Application of RBFNNs Incorporating MIMO Processes for Simultaneous River Flow Forecasting

Joseph Tripura*, Parthajit Roy & Abdul Karim Barbhuiya

Department of Civil Engineering, National Institute of Technology Silchar, Silchar 788010, India
*E-mail: tripurajoseph89@gmail.com

Abstract. Simultaneous flow forecasting using multi-input multi-output (MIMO) processes is an efficient technique for accurate flow forecasting on river systems. The present study demonstrates the capability of radial basis function neural networks (RBFNN) incorporating MIMO processes in simultaneous river flow forecasting. The river system considered in the present study was the Barak river system, Assam, India. Hourly concurrent discharge data were collected from the Central Water Commission, Shillong, India from multiple sections of the Barak river system. The forecasts were tested for short-range time horizons, i.e. 1, 3, 6 and 12 hours in advance, and a comparative analysis was done using the popular Nonlinear Autoregressive with Exogenous Inputs (NARX) time series model. The result shows that MIMO-NARX provided higher prediction accuracy than MIMO-RBFNN, even at longer lead times when compared to following various statistical criterions.

Keywords: *radial basis function neural networks (BFNNs); Nonlinear Autoregressive with Exogenous Inputs (NARX); direct MIMO process; simultaneous forecasting.*

1 Introduction

A river system comprises of a network of connected tributaries, where flooding in any of the tributaries may change the flow conditions of the main river. Together with other meteorological circumstances, this can cause a rise in the water level of the major river channel for a given watershed. Such conditions frequently arise when part of a contributing tributary receives heavy precipitation upstream, thereby feeding flood-like waves into the main river channel. The present study was aimed at monitoring and observing this effect on the Barak river system in the Barak Valley watershed, Assam, India. The forecast was considered simultaneously at multiple sections of the river system using multi-input multi-output (MIMO) processes. In general, researchers have focused mostly on single-input single-output (SISO) processes and multi-input single-output (MISO) processes without extending the weightage to MIMO processes. For instance, a number of model structure selection methods have been derived for SISO nonlinear models in [1-3] and MISO processes in [4-10], most of which can be extended to MIMO processes. MIMO processes have

Received May 31st, 2017, 1st Revision November 15th, 2017, 2nd Revision March 12th, 2018, Accepted for publication August 31st, 2018.
 ISSN: 2337-5779, DOI: 10.5614/j.eng.technol.sci.2018.50.3.9

been applied successfully in many different fields [11-14], but Choudhury and Roy [15] observed their limited use in dealing with hydrological problems. A MIMO process can be described as studying the interrelations of variables between inputs and outputs simultaneously. There are two types of MIMO processes: (i) multi-stage MIMO processes and (ii) direct (or single-stage) MIMO processes. A multi-stage MIMO process is based on prediction horizons where multi-step outputs at different time horizons are considered and learning is attempted by one multiple-output dependency. The model can be computed [16] as:

$$y_{n+1}, y_{n+2}, .., y_{n+k} = f(y_n, \dots\dots, y_{n-d+1}) + w \tag{1}$$

where $y_{n+(.)}$ and $y(.)$ represent the multi-output and multi-input terms in one cycle and k is an integer $k > 1$. The term w is the vector noise term for zero mean and non-diagonal covariance, d is the maximum embedded order. The estimation of the next values for k after the learning process is done with Eq. 2:

$$\hat{y}_{n+1}, \dots\dots\dots, \hat{y}_{n+k} = \hat{f}(y_n, \dots\dots\dots, y_{n-d+1}) \tag{2}$$

This type of MIMO process may suffer from model flexibility because it constrains all prediction horizons within the same model structure. On the other hand, direct MIMO processes are based on model outputs where at one phase the numbers of multiple outputs are considered in a single-step prediction and are direct for different time horizons. This is similar to the direct-step process (a kind of MISO), except that the model's outputs considered are of multiple series. Direct MIMO processes can be computed using Eq. (3):

$$y(n+h) = f\begin{cases} y(n-d_y), \dots\dots\dots, y(n-1), y(n); \\ u(n-d_u), \dots\dots\dots, u(n-1), u(n) \end{cases} \tag{3}$$

where, $u(n) = [u_1(n), \dots\dots\dots, u_r(n)]^T$ and $\hat{y}(n) = [\hat{y}_1(n), \dots\dots\dots, \hat{y}_r(n)]^T$ are the input and output vectors, d_y and d_u are the time lags of input and output, h is the prediction time horizon, and $f(n) = [f_1(n), \dots\dots\dots, f_r(n)]^T$ denotes the nonlinear relation to be estimated. What is common in both methods is that they share the returned prediction of the vector time series.

In this paper, a direct MIMO process is proposed because capturing the temporal stochastic dependencies of the historical data can be used to predict a time series. This direct MIMO process can be incorporated in RBFNNs, where the output node assigns a weight value to each of the radial basis function neurons, and the total response is the multiplication of this weight with the neuron's activation. RBFNNs have been widely used in solving many different problems, also by inclusion of a hybridization state. The use of RBFNs dates back to 1985 [17] and has been exploited in the design of neural networks since 1988 [18]. Popular applications of RBFNNs include function approximation,

control, classification, curve fitting, and time series problems [19]. RBFNNs are universal approximators and are well-suited for function approximation problems [20] and they are faster in the training phase compared to other neural networks due to their simpler network architecture. RBFNNs provide better results than multilayer perceptrons (MLP) and functional link artificial neural networks (FLANN) in stock market prediction [21]. These and other advantages of RBFNNs, including better generalization capability, faster convergence, low error extrapolations and better reliability over chaotic data, make this type of neural network very suitable to deal with similar problems.

