KSCE Journal of Civil Engineering (2017) 21(2):512-522
Copyright ⓒ2017 Korean Society of Civil Engineers
DOI 10.1007/s12205-017-1514-9

Design Optimization and Applications in Civil Engineering

pISSN 1226-7988, eISSN 1976-3808
www.springer.com/12205

Flow Forecasting in Multiple Sections of a River System

Joseph Tripura* and Parthajit Roy**

Received August 31, 2016/Accepted November 15, 2016

Abstract

A river system includes the combination of flows occurring simultaneously in the main river and its contributing tributaries. Any change in the flow condition of the river system is caused due to changes in flow of the main river and/or contributing tributaries. An accurate flow forecasting at multiple sections of a river system is worthy for issuing early warning to the imminent floods and in regulating the reservoir outflows. The application of multiple-inputs and multiple-outputs (MIMO) model is an effective way for simultaneous flow forecasting as it provides interrelation among multiple input and multiple output variables simultaneously. In the present study an Artificial Neural Networks (ANN) based MIMO model has been developed for Barak river system in Assam, India using Partially Recurrent Neural Network (PRNN) and Nonlinear Autoregressive with Exogenous Inputs (NARX) approaches. Performance of the model using both NARX and PRNN provide an efficacy with coefficient of efficiency (CE) > 0.87 and Mean Absolute Percentage Error (MAPE) < 7.66% at 12 hour lead time forecasting. This indicates satisfactory model performances for simultaneous flow forecasting at multiple sections of a river system however the results obtained by MIMO using NARX (MIMO-NARX) perform better than MIMO using PRNN (MIMO-PRNN) in terms of statistical performance criterion.

Keywords: *river systems, NARX, PRNN, MIMO model, short range forecasting*

1. Introduction

Flow in a river system is a complex dynamic process characterized by temporal and spatial variation. An accurate and timely flow forecasting in a river system is considered to be an important task for issuing early warning of the imminent floods and regulating the reservoir outflows. In recent years many nonlinear approaches, such as the Artificial Neural Network (ANN), Genetic Algorithm (GA), and Fuzzy logic approaches, have been widely used in the field of hydrology and water resources with encouraging results (Karunanithi *et al*., 1994; Coulibaly *et al*., 2001; Trajkovic *et al*., 2003; Adeloye and Munari, 2006; Kisi, 2008; Wang and Trarore, 2009; Perera and Lahat, 2014). Different type of ANN algorithms used by the researchers in modeling water resources problems include: generalized regression neural network for river sediment load (Cigizoglu and Alp, 2006); feed forward back propagation network in estimating evapotranspiration (Wang *et al*., 2008), radial bias function in stream flow forecasting (Kagoda *et al*., 2010) etc. ASCE task committee (2000) summarized the application of ANN in hydrology and concluded that ANN can be used as alternative modeling tool subject to further exploration.

There is a considerable volume of literature on the applications of ANN models in river flow studies. Some of the applications include the works by Thirumalaiah and Deo 1998; Chau *et al*. 2005; Dibike and Solomatine, 2001; and Coulibaly and Evora (2007) etc. Most of the ANN based flow models rely on flow matching techniques and employ at site and/other site flow variables along with a set of exogenous variables to predict a flow. The routing type ANN models (Chou *et al*., 2005; Thirumalaiah and Deo, 1998) use other site(s) flow variables along with or without exogenous variables to predict a flow, usually a downstream flow in river reaches.

Hydrological researchers in general focused on Multiple-Input Single-Output (MISO) model or Single-Input Single-Output (SISO) model for river flow studies without giving due emphasis on Multiple-Input Multiple-Output (MIMO) model. A MISO or MIMO model considers the predicted future values along with current and past values for predicting the next time step value. Application of MIMO model can be found in work of Maciel *et al. (*2012), other field includes interest rate forecasting, applied economics and management studies (Yue and Wang, 2007; Bontempi and Taieb, 2010; Claveria *et al*., 2015). But a very limited application was noticed in dealing with hydrological problems. Most of the ANN models for hydrological problems available in literature except a few use Multilayer Layer Perceptron (MLP). MLPs are static linear networks and cannot recognize temporal variation(s).

In case of a river system different multiple flows representing various upstream catchment characteristics joins together to form a common downstream flow. The joining of multiple flows

*Ph.D. Scholar, Dept. of Civil Engineering, National Institute of Technology Silchar, 788010, India (Corresponding Author, E-mail: tripurajoseph89@gmail.com)
**Assistant Professor, Dept. of Civil Engineering, National Institute of Technology Silchar, 788010, India (E-mail: parthajit_roy@yahoo.co.in)

having different temporal and spatial characteristics makes the flow in a river system a further complex dynamic process. The flow at the bounding sections of a river system and the intervening storage varies coherently having a simultaneous correlation among the participating flows and the intervening storage. The unsteady storage variation depicting flood condition(s) in the river reach(s) is reflected through the variation rate(s) of participating flow(s). MISO or MIMO based ANN model can recognize any complex temporal variation(s) to be considered for non-linear time series studies. Choudhury and Roy (2015) investigated the application of focused Time Delay Neural Network (TDNN) and Gamma Memory Neural Network (GMNN) for single step flow forecasting in a river system. Both TDNN and GMNN can recognize and consider the temporal variation of non-linear time series along with current event for modeling. TDNN has a user defined memory consisting of embedded feed forward delays. The GMNN has an adaptable memory unit with local recurrent connection that helps the memory to define its length on the basis of individual input series. An ANN with focused memory includes memory in the input layer only. Thus an ANN model with focused TDNN memory takes into account pre-defined number of exact past values along with the current value of the time series for its prediction. A TDNN looks more like a feedforward network, because time aspect is only inserted through its inputs. The GMNN memory location contains representative values of past samples of time series and thus the time series prediction using a focused GMNN or TDNN depend on past and current values only. Both the TDNN and GMNN do not consider any predicted/real future value as input as they do not have any feedback from output. The focused TDNN and GMNN are limited to direct multistep ahead forecasting.

The present study investigates the application of MIMO model for multi-step ahead simultaneous flow forecasting at multiple sections simultaneously in a river system using NARX neural networks. The literature shows very limited use of MIMO neural network for simultaneous flow forecasting at multiple sections of a river system. The recurrent connection of NARX networks can hold traces of the predicted future values which are processed further by the network along with current and past values for modeling. Flow in a river system involving multiple flows with variable characteristics is highly non-linear and complex. Accuracy of flow forecasting in a river system increases with the use of future predicted flow series in addition to observed flow series. Recurrent networks like NARX networks can suitably be used as an alternative ANN model for simultaneous flow forecasting in a river system.

Further, a time series can also be predicted based on the information of current values directly and some of its past values. The input layer in a partial recurrent neural network has a direct connection with the output layer by-passing the hidden layer. The hidden layer has got local recurrent connection via context units. This helps the network to delay the processed information on input values at the hidden units. The network can suitably use this information along with the current values at the output unit for the model output. There is no evidence in the literature about application of PRNN network for simultaneous flow forecasting in river system. Present study also investigates the application of PRNN network for simultaneous flow forecasting. Thus, the modelling issues using both PRNN and NARX have been investigated for MIMO model in a Barak river system of Barak Valley (Southern Assam, India).

The remainders of the paper are organized as follows. Section 2, 2.1 describes NARX network module and its training algorithm, section 3, 3.1 describes PRNN module and training algorithm, section 4 explains discussion on models application and results, finally the conclusion is presented in section 5.

