient method.

EVALUATION OF MODEL PERFORMANCE

The performances of the models developed in this paper are evaluated using the well-known R^2 model efficiency criterion suggested by Nash & Sutcliffe (1970). This criterion is closely linked to the least-squares objective function being expressed as the sum of the squares of the differences F between the model estimated $\tilde{Q}_i$ and Q_i observed discharges. The R^2 model efficiency criterion can be mathematically expressed as

$$R^2 = \frac{F_o - F}{F_o} \quad (10)$$

where $F = \sum(Q_i - \tilde{Q}_i)^2$ and F_o is the initial sum squares of differences given by

$$F_o = \sum(Q_i - \bar{Q})^2 \quad (11)$$

and $\bar{Q}$ is the average of the observed discharge of the chosen calibration period. The initial sum of squares of errors F_o can be viewed as a measure of performance of a primitive model producing a constant estimated discharge equal to the average of the observed discharge in the calibration period. Thus, the R^2 criterion is, in essence, a global measure of the performance of the substantive model relative to that of the primitive model. A free web resource to calculate the R^2 values can be found at www.hydrotest.org.uk and further details about this resource are described in Dawson *et al.* (2007, 2009).

RESULTS

ANN rainfall–runoff models

The operation of the four ANN models, which are based on the structure of the MLP, requires the specification of the number of neurons in the hidden layer. This number is usually unknown *a priori*. In the present study, the optimum number is estimated by a trial-and-error procedure in which the ANN model is calibrated/trained in

succession with an increasing number of hidden neurons using the calibration procedure described earlier in this paper. The performance of the model is monitored in each trial using the R^2 model efficiency criterion. The optimum number of hidden neurons is that beyond which any subsequent increase in the number of hidden neurons does not result in significant improvements in model performance.

The operation of some of these models requires the estimation of the seasonal expectations of the discharge and the rainfall index time series. These seasonal expectations are found by calculating the average daily values for each day in the year using the data of the calibration period. The resulting series is then smoothed globally to reduce sampling fluctuations by the *discrete Fourier series* using the first four harmonics. Further details about this smoothing procedure can be found in Salas *et al.* (1980).

Figure 4 shows a plot of the R^2 values of the four ANN models in which two and three hidden neurons are used. Inspection of the figure shows that, in both the calibration and verification periods, there are no considerable improvements in the overall performances of the four
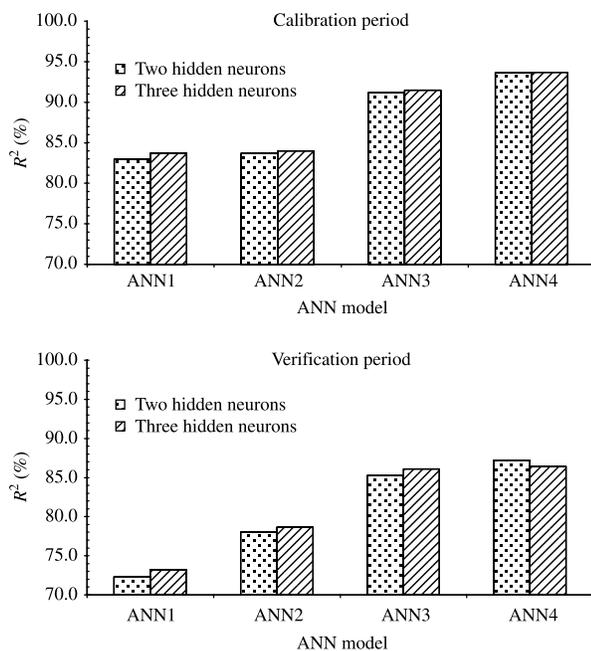
ANN models by increasing the number of hidden neurons beyond two. This can be taken as a strong indication that the data used by each model does support more complex models with several hidden neurons. Hence, the optimum number of the hidden neurons for these four ANN models is taken as two.

Table 2 shows the Nash–Sutcliffe R^2(%) for the four ANN models developed in this study. Examination of the table shows the R^2 efficiency values in the calibration period vary between 82.98% and 93.64% while those of the verification period vary between 72.32% and 87.21%. Not surprisingly, the ANN1 model which uses the least amount of external input information (i.e. the rainfall index only) has the worst model performance results (i.e. lowest R^2 values) among the four models. Likewise, the ANN4 model which uses the maximum amount of external input information (i.e. the rainfall index, the seasonal expectation rainfall index $_sRI_i$ and the seasonal expectation of the discharge) has the best performance results (i.e. highest R^2 values) among the four models.