Similarly, direct MIMO process can also be incorporated in a NARX model, as it can predict past values of the same series and current and past values of the driving series. NARX is one of the typical paradigms of complex recurrent models followed by a state space model and a recurrent multilayer perceptron (RMLP). It was introduced in 1985 [22] and was first applied in the context of neural networks in 1990 [23]. Nonlinear dynamic systems, especially time series, can be appropriately modeled by NARX [24]. NARX can retain information as long as conventional recurrent networks [25] and performs better than conventional recurrent neural networks [26]. NARX is better suited for use with a gradient-descent learning algorithm compared to other recurrent architectures [27]. The network also provides faster convergences with better generalization than other networks. Recently, NARX networks have been applied for solving non-linear problems in various fields other than hydrological applications with remarkable results [28-30]. In the present study, the results of simultaneous forecasting on a river system obtained with NARX were used for comparison with the results obtained with RBFNN. The applicability of direct MIMO processes for simultaneous flow forecasting on multiple sections of a river system using RBFNN was investigated. We considered a multi-step ahead forecasting scheme for short-range prediction horizons by using the model form, because an accurate short-range prediction is sufficient to understand the dynamics and characteristic behavior of such a system.

This paper is organized as follows: Section 2 and its subsections describe the RBFNN network topology and its MIMO process form. Section 3 and its subsections describe NARX networks and their MIMO process form. Section 4 describes the model's application. Section 5 defines the performance criterion used in this study. The results and their discussion are given in Section 6. Finally, Section 7 presents our concluding remarks.

2 RBFNN Networks Topology

A typical RBFNN model generally consists of three layers: an input layer, one or two hidden layers, and an output layer containing one or more nodes. The

network topology of an RBFN network, as depicted in Figure 1, is a MIMO-RBFNN structure, consisting of more than one output node and is characterized by the following aspects: firstly, the input layer is the receptor of the input source, secondly, the middle layer, or the hidden layer, which contains sets of nodes computing a symmetric radial function, provides the nonlinear transformation of the input space to an intermediate space, and, thirdly, the output layer, which contains the output node source, linearly combines the outputs of the hidden layer. Although there are many basis functions available in the literature, this study used a Gaussian function as it can be localized easily, although strictly speaking this is not the main reason. The Gaussian function can be defined as in Eqs. (4) & (5):

$$\emptyset(r) = exp\left(-\frac{r^2}{2\sigma_i^2}\right) \qquad \sigma > 0 \tag{4}$$

$$or, \emptyset(r) = exp\left(-\frac{\|u-\mu_i\|^2}{2\sigma_i^2}\right) \tag{5}$$

This implies that if $r \to \infty$ then $\emptyset(r) \to 0$, where $r = \|u - \mu_i\|$ is the Euclidean distance of the Gaussian, and u and μ are the input and center.

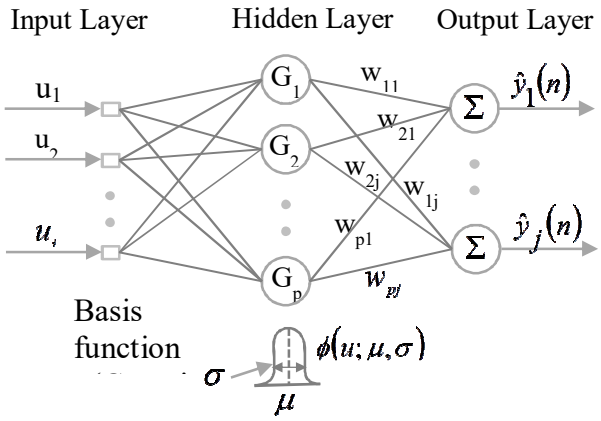

Figure 1 Structure of the radial basis function.

2.1 RBFNNs and its MIMO process form

The MIMO form of RBFNN can be expressed in Eq. (6) as follow:

$$\hat{y}_k(n+1) = \sum_{i=1}^{L} w_{ij} exp\left(-\frac{\|u-\mu_i\|^2}{2\sigma_i^2}\right) \tag{6}$$

where $\hat{y}(n) = [\hat{y}_1(n), \ldots\ldots\ldots, \hat{y}_r(n)]^T$ and $u(n) = [u_1(n), \ldots\ldots\ldots, u_r(n)]^T$ are the system output and input respectively, $exp(.)$ is the Gaussian function, w_{ij} are the weights, μ_i and σ_i are the center and width of the Gaussian, L is the number of nodes in the hidden layer, respectively. The key problem in finding an appropriate solution for the given architecture is to teach the RBFNN, which will possibly be challenging for different problems. The identification problem of MIMO-RBFNN includes finding centers μ_i of the activation functions, spread σ_i of the activation functions, and finally weights w_{ij} from the hidden layer to the output layer. To sort out this problem, a modified backpropagation (M-BP) algorithm was applied in this study, as discussed in the next subsection (2.2).