2. Nonlinear Autoregressive with Exogenous Inputs (NARX) Networks

NARX is one of the typical paradigms of complex recurrent models followed by state space model and Recurrent Multilayer Perceptron (RMLP). The information retaining capacity of NARX network is twice to thrice than that of conventional recurrent networks (Lin *et al.,* 1997). This makes them more dynamic and suitable than a conventional recurrent network (Siegelman, 1997) with limited information retaining capacity. Nonlinear dynamic system especially time series can be appropriately modelled by NARX (Diaconescu, 2008). NARX can also capture well the peak values (Chang *et al.*, 2015) of a nonlinear dynamic time series like river flow. Fig. 1 shows a typical NARX neural network. The inputs to the network consist of current inputs and its previous sequences plus delayed versions of the network outputs. The embedded memory shown in Fig. 1 consists of two sets of tapped delay lines - first set of of order q for receiving current inputs and their previous sequences and second set of order p for receiving the network outputs and

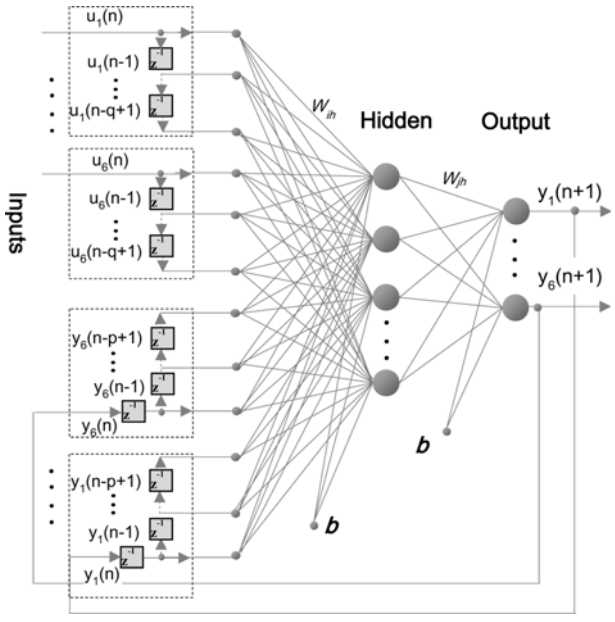

Fig. 1. A Typical NARX Model in the Study

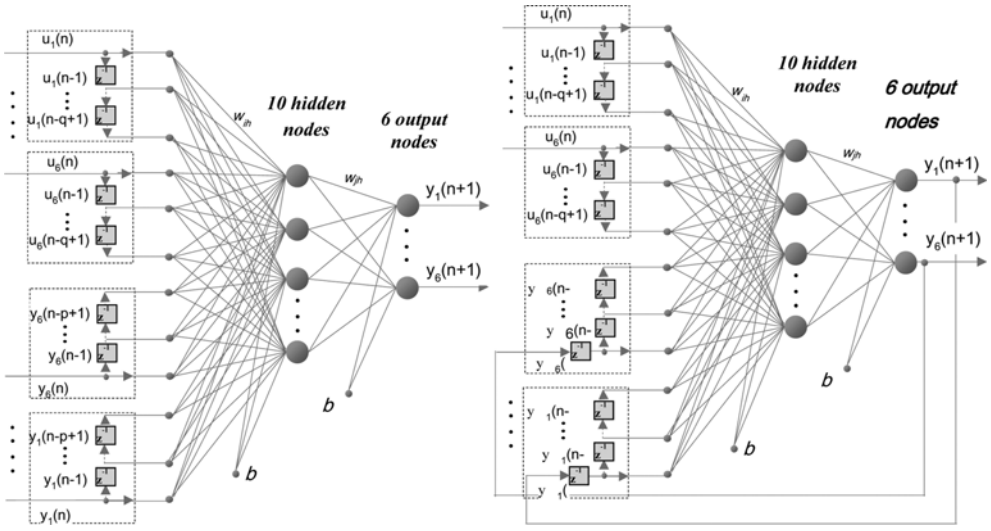

Fig. 2. (a) Represent SP-mode NARX Network During Training Phase, (b) Represent P-mode During the Testing and Predictions Phase for MIMO Model in the Study

their previous sequences through feedback connections. Such inputs can be viewed as the transformation of temporal to other spatial dimensions and are often called as *time window* due to its partial viewed on portion of the series. Inputs series to the network being simultaneously synchronized in case of NARX the times series forecasting is improved.

NARX neural networks can be represented in a state space form (Diaconescu, 2008) as

$$y_k(n+1)=\begin{Bmatrix}\sigma\left(u(n),y_i(n)\right) & for\quad i=1\\ y_i(n) & for\quad i=2,3,.....\ N\end{Bmatrix} \tag{1}$$

This discrete time state-space model form can be further represented in a simple non-linear difference equations as,

$$y_k(n+1)=\sigma\Big[(y_k(n),y_k(n-1),.....y_k(n-p+1)),\\ u(n),u(n-1),.....u(n-q+1)\Big] \tag{2}$$

Where $y_k(n+1)$ is the network output/or predicted output and $(u(n),y_k(n))$ are the input-output pair at time n, and $q\geq p$, σ is the non-linear differentiable function $(\sigma:\mathcal{R}^{p+q}\to\mathcal{R})$. Thus output of the system at time n+1 depends on n past values $y_k(n-j)$ for $(j=0,1,2,\ldots\ldots,p-1)$ similarly, for the q past values of the input $u(n-l)$ for $(l=0,1,2,\ldots\ldots,q-1)$.

2.1 NARX Network Training

NARX networks in general can be trained through two modes: Parallel mode (or P-mode) and series parallel mode (or SP-mode).Mathematically both modes can be expressed as

$$y_k(n+1)=\sigma\begin{cases}\hat{y}_p(n),\hat{u}(n);w\\ y_{sp},u(n);w\end{cases} \tag{3}$$

Here $\sigma(\cdot)$ is the nonlinear function that can be implemented through an MLP. The basic difference between both modes is that P-mode contains p past values of the estimated time series whereas SP-mode contains the q past values of actual time series, other terms like $y(n+1)$, $u(n)$ so on, were defined in section 2. In the present study SP-mode is preferred because system's output regressor(s) becomes more precise as we train the network with actual past outputs. Secondly, the application of direct method of prediction becomes more meaningful to extract the system dynamic features with better predictive quality which is obligatory for MIMO model. In a typical NARX network shown in Fig. 2 two feedback loops are observed during the testing and predictions phase for both the input regressor $u(n)$ and the output regressor $y(n)$ simultaneously. However during the training phase such loops are not used. Hence, closed loop operation is done after the training phase to extract multi-steps ahead prediction.

The optimization technique used to train the network regularization is the standard Levenberg-Marquardt. This technique adaptively varies the parameter update between gradient descent and Gauss-Newton update; say for a function $\sigma(w)$ to be minimized with respect to weight vector w, update of the weight vector can be defined in the form as,

$$w(n+1)=w(n)+\Delta w \tag{4}$$

Where, $\Delta w=-\dfrac{\Delta\sigma(w)}{\nabla^2\sigma(w)}$ (5)

The gradient of $\sigma(w)$ and the hessian matrix of $\nabla^2\sigma(w)$ are defined as,

$$\nabla\sigma(w)=J^T(w).E_k(w) \tag{6}$$

And

$$\nabla^2\sigma(w)=J^T(w).J(w)+\sum_{k=1}^{N}E_k.\nabla^2E_k(w) \tag{7}$$

Where, $J(w)$ is the *Jacobian* matrix and $E(w)$ is the vector of network errors. Thus, Eq. (4) can be equated as,

$$w(n+1)=w(n)-\left[J^T(w).J(w)+\mu I\right]^{-1}.J^T(w).E_k(w) \tag{8}$$

The scalar μ decides the update rule. When μ is zero, using the

approximate Hessian matrix Eq. (8) reduces to Newton's method and when μ is larger, this becomes just gradient descent with a small step size. Since, Newton's method proves to be more accurate and faster (Marquardt, 1963) than gradient descent resulting to a near minimum, the shift is tried to move towards Newton's method as quick as possible. Therefore, μ is decreased after each successful step or else increase when performance function is enhanced.