The performance of the ANN2 model, which uses two external inputs, namely the rainfall index and the seasonal expectation rainfall index, is not significantly different from the ANN1 model which only uses the rainfall index as input information. Thus, the incorporation of the seasonal expectation rainfall index as an additional external input does not necessarily yield substantial improvement in model performance.

The table also shows that the R^2 values of the ANN2 model which uses the rainfall index and seasonal expectation of the discharge are substantially better than those of the ANN1 model. This demonstrates that the use of the seasonal expectation of the discharge as an extra external input leads to significant improvement in model performance. Furthermore, the table indicates that the performance of the ANN3 model which uses two external inputs (the

Figure 4 │ The R^2 value of the four ANN rainfall–runoff models using two and three hidden neurons.

Table 2 │ The R^2 efficiency values of the four ANN models

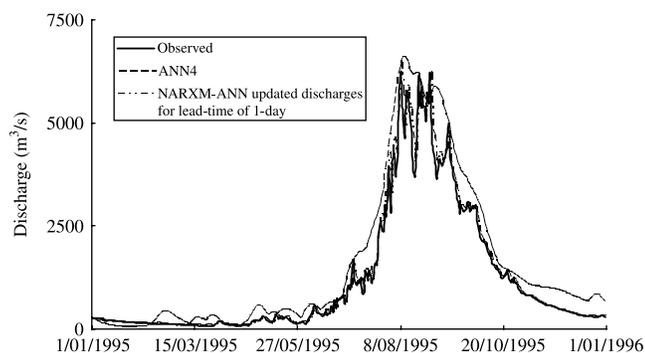|	R^2(%)			
	ANN1	**ANN2**	**ANN3**	**ANN4**
Calibration period	82.98	83.7	91.18	93.64
Verification period	72.32	78.04	85.27	87.21

Figure 5 | Comparison between observed and estimated discharge hydrographs.

rainfall index and the seasonal expectation of the discharge) is better than that of the ANN2 model. Figure 5 shows a plot of the observed and the ANN4 model estimated discharges for the verification year. The figure indicates that the ANN4 model overestimates the observed discharges. In the verification period, the R^2 value of the ANN4 model is 87.21%, which can be regarded as a strong indication of very good model performance. However, Figure 5 shows there are many instances where the ANN4 fails to provide satisfactory estimates of the observed discharges, especially during the high flood season. This is one of the problems associated with the use of the numerical indices such as the R^2 efficiency criterion to evaluate model performance. In the case of the ANN4 model, it appears this high R^2 value is dominated by the model performance during the low flow season, which is very long in the case of the Blue Nile basin. The use of graphical plots such as Figure 5 would help to identify deficiencies in the model performance which are not easily detectable from the R^2 results. This clearly highlights the importance of using both quantitative (numerical efficiency indices) and qualitative (graphical plots) criteria for evaluating model performance as recommended by ASCE (1993).

Updating of the simulation mode estimated discharges of the rainfall–runoff models using the NARXM-ANN updating procedure

As the ANN4 rainfall runoff model has the best results its simulated discharges have been chosen for updating by the NARXM-ANN updating procedure. As mentioned earlier in this paper the calibration of the NARXM-ANN updating procedure requires the specification of the number of hidden neurons and the values of the autoregressive (p) and the moving average (q) orders. The number of hidden neurons and these orders are estimated by a trial-and-error procedure similar to that used in estimating the hidden neurons of the ANN rainfall–runoff models. In this case, the trial-and-error procedure involves iteratively calibrating the network using combinations of different numbers of hidden neurons and different order values. It has been found that the optimum values of the number of hidden neurons, the autoregressive order and the moving average order are 2, 1 and 2, respectively.

Table 3 shows the R^2 values for lead 1 to 6 day of the NARXM-ANN updating procedures applied to the simulation mode discharge of the ANN4 model. The table also displays the corresponding R^2 values of the naive persistence predictor-updating model (PPM) (i.e. the 'no-river flow forecasting-model' situation). This model considers that the discharge forecast over the lead-time of any magnitude is simply equal to the observed discharge at the time of making the forecast. This naive PPM updating model is used in this study merely as a benchmark for comparing the performance of the substantive NARXM-ANN updating procedure.