2.2 Training MIMO-RBFNN

Generally, an RBFNN is trained using two learning modes, unsupervised and supervised. The unsupervised learning mode is implemented between the input layer and the hidden layer and the supervised learning mode is implemented between the hidden layer and the output layer. As mentioned above, this study used a modified backpropagation (M-BP) algorithm to train MIMO-RBFNN. The activation function used is the Gaussian function. Like simple backpropagation, M-BP first initializes the network and forwardpasses the training input-output pairs and computes the network output for every layer and then backpasses the following:

$$\frac{\partial E}{\partial w'}\frac{\partial E}{\partial \mu'}\frac{\partial E}{\partial \sigma^2} \tag{7}$$

Here E is the cost function or error function to be optimized. The parameters are then updated as:

$$w_{pj}(n+1) = w_{pj}(n) + \Delta w_{pj}(n+1) \tag{8}$$

$$\text{and}, \Delta w_{pj}(n+1) = l_3 \delta_p y_j \tag{9}$$

where, $w_{pj}(n)$ is the weight connecting layer p to j at time n, $\Delta w_{pj}(n)$ is the weight change, l_3 is the learning rate, δ_p is the error at p, y_j is the actual output at j.

$$\mu_{ji}(n+1) = \mu_{ji}(n) + \Delta \mu_{ji}(n+1) \tag{10}$$

$$\text{where, } \Delta \mu_{ji}(n+1) = \frac{l_2 \delta_p w_{pj} y_j}{\sigma_p^2 (u_{ji} - \mu_{ji})} \tag{11}$$

Similarly, for the width (σ), the relation is defined as:

$$\sigma_j(n+1) = \sigma_j(n) + \Delta \sigma_j(n+1) \tag{12}$$

where, $\Delta\sigma_j(n+1) = l_1\delta_p w_{pj} y_j \left(\frac{(u_{pj}-\mu_j)^2}{2\sigma_j^4}\right)$ (13)

Here, $\mu_{ji}(n)$ and $\sigma_j(n)$ are the center and width connecting the corresponding subscripted layers at time n. Similarly, $\Delta\mu_{ji}(n)$ and $\Delta\sigma_j(n)$ are their corresponding centers and width changes, l_1 and l_2 are the learning rate factors, μ_{ji} and μ_{pj} are the inputs connected by the corresponding subscripted layers. This process is then repeated until convergence is attained.

3 NARX Network Topology

NARX is a dynamic network architecture with a recurrent structural design for nonlinear dynamic systems. Generally, it is used for input-output modeling issues in solving nonlinear time series problems. Figure 2 represents the typical structure of the NARX model used in this study, consisting of current inputs and previous sequences of inputs into the network. Additionally, it also comprises the delayed versions of the network outputs. Thus, in Figure 2 the embedded memory is shown where the first tapped delay line of order q receives the current inputs and its previous sequences. The second tapped delay line of order p receives the network outputs and its previous sequences through feedback connections.

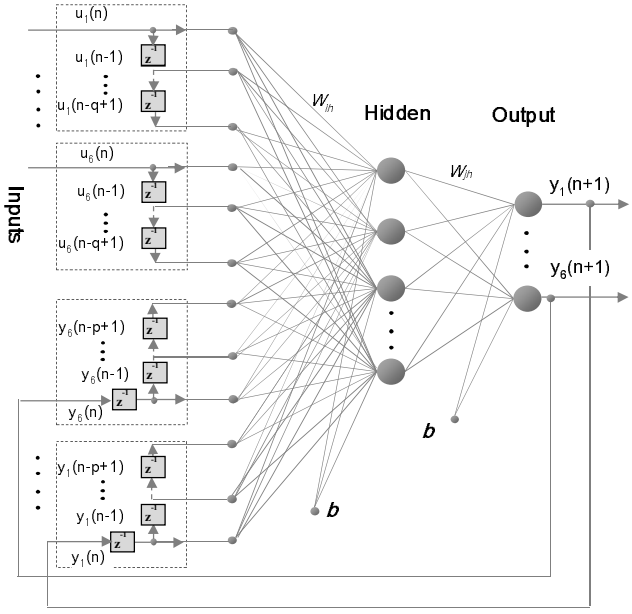

Figure 2 Typical NARX model from this study.

The SISO process [30] for a discrete-time nonlinear system for a NARX model can be represented by:

$$y(n+1) = \sigma\big[\big(y(n), y(n-1), \dots\dots\dots, y(n-p+1)\big); \\ u(n), u(n-1), \dots\dots\dots, u(n-q+1)\big] \quad (14)$$

where, $y(n+1)$ is the network predicted output, $u(n)y(n)$ are the input-output pairs at time n, and $q \geq p$, σ is the non-linear differentiable function $\left(\sigma: \Re^{p+q} \rightarrow \Re\right)$. Thus the output of the system at time $(n+1)$ depends on n past values $y_k(n-i)$ for $(i = 0,1,2, \dots\dots\dots, p-1)$. Similarly, for the q past values of the input $u(n-j)$ for $(j = 0,1,2, \dots\dots\dots, q-1)$.

The mapping function of NARX (or MIMO-NARX) is generally set by a multilayer perceptron (MLP). The total network parameters to be adjusted during the identification process are a summation of all input connection weights, all hidden-layer connected weights and the biases connected to both hidden and output nodes. The hidden layer output can be computed as:

$$\delta_i = \sigma\left[\sum_{j=1}^{\rho} w_{ij} u_j + b_i^l\right] \quad (15)$$

And the estimated model output is represented by:

$$y_k = \sigma\left[\sum_{i=1}^{l} w_{ki} h_i + b_i^l\right] \quad (16)$$

Here, σ denotes the size of the input weights and l represents the bias size/number of hidden layer nodes. For more information on MIMO-NARX, the reader may refer to Tripura and Roy [14].

4 Model Application and Discussion

The proposed models were applied to the Barak River system in the Barak Valley in the southern part of Assam, India. Figure 1 shows a map of the study area and Table 1 describes the six gauging stations at different locations of the Barak River system, naming its contributing tributaries. The river Barak originates from the Barail Range of Naga Hills at an altitude of about 2995 m and enters the plains of South Assam between 24°8' and 25°8' N latitude, and 92°15' and 93°15' E longitude. The recorded hourly flow data have a statistical observed mean value of 2160.372, 361.843, 1284.779, 37.527, 2206.603 and 1428.319 Cumec for the respective stations listed in Table 1. As for the CWCs, this observed statistical mean is sufficient to provide excess water deluging the floodplains, causing severe damage to crops and property in this valley almost every year during the monsoon season.