3. Partially Recurrent Neural Networks (PRNNs)

PRNN is obtained by constraining a Fully Recurrent Neural Network (FRNN) in which certain weights are pruned (forcing weights to zero). Many such recurrent networks architecture is possible by detaching feedback links from output neurons of FRNN. Robinson *et al*. (1991) obtained a PRNN by constraining a FRNN by pruning certain weights and was used for speech recognition instead of general FRNN. Application of many of its kind other than the applied PRNNs can also be found in the works of (Weigend and Rumelhart, 1992; Elizondo *et al*., 1997; Chang *et al*., 2012; Navarra *et al*., 2015). For any recurrent networks the fundamental feature of the architecture is that it contains at least one feed-back connections to make the activation flow round the loop (Mansouri and Mohammad, 2014) whereas PRNNs accomplished its feed-back connections from hidden to inputs and itself.

The use of PRNN network avoids the coincidence between the state and output and also reduces number of weights. In addition PRNN makes the network computation and training algorithm more localized in space. Fig. 3 shows the PRNN network consisting of input layer, output layer and single hidden layer used in the present study.

Direct connections between input nodes and output nodes may be advantageous to model non-linear structure of the data and increases its ability in capturing the underlying complexity (Zhang *et al*., 1998). The feedback connection in the network felicitates the use of neuron states for the feed-forward networks.

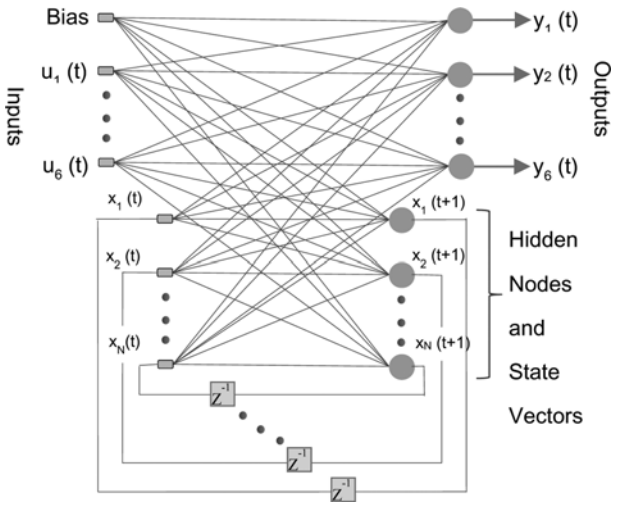

Fig. 3. Partially Recurrent Neural Network structure with Unit delays

The MIMO model form described by the dynamical system of PRNN in state space form for discrete time t is defined as

$$\begin{aligned} x_p(t+1) &= \sigma_p\{x_p(t), u_p(t)\} \\ y_p(t+1) &= f_p\{x_p(t), u_p(t)\} \end{aligned} \tag{9}$$

Where x_p represent the state vector of the system, u_p the system input vector and y_p the system output vector at discrete time t. Here p represent the state-space system, σ_p and f_p are the vector functions where σ_p is duplicated inside functions f_p (output vectors). The PRNN in this study is obtained by pruning the feedback loops from all output nodes from a FRNN. This conforms to a separate output equations which are function of external inputs and states. Thus, the values of output nodes are no longer a part of the state vector. The neuron outputs which are feedback into the network summarizing the information of the past become part of the state. Thus, Eq. (9) can be further modified to represent the PRNN in a specific way by redefining the state vector and vector functions. It is explained by considering a network with Q numbers of processing elements (PEs), M external outputs, N external inputs. W_i and W_o are the weights for all feedback loops and the links connecting all the external input nodes to hidden or output nodes. Weight matrixes for FRNN are defined as:

$$W_i = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1Q} \\ w_{21} & w_{22} & \cdots & w_{2Q} \\ \cdots & \cdots & \cdots & \cdots \\ w_{Q1} & w_{Q2} & \cdots & w_{QQ} \end{pmatrix}; \quad W_o = \begin{pmatrix} w_{1;Q+1} & w_{1;Q+2} & \cdots & w_{1;Q+N} \\ w_{2;Q+1} & w_{2;Q+2} & \cdots & w_{2;Q+N} \\ \cdots & \cdots & \cdots & \cdots \\ w_{Q;Q+1} & w_{Q;Q+2} & \cdots & w_{Q;Q+N} \end{pmatrix} \tag{10}$$

Now weights matrices for PRNN excluding the removed connections from Eq. (12) are then redefined as

$$W_{iU} = \begin{pmatrix} w_{1;M+1} & w_{1;M+2} & \cdots & w_{1Q} \\ w_{2;M+1} & w_{2;M+2} & \cdots & w_{2Q} \\ \cdots & \cdots & \cdots & \\ w_{M;M+1} & w_{M;M+2} & \cdots & w_{MQ} \end{pmatrix};$$

$$W_{iZ} = \begin{pmatrix} w_{M+1;M+1} & w_{M+1;M+2} & \cdots & w_{M+1;Q} \\ w_{M+2;M+1} & w_{M+2;M+2} & \cdots & w_{M+2;Q} \\ \cdots & \cdots & \cdots & \cdots \\ w_{Q;M+1} & w_{Q;M+2} & \cdots & w_{Q;Q} \end{pmatrix} \tag{11}$$

$$W_{oU} = \begin{pmatrix} w_{1;Q+1} & w_{1;Q+2} & \cdots & w_{1;Q+N} \\ w_{2;Q+1} & w_{2;Q+2} & \cdots & w_{2;Q+N} \\ \cdots & \cdots & \cdots & \cdots \\ w_{M;Q+1} & w_{M;Q+2} & \cdots & w_{M;Q+N} \end{pmatrix};$$

$$W_{oZ} = \begin{pmatrix} w_{M+1;Q+1} & w_{M+1;Q+2} & \cdots & w_{M+1;Q+N} \\ w_{M+2;Q+1} & w_{M+2;Q+2} & \cdots & w_{M+2;Q+N} \\ \cdots & \cdots & \cdots & \cdots \\ w_{Q;Q+1} & w_{Q;Q+2} & \cdots & w_{Q;Q+N} \end{pmatrix} \tag{12}$$

Where, W_{iU} and W_{iZ} are the combined effective weights of weight W_i, weight W_{0U} and W_{0Z} the corresponding combined weight elements of weight W_0. Thus, Eq. (9) is rewritten as

$$x_p(t) = \sigma_p\{W_{iZ}x_p(t-1) + W_{oZ}u_p(t)\}$$
$$y_p(t) = f_p\{W_{iU}x_p(t-1) + W_{oU}u_p(t)\} \quad (13)$$

Note that the matrix size for weights W_{iU}, W_{iZ}, W_{0U} and W_{0Z} are (Q by $Q-M$), ($Q-M$ by $Q-M$), (M by N) and ($Q-M$ by N).