Comparison of Tables 2 and 3 shows that the updating of the simulation mode discharges of the ANN4 rainfall–runoff model by the NARXM-ANN updating procedures has substantially improved the corresponding R^2 values of

Table 3 | The lead-time R^2 (%) efficiency values of the NARXM-ANN and the PPM updating procedures

| Model | Calibration period | | | | | | Verification period | | | | | |
| | Lead-time | | | | | | Lead-time | | | | | |
	1-day	2-day	3-day	4-day	5-day	6-day	1-day	2-day	3-day	4-day	5-day	6-day
ARXM	98.25	97.06	96.13	95.72	95.53	95.39	97.80	96.07	94.76	93.78	92.88	92.14
PPM	97.88	95.75	93.2	91.09	89.32	87.67	97.69	95.67	93.89	92.63	91.42	90.31

the different lead-times for both the calibration and verification periods.

Further examination of Table 3 indicates the NARXM-ANN updating procedure, operating on the simulation mode discharge of the ANN4 model, performs substantially better than the naive PPM updating procedure for all lead-times for both the calibration and verification periods.

Furthermore, inspection of Table 3 shows that the lead-time R^2 efficiency values of the updating procedures decrease with the increase in the value of the lead-time. However, the rate of decrease in the R^2 efficiency values of PMM is higher than that of the NARXM-ANN updating procedure.

Figure 5 shows a time series plot of the NARXM-ANN updated discharge forecasts, the simulation mode discharge of the ANN4 model and the observed discharge. The figure indicates that the updating of the estimated discharges of the ANN4 model by the NARXM-ANN output procedure is very successful.

CONCLUSIONS

In the present study, four Artificial Neural network (ANN) rainfall–runoff models are developed for the Blue Nile River in Sudan. A common feature of these four models is that they are based on the structure of the multi-layer perceptron (MLP). These models differ in terms of the external inputs being used by the model. The four models use the rainfall index as a common external input. The first model (ANN1) only uses this common external input. Two of the models (ANN2 and ANN3) use either the seasonal expectation rainfall index or the seasonal expectation of the discharge as additional external input information. However, the fourth remaining model (ANN4) uses both the seasonal expectation rainfall index and the seasonal expectation of the discharge as additional external input information. The results reveal the ANN4 model has the best performance (i.e. the highest R^2 values) among the four models developed in this study.

A real-time river forecasting model consisting of the ANN4 rainfall–runoff model and the add-on NARXM-ANN output updating procedure is developed in this study. The results show that the updating of the estimated discharges of the ANN4 model by the NARXM-ANN updating procedure significantly enhances the quality of the discharge forecasts.

The results obtained in this study support the premise that neural network models have considerable potential and promise to be used as an alternative approach for river flow forecasting in developing countries. The results also show the selection of appropriate external inputs for the neural network model is very important in its success. A combination approach based on the utilisation of prior hydrological knowledge about the catchment and the use of trial-and-error procedures as adopted in this study can successfully be used in determining the external inputs to the neural network model.

In future applications of neural network models to river flow forecasting in the Blue Nile River consideration should be given to using different neural network types other that the MLP and different data-driven models which may lead to further improvements in the forecasting performance.

ACKNOWLEDGEMENTS

REFERENCES

Abrahart, R. J., Heppenstall, A. J. & See, L. M. 2007 Timing error correction procedure applied to neural network rainfall–runoff modelling. Hydrol. Sci. J. 52(3), 414–431.

Antar, A. M., Elassiouti, I. & Allam, M. N. 2005 Rainfall–runoff modelling using artificial neural networks technique: a Blue Nile catchment case study. Hydrol. Process. 20(5), 1201–1216.

ASCE Task Committee on Definition of Criteria for Evaluation of Watershed Models of the Watershed Management Committee Irrigation and Drainage Division 1993 Criteria for evaluation of watershed models. J. Irrig. Drain. Eng. 119(3) 429–442.

Berz, G. 2000 Flood disasters: lessons from the past—worries for the future. Water Marit. Eng. Proc. Inst. Civil Eng. 142(1), 3–8.

Blum, A. 1992 Neural Networks in C++; An Object-oriented Framework for Building Connectionist Systems. John Wiley & Sons, New York.

Boucher, M. A., Perreault, L. & Anctil, F. 2009 Tools for the assessment of hydrological ensemble forecasts obtained by neural networks. J. Hydroinform. 11(3–4), 297–307.

Bowden, G. J., Dandy, G. C. & Maier, R. H. 2005 Input determination for neural network models in water resources applications. Part 1–background and methodology. *J. Hydrol.* **301**(1–4), 75–92.

Chang, F.-J. & Chen, Y. C. 2001 A counterpropagation fuzzy-neural network modeling approach to real time streamflow prediction. *J. Hydrol.* **245**(1–4), 153–164.