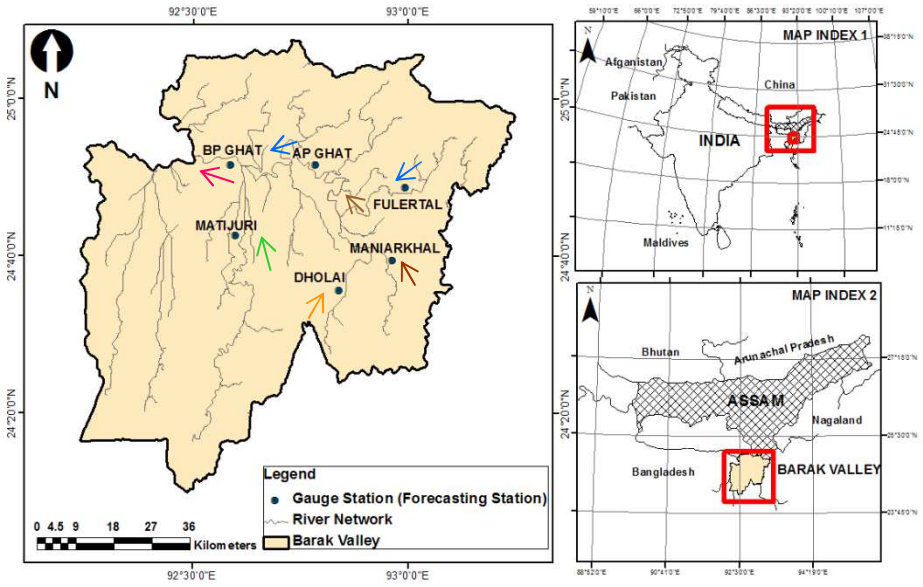

Figure 3 Barak river system and different forecasting stations.

Multiple configurations of both MIMO-NARX and MIMO-RBFNN were considered during the experiment, the best of which are reported. The experimental provision of data for conducting simultaneous flow forecasting in the river system were as follows: first 60% of the data is set for training, 20% for cross-validation and another 20% for out-of-sample testing. For efficient network training, the pre-processing steps known as normalization were done for inputs and targets, besides for the application of nonlinear activation functions (AFs). The method for normalization adopted here was the external normalization method, as it provides better performance in solving time series problems [31].

Thus, depending on the types of AFs used, the interval was a range of either [0 1], [-1 1], etc. to simplify network outliers. Two AFs tested were sigmoid and hyperbolic tangent (*tanh*) functions. The derivatives of the AFs are important when the gradients are very dependent. The derivatives of the sigmoid functions can be defined as:

$$Q'(Z_k) = \tau Q(Z_k)\big(1 - Q(Z_k)\big) \tag{17}$$

where, $Q(Z_k) = \frac{1}{(1+e^{-\tau Z_k})}$ and $Q(Z_k)$ is bounded to [0 1] but τZ_k has a range of $\pm\infty$. Similarly, in the case of tan hyperbolic functions $G(Z_k) = tanh(\tau Z_k)$, the derivative of $G(Z_k)$ is:

$$G'(Z_k) = \tau\big(1 - Q(Z_k)\big)\big(1 + Q(Z_k)\big) \tag{18}$$

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

Table 1 Description of site locations for all gauge stations (forecasting stations).

Station Code	Station Number	Gauge Station	Location (River Name)	Remarks*
01-11-01-007	1	Badharpur Ghat (BG)	Barak	Major
01-11-01-003	2	Matijuri (MJ)	Katakhal	Tributary
01-11-01-008	3	Annapurna Ghat (AG)	Barak	Major
01-11-06-004	4	Dholai (DL)	Rukni	Tributary
01-11-01-005	5	Maniarkhal (MK)	Sonai	Tributary
01-11-01-009	6	Fulertal (FT)	Barak	Major

**Major:* gauge station at main river site; *tributary:* gauge station at tributary site.

5 Performance Criteria

Statistical criteria such as RMSE (root mean square error), *r* (correlation coefficient), CE (coefficient of efficiency), MAPE (mean absolute percentage error) were used to investigate network performance. These are considered to achieve prediction accuracy beyond network training. In addition to these criteria, PFC (peak flow criteria) were also used, as they provides more accurate peak values than RMSE during flood periods. A PFC value equal to zero represents a perfect fit. The statistical performance criteria were defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\big(\hat{y}(n) - y(n)\big)^2} \tag{19}$$

$$r = \frac{\Sigma\Big(\big(\hat{y}(n) - \hat{y}_{avg}(n)\big)\big(y(n) - y_{avg}(n)\big)\Big)}{\sqrt{\Sigma\Big(\big(\hat{y}(n) - \hat{y}_{avg}(n)\big)^2\big(\hat{y}(n) - \hat{y}_{avg}(n)\big)^2\Big)^2}} \tag{20}$$

$$CE = 1 - \left(\sum_{n=1}^{N}\big(y(n) - \hat{y}(n)\big)^2 \Big/ \sum_{n=1}^{N}\Big(y(n) - y_{avg}(n)\Big)^2\right) \tag{21}$$

$$MAPE = \frac{100}{N}\sum_{1}^{N}\left(\frac{|y(n) - y_{avg}(n)|}{|y(n)|}\right) \tag{22}$$

$$PFC = \frac{\left(\sum_{n=1}^{N_p}(y(n)-\hat{y}(n))^2(y(n))^2\right)^{1/4}}{\left(\sum_{n=1}^{N_p}(y(n))^2\right)^{1/2}} \quad (23)$$

where, $y(n)$ is the observed flow at time n, $\hat{y}(n)$ is the forecasted flow at time n. Here, $y_{avg}(n)$ and $\hat{y}_{avg}(n)$ denote the average of observed and predicted flow, N represent the total number of observations, and N_p represents the number of peak flows greater than one-third of the mean peak flow observed.