3.1 PRNN Training

Weights in PRNN network are updated using supervised learning, which means learning are made by means of input and error (output and target response difference). This requires some basic assessment like choice of error criterion that computes error using cost functions, mechanisms for network parameter modifications and for decision in constraining the network output and target values. PRNN network can be trained by both trajectory learning and fixed point learning. In this study, trajectory learning method is followed for training PRNN network since for different time steps, the actions are coupled by policy parameter which makes trajectory optimization easier (Levine and Koltun, 2014). Constraining the network system output is the goal of trajectory learning through a time period. The, cost function for trajectory learning can be written as,

$$E = \sum_{n=1}^{\tau}\sum_{i}\left(d_i(n) - y_i(n)\right)^2 \quad (14)$$

Where τ being training sequence length which is taken to be 50 in the present study and i is the output index units. Two types of learning procedures available in literatures- Back Propagation Through Time (BPTT) (Rumelhart *et al.*, 1986; Williams and Zipser, 1995) and Real Time Recurrent Learning (RTRL) (Williams and Zipser, 1989). The present study used BPTT algorithm for network training. The network training using BPTT runs forward in time until the end of the trajectory and the activation of Each Processing Elements (PEs) is stored locally in a memory structure for each time step. The output errors are computed and then back-propagates as in static network in the form of Eq. (15) defined as

$$\frac{\partial E}{\partial W_{ij}} = \sum_{n=1}^{\tau}\delta_i(n)\phi'(net_i(n))x_j(n-1) \quad (15)$$

The change in weight for any particular weight w_{ij} is adjusted based on gradient search in steepest directions of error surface. Similar process is being implemented for both the weights connecting hidden and output nodes. On the other hand poor performance of the test set is likely if training set doesn't envoy input data class and it happens sometimes even though performances in training set is excellent.

4. Discussion on Models Application and Results

4.1 Study Area

Application of MIMO model is demonstrated using PRNN and NARX neural networks for simultaneous flows forecasting at multiple sections in Barak river system. The Barak river system in Barak Valley lies in the southern part of Assam (India). It is the second largest river system in North-East India lying between 24° 8' to 25°8' N latitudes and 92° 15' to 93° 15' E longitudes. Fig. 4 shows the Barak valley and Barak river system. Out of the total length of about 900 km, it traverses for about 532 km in Indian Territory with 129 km in Assam. The

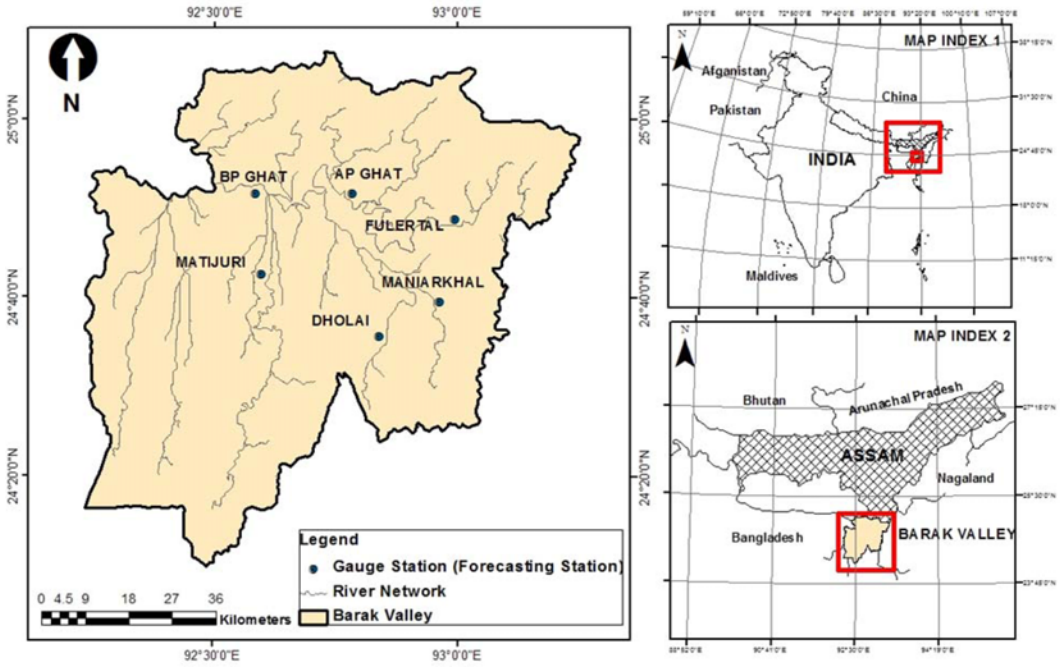

Fig. 4. Barak Valley and Barak River System Showing the Location of Six Gauging Stations

Table 1. Description of Site Locations for the Entire Gauge Stations/forecasting Stations

Stations Code	Gauge Stations	Locations (River Name)	Remarks*
01-11-01-007	Badarpurghat	Barak	Main
01-11-01-003	Matijuri	Katakhal	Tributary
01-11-01-008	Annapurnaghat	Barak	Main
01-11-06-004	Dholai	Rukni	Tributary
01-11-01-005	Maniarkhal	Sonai	Tributary
01-11-01-009	Fulertal	Barak	Main

sub-basin is situated on the route of south-west monsoon. It receives average annual rainfall of 2500-4000 mm, with 80-85% from Mid-April to Mid-October. Generally 70-75% of the total population of the area depend on agriculture and the impact of recurrent flood jeopardizes the socioeconomic growth of the region. During study period also it received 2-3 flood waves in monsoon seasons inundating vast part of the valley. In this study19,728 sets of recorded hourly concurrent discharge data for six gauging sites have been collected from CWCs, Shillong, India for a period of six years pertaining to monsoon seasons (1st June 2000 to 15th October 2005).

Descriptions of the site locations and the statistical analysis of the observed data are shown in Table 1 and 2. The six gauging sites listed in Table 1 has a peak discharge of 4783.98, 1466.90, 3314.78, 120.59, 5409.30 and 5193.19 cubic meters respectively.

4.2 Data Pre-processing

In the literature there is no clear guidance available for divisions of data into training, validation and testing. Due to this majority of the researchers made arbitrary assumptions, although the divisions of data depend on the problems types, quantity of data and others. The recorded concurrent discharge data have been divided as 60% for training, 20% for cross validation and 20% for out-of sample. The same proportion is maintained for both MIMO-NARX and MIMO-PRNN.

For efficient network training the recorded discharge data have been normalized to simplify the network outliers. So normalization interval are range as [0 1] for sigmoidal functions and [-1 1] for hyperbolic tangent function (tanh). The gradient is much dependent on the derivatives of the activation functions. The derivatives of sigmoid functions can be defined as,

$$Q'(Z_k) = \tau Q(A_k)(1 - Q(Z_k)) \tag{16}$$

Where $Q(Z_k) = 1/(1+e^{-\tau Z_k})$, $Q(Z_k)$ is bounded to [0 1] but τZ_k has a range of $\pm\infty$. Similarly, in the case of hyperbolic tangent functions $G(Z_k) = \tanh(\tau Z_k)$, the derivatives of $G(Z_k)$ becomes,

$$G'(z_k) = \tau(1 - Q(Z_k))(1 + Q(Z_k)) \tag{17}$$

The term τZ_k for tanh has a range of $\pm\infty$ and $G(Z_k)$ is bounded to [-1 1]. Where τ is the steepness of the function and is known as gain factor for neuron which is normally equal to one. However, derivatives of tanh functions have maximum values equal to τ which correspond to steeper functions than that of sigmoid functions.