Chicago Sun-Times 1988 Article: 2 million homeless in Sudan flood. Aug 4 1988 Khartoum, Sudan, Africa. Available at: http://www.highbeam.com/doc/1P2-3899698.html

Conway, D. 1997 A water balance model of the Upper Blue Nile in Ethiopia. *Hydrol. Sci. J.* **42**(2), 265–286.

Coulibaly, P., Anctil, F. & Bobée, B. 2000 Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *J. Hydrol.* **230**(3–4), 244–257.

Coulibaly, P., Babovic, V., Cluckie, I., Mynett, A. & Ball, J. (eds) 2009 *Advances in Hydroinformatics*. IWA Publishing, London.

Dawson, C. W., Abrahart, R. J. & See, L. M. 2007 HydroTest: a web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts. *Environ. Model. Softw.* **22**(7), 1034–1052.

Dawson, C. W., Abrahart, R. J. & See, L. M. 2009 HydroTest: further development of a web resource for the standardised assessment of hydrological models. *Environ. Model. Softw.*

Dawson, C. W., See, L. M., Abrahart, R. J. & Heppenstall, A. J. 2006 Symbiotic adaptive neuro-evolution applied to rainfall–runoff modelling in northern England. *Neural Networks* **19**(2), 236–247.

Dawson, C. W. & Wilby, R. L. 2001 Hydrological modelling using artificial neural networks. *Prog. Phys. Geogr.* **25**(1), 80–108.

Department for Environment, Food and Rural Affairs (DEFRA)/ Environmental Agency (EA) 2005 *The Social Performance of Flood Warning Communications Technologies*. Technical Report W5C-016. Environment Agency, Bristol.

Dibike, Y. B. & Solomatine, D. P. 2001 River flow forecasting using artificial neural networks. *Phys. Chem. Earth Part B Hydrol. Ocean. Atmos.* **26**(1), 1–7.

El Sebai, I. H. 1998 *The Application of the Remote Sensing and Geographical Information Systems in Hydrologic Modelling of the Blue Nile Basin*. MSc Thesis, Department of Engineering Hydrology, National University of Ireland, Galway, Ireland, unpublished.

Elmahi, A. B. & O'Connor, K. M. 1995 The application of the multiple input simple linear model and the multiple input linear perturbation model to flow forecasting. In: *Proc. Nile 2000 Conference, Kampala (Uganda)*.

Eltahir, E. A. 1996 El Niño and the natural variability in the flow of the Nile River. *Water Resour. Res.* **32**(1), 131–137.

Fernando, A. & Shamseldin, A. Y. 2009 Investigation of the internal functioning of the radial basis function neural network river flow forecasting models. *J. Hydrol. Eng.* **14**(3), 1–7.

Goswami, M., O'Connor, K. M., Bhattarai, K. P. & Shamseldin, A. Y. 2005 Real-time river flow forecasting for the Brosna

catchment in Ireland using eight updating models. *J. Hydrol. Earth Syst. Sci.* **9**, 394–411.

Grijsen, J. G., Snoeker, X. C., Vermeulen, C. J. M., Nur Mohamed, E. A. M. & Mohamed, Y. A. 1992 An information system for flood early warning. In *Floods and Flood Management* (ed. A. J. Saul), pp. 263–289. Kluwer, Dordrecht.

IIASA (International Institute for Applied Systems Analysis) 2009 *Natural Catastrophes and Developing Countries*. Available at: http://www.iiasa.ac.at/Research/CAT/research.html

Kachroo, R. K. & Liang, G. C. 1992 River flow forecasting. Part 2: algebraic development of linear modelling techniques. *J. Hydrol.* **133**, 17–40.

Khatibi, R. & Cluckie, I. 2006 Integration and goal-orientation in flood forecasting and warning. In *River Basin Modelling for Flood Risk Mitigation* (ed. D. W. Knight & A. Y. Shamseldin),. Taylor & Francis, London, Chapter 30.

Maier, H. R. & Dandy, G. C. 2000 Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environ. Model. Softw.* **15**, 101–124.

Masters, T. 1993 *Practical Neural Networks Recipes in C++*. Academic, New York.

Nash, J. E. 1957 The form of the instantaneous unit hydrograph. *Int. Assoc. Sci. Hydrol. Publ.* **45**(3), 114–118.

Nash, J. E. & Sutcliffe, J. V. 1970 River flow forecasting through conceptual models. Part 1: a discussion of principles. *J. Hydrol.* **10**, 282–290.