6 Results and Discussions

The experiment was implemented using MATLAB version 8.1.0.604 (R2013a) for simultaneous flow forecasting for direct MIMO processes. Random initializations of weights were chosen from the normalized input data set for all networks. This was done to specify the appropriate range of the initial weights, while the final predictions were obtained by averaging the best three predictions of each model. The performances criteria described in Section 5 were used to test the prediction accuracy of the models when the test data were used. The performance statistics of the models for different time horizons are summarized in Table 2. The optimal network structure adopted for MIMO-RBFN was 6:12:6.

The RMSE and other performance criteria presented in Table 2 show an improvement of MIMO-NARX over MIMO-RBFNN for different lead times of the forecast. For instance, for one hour ahead forecasts, the RMSE value for the BG station was 16.802 Cumec in the case of MIMO-NARX, whereas MIMO-RBFNN had an RMSE value of 19.582 Cumec. Similarly, for three hours ahead, the RMSE values for the BG station were 36.063 Cumec and 47.841 Cumec, for six hours ahead they were 61.003 and 82.696 Cumec, and for twelve hours ahead they were 97.047 and 107.782 Cumec, respectively. Similarly, for the other forecasting stations, the RMSE values for MIMO-NARX were comparatively lower than those of MIMO-RBFNN.

The MAPE forecasting values presented in Table 2 show a similar improvement for MIMO-NARX over MIMO-RBFNN for all forecasting stations of the river system. Correlation coefficient r determines whether the forecasted values and actual values move in the same direction or not, and is confined to a range of $[-1\ 1]$. Values closer to 1 indicate more accuracy of the model in giving perfect movement of the two datasets and closer to -1 indicates imperfect fit. In the present case, the values of r for all the leading time horizons were > 0.90. The coefficient of efficiency (CE) expresses the model's efficiency and is confined to a range of $[-\infty\ 1]$. A CE value closer to 1 indicates perfect

suitability of the model. In the present study, from the CE values presented in Table 2, the model was within acceptable limits.

Table 2 Comparative performance index of MIMO-NARX and MIMO-RBFNN for different time horizons on tested data.

Forecasted Stations	Time Horizons	MIMO-NARX				MIMO-RBFNN			
		RMSE (Cumec)	*r*	CE	MAPE forecasting (%)	RMSE (Cumec)	*r*	CE	MAPE forecasting (%)
BG	1	16.802	0.99973	0.99946	0.12	19.582	0.99670	0.99316	1.25
	3	36.063	0.99878	0.99750	0.09	47.841	0.97979	0.94574	2.84
	6	61.003	0.99656	0.99285	0.02	82.696	0.98102	0.95509	1.54
	12	97.047	0.99147	0.98194	0.08	107.782	0.97687	0.94578	3.69
AG	1	14.152	0.999465	0.99892	0.13	18.550	0.99824	0.99530	1.38
	3	27.270	0.998051	0.99600	0.11	35.097	0.98096	0.96105	1.48
	6	45.366	0.994822	0.98895	0.12	53.772	0.98201	0.96226	1.88
	12	70.504	0.987737	0.97333	0.25	98.027	0.97572	0.94832	2.18
FT	1	25.992	0.999268	0.99853	0.16	35.584	0.99355	0.98408	2.61
	3	48.153	0.997527	0.99496	0.23	59.956	0.99004	0.97828	2.03
	6	80.768	0.993167	0.98583	0.29	98.202	0.98506	0.96427	2.19
	12	131.398	0.982481	0.96252	0.72	140.803	0.96134	0.91235	2.92
MJ	1	11.802	0.997903	0.99557	0.18	18.823	0.99121	0.98042	3.74
	3	22.090	0.992801	0.98450	0.13	27.822	0.98797	0.97540	3.69
	6	37.894	0.979787	0.95441	0.48	39.231	0.97658	0.95109	1.68
	12	59.722	0.949928	0.88688	1.81	64.957	0.94387	0.86592	3.78
DL	1	4.194	0.98918	0.97704	0.28	5.994	0.97918	0.95308	1.84
	3	5.822	0.97928	0.95577	0.44	6.234	0.97652	0.94925	2.74
	6	8.753	0.95340	0.90003	1.65	9.341	0.94808	0.88607	4.60
	12	13.449	0.89396	0.76411	3.56	13.711	0.88705	0.75449	4.11
MK	1	23.530	0.99946	0.99891	0.12	26.534	0.99449	0.98596	1.54
	3	44.714	0.99808	0.99607	0.12	62.102	0.98052	0.96032	1.44
	6	74.197	0.99491	0.98919	0.10	88.408	0.98202	0.96235	1.43
	12	114.982	0.98805	0.97408	0.26	123.569	0.97535	0.94742	1.88

Figure 5 shows a plot of model PFC for the test data. The PFC plot in Figure 4 for MIMO-NARX (indicated by the solid blue line) varies from 0.00 to 0.08 and that for MIMO-RBFN varies from 0 to 0.12 for different forecast lead times. The PFC value for both models increased with the increase in lead time but for all forecast lead times the PFC value for MIMO-NARX was smaller than that of MIMO-RBFN for any of the forecasting stations. However, the PFC value for both models indicates that both efficiently captured the peak flood values. The plots depicted in Figures 5 and 6 represent the response of MIMO-RBFNN and MIMO-NARX to the target data for multiple gauge stations. Figure 7 shows the plot for different lead times for the BG forecasting station. The plots for the other five stations remain undisclosed because a single station plot is sufficient to justify the model's accuracy. However, an overall analysis of the result based on statistical criteria for the other forecasting stations is shown in Table 2.