4.3 Performance Criteria

Statistical performance criterions like Root Mean Square Error (RMSE) and Mean Absolute Percentage Errors (MAPE) were used to determine prediction accuracy beyond network training data and coefficient of efficiency (CE) for models efficiency. RMSE determines how the network outputs fits the targets; MAPE check the models validations and its forecasting accuracy and CE examines perfect matches between the model discharge to the observed data which has a range confined to $[\infty\ 1]$; They are defined in Eqs. (18), (19) and (20) as,

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\hat{y}(n) - y(n)\right)^2} \tag{18}$$

$$CE = 1 - \frac{\sum_{n=1}^{N}\left(y(n) - \hat{y}(n)\right)^2}{\sum_{n=1}^{N}\left(y(n) - \bar{y}(n)\right)^2} \tag{19}$$

$$MAPE = \frac{100}{N}\sum_{1}^{N}\left(\frac{\left|y(n) - \hat{y}(n)\right|}{\left|y(n)\right|}\right) \tag{20}$$

Where $y(n)$ is the observed/actual flow at time n, $\hat{y}(n)$ is the simulated/forecasted flow at time n, $\bar{y}(n)$ mean of actual flow $\bar{y}(n)$ mean of forecasted flow and N the total number of observations.

4.4 Results and Discussions

The training algorithm discussed in section 2.1 and 3.1 is used

Table 2. Statistical Analysis for Actual Data on All Six Gauge Stations

Gauge Stations	Training				Testing			
	Min (m^3/s)	Max (m^3/s)	Mean (m^3/s)	St. dv (m^3/s)	Min (m^3/s)	Max (m^3/s)	Mean (m^3/s)	St. dv (m^3/s)
Badarpurghat	378.999	4114.756	2160.372	824.951	449.61	4783.981	2054.469	971.112
Matijuri	29.056	1504.194	361.843	285.266	32.412	1523.095	333.332	298.121
Annapurnaghat	157.612	2830.769	1284.779	547.080	126.074	3331.208	1260.056	622.047
Dholai	6.262	194.303	37.527	27.746	6.764	184.399	39.151	29.526
Maniarkhal	308.377	4663.100	2206.603	889.485	249.974	5434.511	2163.571	1006.613
Fulertal	105.386	4220.955	1428.319	843.548	185.127	5662.626	1454.312	937.134

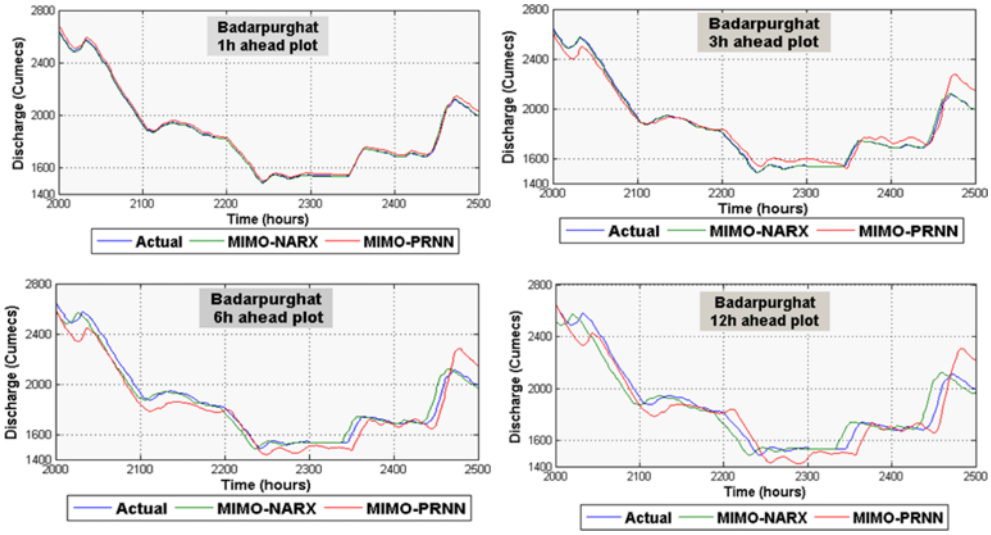

Fig. 5. Actual Versus Predicted Flows at Badarpurghat for 1-h, 3-h, 6-h and 12-h Ahead for Both MIMO-PRNN and MIMO-NARX

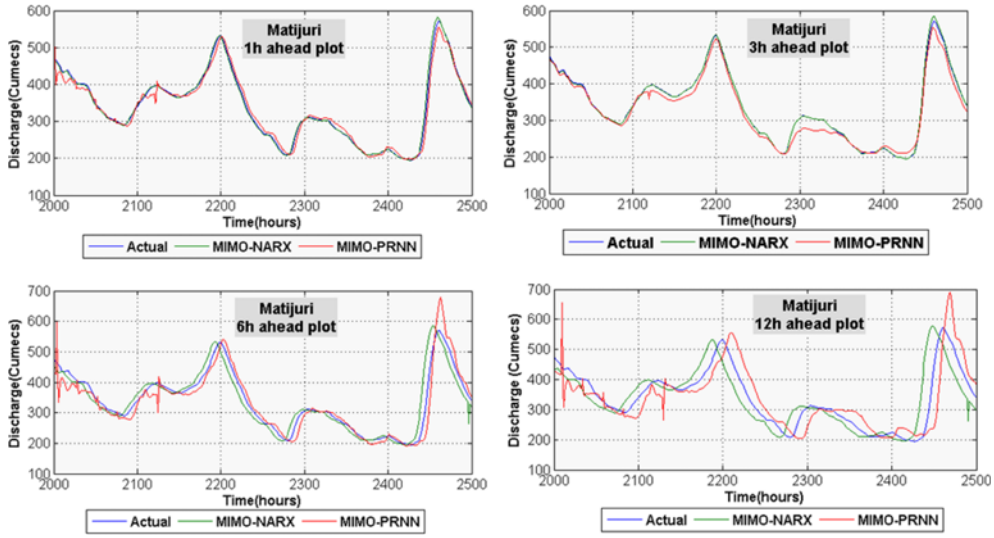

Fig. 6. Actual Versus Predicted Flows at Matijuri for 1-h, 3-h, 6-h and 12-h Ahead for Both MIMO-PRNN and MIMO-NARX

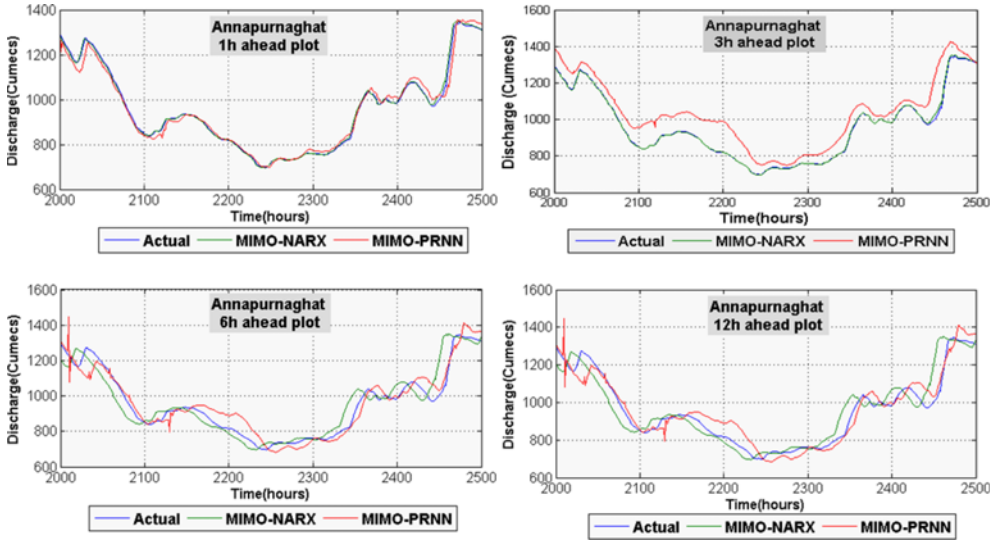

Fig. 7. Actual Versus Predicted Flows at Annapurnaghat for 1-h, 3-h, 6-h and 12-h Ahead for Both MIMO-PRNN and MIMO-NARX