Nelder, J. A. & Mead, R. 1965 A simplex method for function optimization. *Comput. J.* **7**, 308–313.

Parent, A. C., Anctil, F., Cantin, V. & Boucher, M. A. 2008 Neural network input selection for hydrological forecasting affected by snowmelt. *J. AWRA* **44**(3), 679–688.

Pramanik, N. & Panda, R. K. 2009 Application of neural network and adaptive neuro-fuzzy inference systems for river flow prediction. *Hydrol. Sci. J.* **54**(2), 247–260.

Press, W. H., Flannaery, B. P., Teukolsky, S. A. & Vetterling, W. T. 1989 *Numerical Recipes*, pp. 301–306. Cambridge University Press, New York.

Price, K. R. & Jemberie, A. A. 2006 The application of hydroinformatics. In *River Basin Modelling for Flood Risk Mitigation* (ed. D. W. Knight & A. Y. Shamseldin). Taylor & Francis, London, Chapter 6.

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.

Said, R. 1999 Solving the riddle of the Nile. *Al-Ahram Weekly* 16–22 September, no. 447. Available at: http://weekly.ahram.org.eg/1999/447/spec2.htm

Salas, J. D., Delleur, J. W., Yevjevich, V. & Lane, W. L. 1980 *Applied Modelling of Hydrologic Time Series*. Water Resources Publications, Littleton, CO.

Senbeta, D. A., Shamseldin, A. Y. & O'Connor, K. M. 1999 Modification of the probability distributed soil moisture storage capacity model. *J. Hydrol.* **224**, 149–168.

Shahin, M. 1985 *Hydrology of the Nile Basin. Developments in Water Science 21*. Elsevier, Amsterdam.

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

Shamseldin, A. Y. 2006 Topics related to rainfall–runoff models. In *River Basin Modelling for Flood Risk Mitigation* (ed. D. W. Knight & A. Y. Shamseldin). Taylor and Francis, London, Chapter 10.

Shamseldin, A. Y., Abdo, G. M. & Elzein, A. S. 1999 Real-time flood forecasting on the Blue Nile River using the SMAR-AR model. *Water Int.* **24**(1), 39–45.

Shamseldin, A. Y., Nasr, A. E. & O'Connor, K. M. 2002 Comparison of different forms of the Multi-layer Feed-Forward Neural Network method used for river flow forecast combination. *J. Hydrol. Earth Syst. Sci.* **6**(4), 671–684.

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

Shamseldin, A. Y. & O'Connor, K. M. 2003 Consensus real-time river flow forecasting model for the Blue Nile River. *IAHS Publ.* 281, pp. 82–89.

Shamseldin, A. Y., O'Connor, K. M. & Nasr, A. E. 2007 A comparative study of three neural network forecast combination methods for river flow forecasting. *Hydrol. Sci. J.* **55**(2), 898–916.

Thirumalaiah, K. & Deo, M. C. 2000 Hydrological forecasting using neural networks. *J. Hydrol. Eng.* **5**(2), 180–189.

Todini, E. 2006 Holistic flood management and decision support systems. In *River Basin Modelling for Flood Risk Mitigation* (ed. D. W. Knight & A. Y. Shamseldin). Taylor & Francis, London, Chapter 16.

United Nations (UN) 2009 *Guidelines for Reducing Flood Losses*. Available at: http://www.unisdr.org/eng/library/isdr-publication/flood-guidelines/Guidelines-for-reducing-floods-losses.pdf

United Nations Educational, Scientific and Cultural Organization (UNESCO) 2005 *Word Assessment Programme for Development Capacity Building and the Environment*. Available at: http://www.unesco.org/water/wwap/facts_figures/managing_risks.shtml

United States Bureau Of Reclamation (USBOR) 1964 *Land and Water Resources of the Blue Nile Basin*. US Department of Interior.

Wheater, H. S., Jakeman, A. J. & Beven, K. J. 1993 Progress and directions in rainfall–runoff modelling. In *Modelling Change in Environmental Systems* (ed. A. J. Jakeman, M. B. Beck & M. J. McAleer). John Wiley & Sons, New York, Chapter 5.

World Meteorological Organization (WMO) 1992 *Simulated Real-time Intercomparison of Hydrological Models*. Oper. Hydrol. Rep. 38, No. 779, WMO, Geneva.

World Metrological Organization (WMO) 2005 *Preliminary Survey of Early Warning Systems Background Document in Support of the UN Secretary General Kofi Annan Report to the General Assembly*. 29 July. WMO, Geneva.