Thus, MIMO-NARX having the input delay lines and feedback delay lines of the network provides significant forecast accuracy even at peak flow. MIMO-NARX takes more advantage of the input data to model at different lead horizons over MIMO-RBFNN and can be considered a suitable model for simultaneous forecasting of multiple flows of a river system.

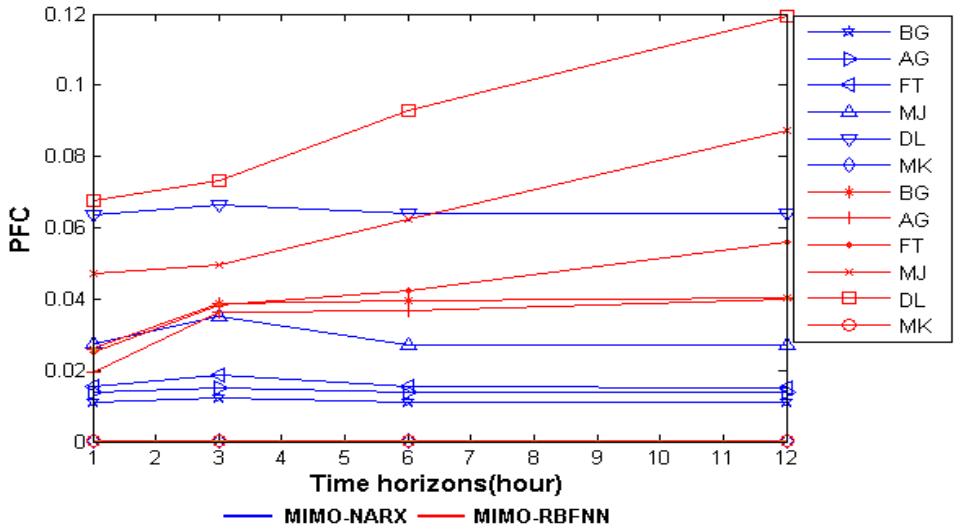

Figure 4 Comparison of PFC using MIMO-NARX and MIMO-RBFNN.

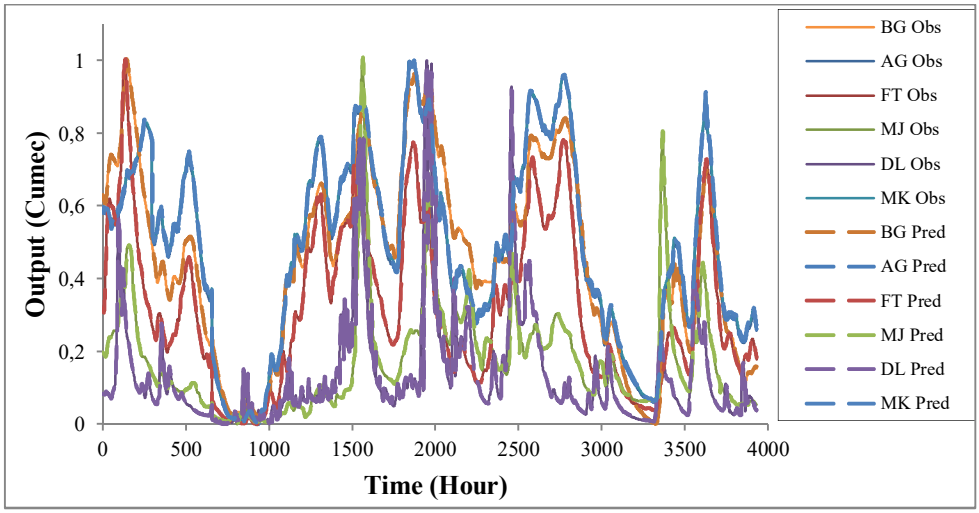

Figure 5 Response of MIMO-NARX vs. observed data for multiple gauge stations.

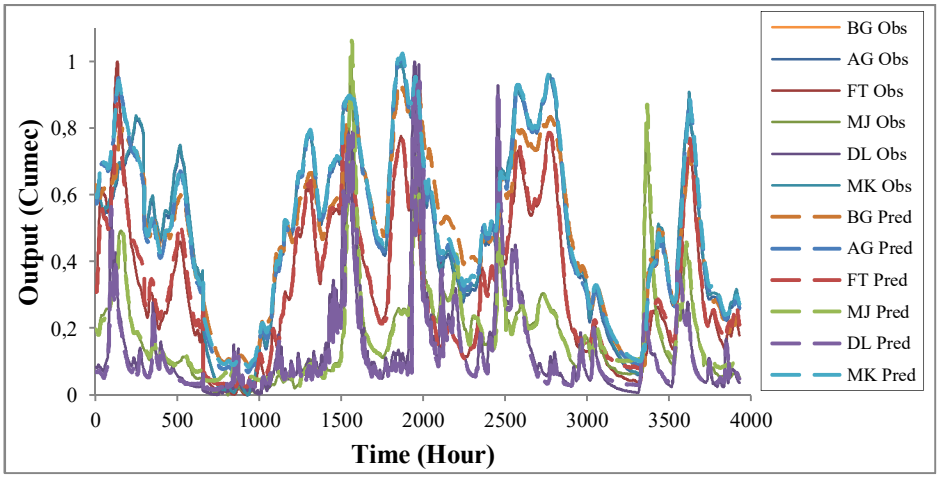

Figure 6 Response of MIMO-RBFN vs. observed data for multiple gauge stations.