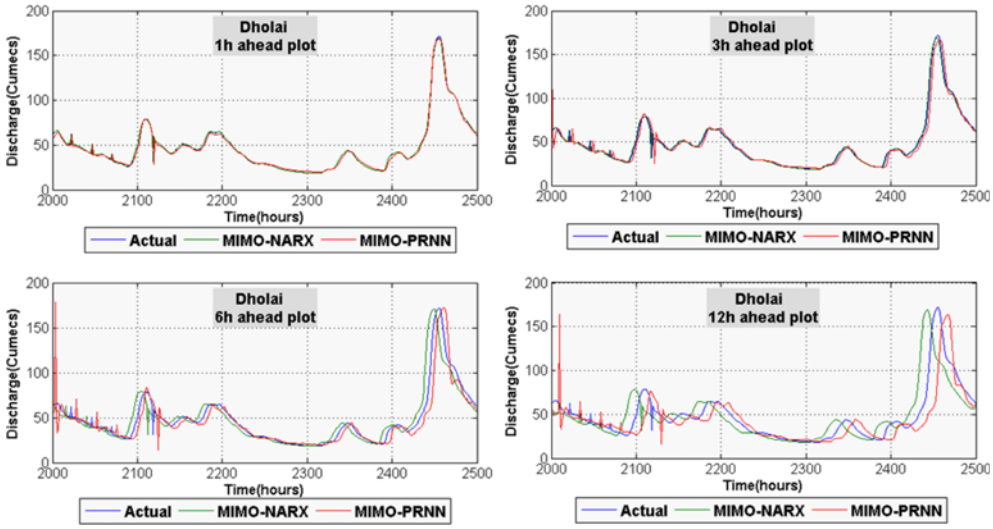

Fig. 8. Actual Versus Predicted Flows at Dholai for 1-h, 3-h, 6-h and 12-h Ahead for Both MIMO-PRNN and MIMO-NARX

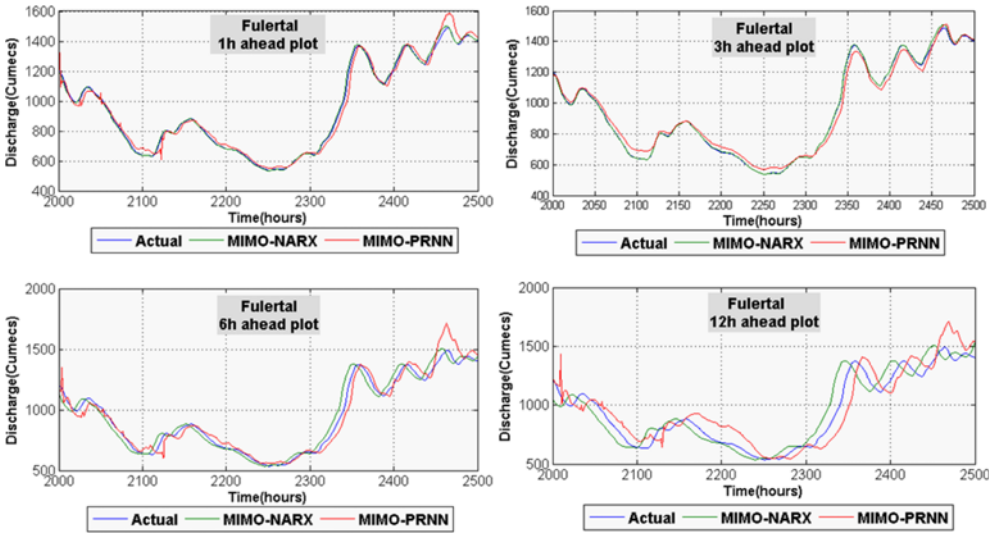

Fig. 9. Actual Versus Predicted Flows at Fulertal for 1-h, 3-h, 6-h and 12-h Ahead for Both MIMO-PRNN and MIMO-NARX; Zoom in with Corresponding Time Series Displays in fig. 8 and for the Same Purpose Contain in it

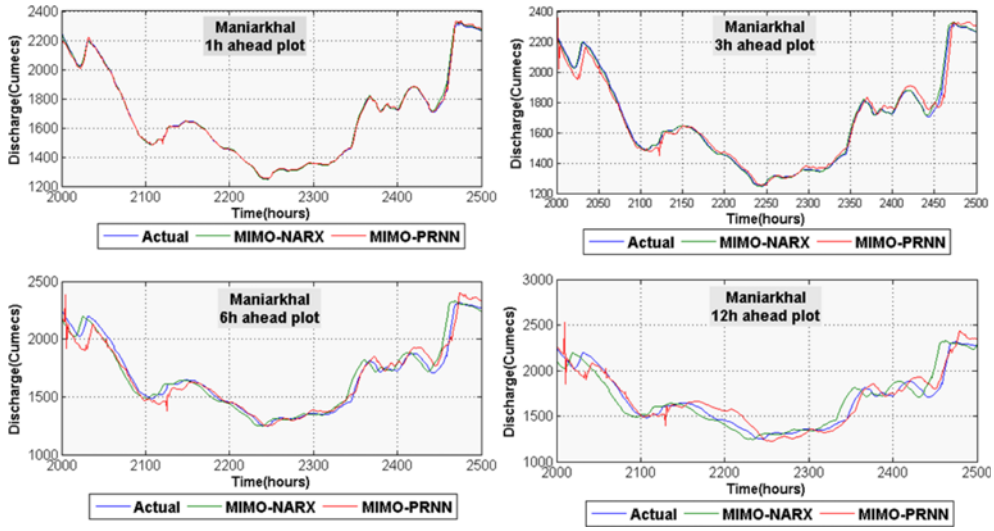

Fig. 10. Actual Versus Predicted Flows at Maniarkhal for 1-h, 3-h, 6-h and 12-h Ahead for Both MIMO-PRNN and MIMO-NARX

to train MIMO-NARX and MIMO-PRNN and has been used for single step (1 hour) and multi-steps (3 hours, 6 hours and 12 hours) ahead forecast. Multi-steps ahead forecasting is characterized by prediction using next values of time series h, where $h>1$ being the prediction horizon for given time series. Figs. 5 to 10 shows the plots for 1 hour, 3 hour, 6 hour and 12 hour at each forecasting stations. To validate the models accuracy, the predicted values has been compared with the actual (observed) values. Table 3 and 4 shows the result of the network performances of proposed statistical criterions to justify network accuracy. The RMSE and CE values are presented in Table 3 and Table 4 shows the MAPE. Table 3 and 4 shows that the values of performance criterions are within the acceptable limits for both the models, the CE is close to unity and the maximum RMSE 118 m^3/s. However, it is observed that MIMO-NARX consistently provides better performances for every lead time compared to MIMO-PRNN.

The networks have been designed by using trial and error methods. For better network generalization large step sizes have been considered from the beginning rather than random values. The step size in the range of 0.9 - 0.01 with an interval of 0.01 have been used. The steps sizes of 0.3 and 0.1 are found to be satisfactory for MIMO-PRNN and MIMO-NARX. Network hidden units have been varied from 2 to 50 with a fixed momentum term of 0.7, as the change in momentum term doesn't affect the network performance. For MIMO-PRNN, 24 numbers of PEs with 5000 iterations and the input delayed by 12 samples provides optimal values single step and multi-steps ahead forecasting. MIMO-NARX provides optimal results with 10 numbers of PEs in the hidden layer for both on single and multi-steps ahead forecasting. The optimal number of iterations for 1h, 3h, 6h and 12h lead time forecasting are 252, 267, 277, and 605 respectively for MIMO-NARX whereas MIMO-PRNN required large iterations stating 5000, 7000, 10,000 and 25,000. The input delayed by 1 and 2 samples provides satisfactory results for MIMO-PRNN and MIMO-NARX for each lead time.

5. Conclusions

The research work reported in this study demonstrates the applicability of MIMO model using PRNN and NARX in Barak river system, Assam, India. The statistical performance criterion (Table 3) shows that both MIMO-PRNN and MIMO-NARX are applicable for simultaneous flow forecasting in a river system. MIMO-NARX proves to be superior to MIMO-PRNN as MIMO-NARX does not suffer from any computational loss. It is also observed that with the increase in input delays for both MIMO-PRNN and MIMO-NARX, the performance of MIMO-NARX degrade in a softer fashion. However, the computational

Table 3. Comparison of Performances Indices for Both MIMO-NARX and MIMO-PRNN

Lead Time (hr.)	Forecasted stations	MIMO-NARX				MIMO-PRNN			
		Training		Testing		Training		Testing	
		RMSE (m^3/s)	CE	RMSE (m^3/s)	CE	RMSE (m^3/s)	CE	RMSE (m^3/s)	CE
1	Badarpurghat	10.0037	0.9990	11.6999	0.9986	27.0124	0.9984	30.9107	0.9982
	Annapurnaghat	12.7579	0.9986	9.9605	0.9989	17.0931	0.9967	18.6417	0.9981
	Fulertal	14.0346	0.9976	13.3948	0.9964	37.2108	0.9957	45.5238	0.9955
	Matijuri	5.6704	0.9690	6.9865	0.9535	25.9292	0.9220	23.3861	0.9827
	Dholai	4.4787	0.9905	4.2365	0.9858	6.1223	0.9916	4.7974	0.9700
	Maniarkhal	18.2757	0.9934	20.4837	0.9901	36.4626	0.9829	33.4591	0.9978
3	Badarpurghat	15.8753	0.9898	13.0629	0.9908	66.6482	0.9727	63.3540	0.9923
	Annapurnaghat	25.0541	0.9785	27.8681	0.9678	36.8588	0.9453	37.9823	0.9922
	Fulertal	27.5613	0.9932	27.8295	0.9928	47.0630	0.9981	43.4901	0.9959
	Matijuri	11.1356	0.9903	12.1986	0.9855	17.3161	0.9955	20.6185	0.9865
	Dholai	8.7953	0.9673	9.2288	0.9510	7.5939	0.9827	5.8062	0.9560
	Maniarkhal	35.8899	0.9875	34.6904	0.9883	66.4190	0.9700	63.6839	0.9920
6	Badarpurghat	28.6351	0.9987	29.0956	0.9980	86.6823	0.9978	89.3659	0.9847
	Annapurnaghat	36.5189	0.9968	36.0617	0.9952	95.5637	0.9923	52.1464	0.9854
	Fulertal	40.1734	0.9959	42.8053	0.9939	95.8560	0.9922	80.1182	0.9861
	Matijuri	16.2313	0.9897	17.2433	0.9846	34.6848	0.9865	37.2406	0.9560
	Dholai	12.8201	0.9753	11.2325	0.9830	7.8684	0.9560	8.8918	0.8968
	Maniarkhal	52.3133	0.9966	51.5745	0.9979	83.6250	0.9920	86.4225	0.9853
12	Badarpurghat	37.6249	0.9923	36.1194	0.9935	103.7433	0.9847	114.3392	0.9749
	Annapurnaghat	47.9837	0.9925	49.9558	0.9888	71.0240	0.9854	72.3828	0.9719
	Fulertal	52.7856	0.9940	50.8972	0.9951	138.3073	0.9861	136.8254	0.9594
	Matijuri	21.3270	0.9808	20.9317	0.9813	75.4438	0.9560	74.0109	0.8263
	Dholai	16.8449	0.9509	18.7897	0.9864	12.5779	0.8968	13.6495	0.7570
	Maniarkhal	68.7366	0.9827	68.0983	0.9740	119.6445	0.9560	118.4754	0.9725

Table 4. Performance Comparison Based on MAPE Forecasting on Tested Data Along with Corresponding Peak and Lag Difference for Both MIMO-PRNN and MIMO-NARX at Different Lead Times

Lead Time (hr.)	Forecasted Stations	MIMO-NARX			MIMO-PRNN		
		MAPE forecasting (%)	Peak Q (m^3/s)	*Lag diff. (hr.)	MAPE forecasting (%)	Peak Q (m^3/s)	*Lag diff. (hr.)
1	Badarpurghat	1.86	2630	60	3.74	2565	5
	Annapurnaghat	1.90	1290	15	3.13	1270	5
	Fulertal	1.15	1580	90	5.84	1500	10
	Matijuri	1.28	555	15	8.22	580	10
	Dholai	1.95	168.8	2	5.51	170	0.8
	Maniarkhal	1.86	2222	22	4.62	2196	4
3	Badarpurghat	1.64	2130.5	20	8.53	2120.5	10
	Annapurnaghat	1.79	1077	122	4.44	1078	121
	Fulertal	1.61	1340	36	6.08	1375	1
	Matijuri	1.91	555	17	4.07	585	13
	Dholai	3.06	168	5	8.23	169	4
	Maniarkhal	1.85	1909	31	4.33	1880	2
6	Badarpurghat	1.11	2162	44	4.63	2121	3
	Annapurnaghat	1.82	1410	65	5.63	1350	5
	Fulertal	2.00	1700	202	6.57	1500	2
	Matijuri	3.33	670	100	5.03	576	6
	Dholai	4.53	70	3	10.81	71	2
	Maniarkhal	1.70	2400	102	4.29	2300	2
12	Badarpurghat	3.53	2601	251	11.07	2352	2
	Annapurnaghat	2.76	900	75	6.53	778	47
	Fulertal	5.01	2170	5	8.75	2180	5
	Matijuri	9.18	555	25	11.33	529	1
	Dholai	10.92	164	7.8	14.32	168.9	2.9
	Maniarkhal	2.82	2410	110	5.65	2310	10

*Lag Diff. (hr.) implies the difference of corresponding lag time of peak flows from the actual flows

time becomes lengthy for MIMO-NARX when dealing with the large numbers of input sequences. Besides, the results also demonstrate the efficacy of both MIMO-NARX and MIMO-PRNN for short range multi-steps ahead forecast. The improved performance for MIMO-NARX using MIMO model for simultaneous flow forecasting at higher lead times encourages its applicability in case of many other complex nonlinear time series.

References

Adeloye, A. J. and Munari, A. D. (2006). "Artificial neural network based generalized storage-yield-reliability models using the Levenberg-Marquardt algorithm." *Journal of Hydrology*, Vol. 326, Nos. 1-4, pp. 215-230, DOI: 10.1016/j.jhydrol.2005.10.033.

Bontempi, G. and Taieb, S. B. (2010). "Conditionally dependent strategies for multiple-step-ahead prediction in local learning." *International Journal of Forecasting*. In Press, Corrected Proof :{ 2011. ISSN 0169-2070, DOI: 10.1016/j.ijforecast.2010.09.004.

Chau, K., Wu, C., and Li, Y. (2005). "Comparison of several flood forecasting models in yangtze river." *J. Hydrol. Eng*., Vol. 10, No. 6, pp. 485-491, URL, DOI: 10.1061/(ASCE)1084-0699(2005)10:6(485).

Chang, F. J., Tsai, Y. H., Chen, P. A., Coynel, A., and Vachaud, G. (2015). "Modeling water quality in an urban river using hydrological factors - Data driven approaches." *J. Environ. Manage*, Vol. 151, pp. 87-96, URL: DOI: 10.1016/j.jenvman.2014.12.014.