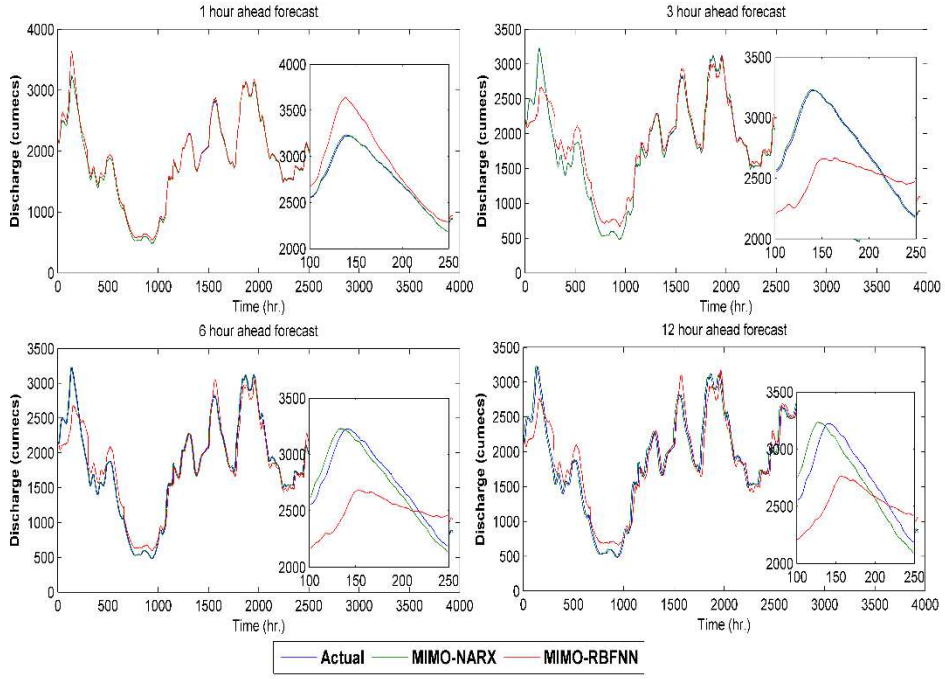

Figure 7 Forecasted time series for MIMO-NARX and MIMO-RBFNN vs. observed data for BG gauge station at different time horizons.

7 Conclusions

Application of RBFNN for simultaneous flow forecasting on a river system was adopted through direct MIMO processes. The result of MIMO-RBFNN was compared with that obtained by the application of MIMO-NARX. Both the MIMO-RBFNN and MIMO-NARX models were applied based on Eq. 3, which shows that the predictions are made directly for multiple time steps ahead, i.e. not iteratively. It was implemented through MATLAB version 8.1.0.604 (R2013a). The models were applied to the Barak River system, Assam (India) based on real-time hourly flow data pertaining to the monsoon seasons of a six-year period (2000-2005). The key conclusions that can be drawn from this study are:

1. MIMO-RBFNN provided equally satisfactory results as MIMO-NARX, which indicates the efficacy of MIMO-RBFNN for simultaneous flow forecasting on river systems.
2. It was observed that with large numbers of input sequences, computational time was longer for MIMO-NARX compared to MIMO-RBFNN.
3. Comparatively, MIMO-NARX proved to perform better than MIMO-RBFNN on all counts, even at higher forecast lead times. However, the results obtained using MIMO-RBFN are also acceptable.

The applicability of MIMO-RBFN can also be investigated for other nonlinear time series problems, particularly in the area of hydrology. It can also be tested for long-range operational time horizons using multivariate data.

References

[1] Billings, S.A. & Voon, W.S.F., *A Prediction-error and Stepwise-Regression Estimation Algorithm for Non-linear Systems*, International Journal of Control, **44**(3), pp. 803-822, Oct.1986.

[2] Korenberg, M., Billings, S.A., Liu, Y.P. & McIlroy, P.J., *Orthogonal Parameter Estimation Algorithm for Non-linear Stochastic Systems*, International Journal of Control, **48**(1), pp. 193-210, Oct. 2007.

[3] Leontaritis, I.J. & Billings, S.A., *Model Selection and Validation Methods for Non-linear Systems*, International Journal of Control, **45**(1), pp. 311-341, Mar. 1987.

[4] Akhtar, M.K., Corzo, G.A., Andel, S.J.V. & Jonoski, A., *River Flow Forecasting with Artificial Neural Networks Using Satellite Observed Precipitation Pre-processed with Flow Length and Travel Time Information: Case Study of the Ganges River Basin*, Hydrol. Earth Syst. Sci., **13**, pp. 1607-1618, Sept. 2009.

[5] Chen, S.M, Wang, Y.M. & Tsou, I., *Using Artificial Neural Network Approach for Modeling Rainfall–runoff due to Typhoon*, J. Earth Syst. Sci. **122**(2), pp. 399-405, April 2013.

[6] Firat, M., *Articial Intelligence Techniques for River Flow Forecasting in the Seyhan River Catchment*, Turkey, Hydrology and Earth System Sciences Discussions, European Geosciences Union, **4**(3), pp. 1369-1406, June 2007.