Chang, P. C., Wang, D., and Zhou, C. (2012). "A novel model by evolving partially connected neural network for stock price trend forecasting." *Expert Systems with Applications,* Vol. 39, pp. 611-620, URL: DOI: 10.1016/j.eswa.2011.07.051.

Choudhury, P. and Roy, P. (2015). "Forecasting concurrent flows in a river system using ANNs." *Journal of Hydrologic Engineering*, Vol. 20, No. 8, 06014012, DOI: 10.1061/(ASCE)HE.1943-5584.0001107.

Claveria, O., Monte, E., and Torra, S. (2015). "Multiple-input multiple-output vs. single-input single-output neural network forecasting." Research Institute of Applied Economics www.ub-irea.com. URL: http://www.ub.edu/irea/working_papers/2015/201502.pdf.

Cigizoglu, H. K. and Alp, M. (2006). "Generalized regression neural network in modelling river sediment yield." *Advances in Engineering Software*, Vol. 37, Issue 2, pp 63-68, DOI: 10.1016/j.advengsoft.2005.05.002.

Coulibaly, P., Anctil, F., and Bobée, B. (2001). "Multivariate reservoir inflow forecasting using temporal Neural Networks." *J. Hydrol. Eng.,* DOI: 10.1061/(ASCE)1084-0699(2001)6:5(367), pp. 367-376.

Coulibaly, P. and Evora, N. D. (2007). "Comparison of neural network methods for infilling missing daily weather records." *Journal of Hydrology*, Vol. 341, Issues 1-2, pp. 27-41, DOI: 10.1016/j.jhydrol.2007.04.020.

Dibike, Y. B. and Solomatine, D. P. (2001). "River flow forecasting

using artificial Neural Networks." *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere*, Vol. 26, Issue 1, pp. 1-7, DOI: 10.1016/S1464-1909(01)85005-X.

Diaconescu, E. (2008). "The use of NARX neural networks to predict chaotic time series." *WSEAS Transactions on Computer Research*, Vol. 3, No. 3, ISSN: 1991-8755.

Elizondo, D., Fiesler, D., and Korczak, J. (1997), "A Survey of Partially Connected Neural Networks." *Int J Neural Syst,* Vol. 8, Nos. 5-6, pp. 535-558, URL: http://www.korczak-leliwa.pl/files/papers/9-Elizondo-Fiesler-Korczak.pdf.

Horne, B. G. and Giles, C. L. (1995). "An experimental comparison of recurrent Neural Networks." *Advances in Neural Information Processing Systems* 7, URL: https://clgiles.ist.psu.edu/papers/NIPS94.rnn.comp.pdf.

Karunanithi, N., Grenney, W., Whitley, D., and Bovee, K. (1994). "Neural networks for river flow prediction." *J. Comput. Civ. Eng*., DOI: 10.1061/(ASCE)0887-3801(1994)8:2(201), 201-220.

Kagoda, P. A., Ndiritu, J., Ntuli, C., and Mwaka, B. (2010). "Application of radial basis function neural network to short term stream flow forecasting." *Physics and Chemistry of the Earth Parts A/B/C,* Vol. 35, No. 13, pp. 571-581, DOI: 10.1016/j.pce.2010.07.021.

Kişi, Ö. (2008). "Stream flow forecasting using neuro-wavelet technique." Vol. 22, Issue 20, pp. 4142-4152, DOI: 10.1002/hyp.7014.

Levine, S. and Koltun, V. (2014). "Learning complex neural network policies with trajectory optimization." *Proceedings of the 31st International Conference on Machine Learning, Beijing*, China. JMLR: W&CP Vol. 32.

Lin, T. N., Giles, C. L., Horne, B. G., and Kung, S. Y. (1997). "A delay damage model selection algorithm for NARX Neural Networks." *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2719-2730, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=650098.

Marquardt, D.W. (1963). "An algorithm for least-squares estimation of nonlinear parameters." *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, pp. 431-441.

Maciel, L., Gomide, F., and Ballini, R. (2012). "MIMO Evolving functional fuzzy models for interest rate forecasting." *Computational Intelligence for Financial Engineering & Economics (CIFEr)*, IEEE Conference ISBN:978-1-4673-1802-0, DOI: 10.1109/CIFEr.2012.6327781.

Mansouri, V. and Mohammad E. Akbari, M. E. (2014) "Efficient short-term electricity load forecasting using recurrent Neural Networks." *Journal of Artificial Intelligence in Electrical Engineering*, Vol. 3, No. 9, pp. 46-54, URL: http://jaiee.iau-ahar.ac.ir/article_514345_2bccca8b82b63e8aeb22051dd5ecd26f.pdf.

Navarra, M. M., Lessmann, S., and Vob, S. (2015). "Sales forecasting with partial recurrent neural Networks: Empirical insights and benchmarking results." *48th Hawaii International Conference on System Sciences*, pp. 1108-1116, DOI: 10.1109/HICSS.2015.135.

Perera, E. D. P. and Lahat, L. (2014). "Fuzzy logic based flood forecasting model for the Kelantan River basin, Malaysia." URL: Journal of Hydro-environment Research, pp. 1-12, DOI: 10.1016/j.jher.2014.12.001.

Robinson, T. and Fallside, F. (1991). "A Recurrent error propagation network speech recognition system." *Computer Speech and Language*, Vol. 5, No. 3, pp. 259-274, DOI: 10.1016/0885-2308(91)90010-N.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation, In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations. Cambridge: MIT Press/Bradford Books.

Siegelmann, H. T., Horne, B. G., and Giles, C. L. (1997). "Computational capabilities of recurrent NARX Neural Networks." *IEEE Trans. Syst., Man Cybern*., pt. B, Vol. 27, p. 208.

Trajkovic, S., Todorovic, B., and Stankovic, M. (2003). "Forecasting of Reference Evapotranspiration by Artificial Neural Networks." *J. Irrig. Drain Eng.*, DOI: 10.1061/(ASCE)0733-9437(2003)129:6(454), pp. 454-457.

Thirumalaiah, K. and Deo, M. (1998). "River Stage Forecasting Using Artificial Neural Networks." *J. Hydrol. Eng.*, DOI: 10.1061/(ASCE)1084-0699(1998)3:1(26), pp. 26-32.

Wang, Y. M., Traore, S., and Kerh, T. (2008). "Neural network approach for estimating reference evapotranspiration from limited climatic data in Burkina Faso." *Journal WSEAS Transactions on Computers*, Vol. 7 Issue 6, pp. 704-713.

Wang, Y. M. and Seydou, T. (2009). "Time-lagged recurrent network for forecasting episodic event suspended sediment load in typhoon prone area." *International Journal of Physical Sciences,* Vol. 4, No. 9, pp. 519-528, URL: https://www.researchgate.net/publication/237136821.

Weigend, A. S. and Rumelhart, D. E. (1992). Generalization through minimal networks with application to forecasting, In: Keramidas, E. M., editor, INTERFACE'91 - 23rd Symposium on the Interface: Computing Science and Statistics, pp. 362-370.

Williams, R. J. and Zipser, D. (1995). "Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity." pp. 433-486, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.5418.

Williams, R. J. and Zipser, D. (1989). "Experimental analysis of the real-time recurrent learning algorithm." *Connection Science*, Vol. 1, No. 1, pp. 87-111, DOI: 10.1080/09540098908915631.

Yiu, J. C. M. and Wang, S. (2007). "Multiple ARMAX modeling scheme for forecasting air conditioning system performance." *Energy Conversion and Management,* Vol. 48, pp. 2276-2285, URL: DOI: 10.1016/j.enconman.2007.03.018.