[7] Lekkas, D.F., Onof, C., Lee, M.J. & Baltas, E.A., *Application of Artificial Neural Networks for Flood Forecasting*, Global Nest: the Int. J., **6**(3), pp. 205-211, Jan. 2004.

[8] Partal, T., *River Flow Forecasting using Different Artificial Neural Network Algorithms and Wavelet Transform*, Can. J. Civ. Eng., **36**(1), pp. 26-39, Dec. 2009.

[9] Rezaei, M., Motlaq, A.A.A., Mahmouei, A.R. & Mousavi, S.H., *River Flow Forecasting using Artificial Neural Network (Shoor Ghaen)*, Ciência e Natura, Santa Maria, **37**(1), pp. 207−215, 2015.

[10] Shamseldin, A.Y., *Artificial Neural Network Model for River Flow Forecasting in a Developing Country*, Journal of Hydroinformatics, **12**(1), pp. 22-34, Jan. 2010.

[11] Bontempi, G. & Ben Taieb, S., *Conditionally Dependent Strategies for Multiple-step-ahead Prediction in Local Learning*, International Journal of Forecasting, **27**(3), pp. 689-699, Sept. 2011.

[12] Claveria, O., Monte, E. & Torra, S., *Multiple-input Multiple-output vs. Single-input Single-output Neural Network Forecasting*, Research Institute of Applied Economics, 2015.

[13] Maciel, L., Gomide, F. & Ballini, R., *MIMO Evolving Functional Fuzzy Models for Interest Rate Forecasting*, Computational Intelligence for Financial Engineering & Economics, IEEE Conference, ISBN: 978-1-4673-1802-0, 2012.

[14] Tripura, J. & Roy, P., *Flow Forecasting in Multiple Sections of a River System*, KSCE Journal of Civil Engineering, **21**(2), pp. 512-522, Feb. 2017.

[15] Choudhury, P. & Roy, P., *Forecasting Concurrent Flows in a River System Using ANNs*, J. Hydrol. Eng., **20**(8), March 2015.

[16] Taieb, S.B., Bontempi, G., Sorjamaa, A. & Lendasse, A., *Long-term Prediction of Time Series by Combining Direct and MIMO Strategies*, International Joint Conference on Neural Networks, pp. 3054-3061, 2009.

[17] Powell, M., *Radial Basis Functions for Multivariable Interpolation: a Review*, in J.C. Mason, M. G. Cox (Eds.), Algorithms for Approximation, Clarendon Press, New York, NY, USA, pp. 143-167, 1987.

[18] Broomhead, D.S. & Lowe, D., *Multi-variable Functional Interpolation and Adaptive Networks*, Complex Syst. **2**(3), pp. 321-355, 1998.

[19] Park, J. & Sandberg, I.W., *Universal Approximation using Radial-Basis-Function Networks*, Neural Computation, **3**(2), pp. 246-257, June 1991.

[20] Awad, M., Pomares, H., Ruiz, I.R., Salameh, O.I. & Hamdon, M., *Prediction of Time Series Using RBF Neural Networks: A New Approach of Clustering*, International Arab Journal of Information Technology, **6**(2), pp. 138-143, April 2009.

[21] Rout, M., Majhi, B., Mohapatra U.M. and Mahapatra R., Stock Indices *Prediction Using Radial Basis Function Neural Network*, in: B.K Panigrahi et al. (Eds.): SEMCCO, LNCS 7677, pp. 285-293, 2012.

[22] Leontaritis, I.J. & Billings, S.A*., Input-output Parametric Models for Non-linear Systems Part I: Deterministic Non-linear Systems*, International Journal of Control, **41**(2), pp. 303-328), May 1985.

[23] Chen, S., Billings, S.A. & Grant, P.M., *Non-linear System Identification Using Neural Networks*, Int. J. Control, **51**(6), pp. 1191-1214, Jan. 1990.

[24] Diaconescu, E., *The Use of NARX Neural Networks to predict Chaotic Time Series*, WSEAS Transactions on Computer Research, **3**(3), pp. 182-191, March 2008.

[25] Lin, T., Horne, B.G., Tino, P. & Giles, C.L., *A Delay Damage Model Selection Algorithm for NARX Neural Networks*, IEEE Transactions on Signal Processing, **45**(11), pp. 2719-2730, Nov. 1997.

[26] Siegelmann, H.T., Horne, B.G. & Giles, C.L., *Computational Capabilities of Recurrent NARX Neural Networks*, IEEE Trans. Syst., **27**(2), pp. 208-215, April 1997.

[27] Horne, B.G. & Giles, C.L., *An Experimental Comparison of Recurrent Neural Networks*, Advances in Neural Information Processing Systems 7, pp. 697-704, 1995.

[28] Jiang, C. & Song, F., *Sunspot Forecasting by Using Chaotic Time series Analysis and NARX Network*, Journal of Computers, **6**(7), pp. 1424-1429, July 2011.

[29] Aguilar-Lobo, L.M., Loo-Yau, J.R., Rayas-Sanchez, J.E., Ortega-Cisneros, S., Moreno, P. & Reynoso-Hernandez, J.A., *Application of The Narx Neural Network as a Digital Predistortion Technique for Linearizing Microwave Power Amplifiers*, Microwave and Optical Technology Letters, **57**(9), pp. 2137-2142, Sept. 2015.

[30] Yiu, J.C. & Wang S., *Multiple ARMAX Modeling Scheme for Forecasting Air Conditioning System Performance*, Energy Conversion and Management, **48**(2007), pp. 2276-2285, June 2007.

[31] Zhang, X., *Time Series Analysis and Prediction by Neural Networks*, Optimization Methods and Software **4**(2), pp. 151-170, Feb.1994.