International Journal of Water Resources and Environmental Engineering Vol. 3(1), pp. 30-40, January 2011
Available online at http://www.academicjournals.org/ijwree
ISSN-2141-6613 ©2011 Academic Journals

Full Length Research Paper

Stream flow forecasting using Levenberg-Marquardt algorithm approach

Deepika Yadav, R. Naresh and V. Sharma*

Electrical Engineering Department, National Institute of Technology Hamirpur, HP, India-177005.

Accepted 3 January, 2010

Many of the activities associated with the planning and operation of water resource systems require forecasts of future events. For the hydrologic component that forms the input for water resource systems, there is a need for both short and long term forecasts of stream flow events, in order to optimize the real-time operation of the system or to plan for future expansion. For this historical inflow series from Sewa hydroelectric Project Stage-II which is a run-of-the river project has been used. For model development, 16 years historical inflows data of the river out of available 18 years inflow data has been used and the Artificial Neural Network Model has been trained to predict 2 years inflows. In order to accomplish this task, historical inflow series is employed for training, validating and testing with three different proportions of ratio 60:20:20, 80:10:10 and 90:05:05 were analyzed. The analysis of this study demonstrates the ability of neural network prediction model, to forecast quite accurately ten days inflows of two years ahead and generate synthetic series of ten days inflows that preserve the key statistics of the historical ten days inflows which in a way helps in effective utilization of available water, especially in a multipurpose context.

Key words: Artificial neural network, Levenberg-Marquardt algorithm, stream flow forecasting.

INTRODUCTION

To begin with, as water inflow is the fuel for hydro power production, the main challenge for a hydro producer with reservoir capacity is deciding on how much electricity to produce today, versus future periods become essential as per ABT norms (Bhushan, 2005; Christensen and Soliman, 1988; Deshmukh et al., 2008) for energy system planning. To fulfill that criteria, proper planning of hydro power through short and long-term forecasts of stream flow are carried out for knowing the hydrological behaviour of a water structure. Short term forecasts are applicable for real time operation of water management system and for flood warning. Long term forecasts are applicable to operation and management of water supply systems. Moreover, stream flow data are very important for many areas of water engineering such as dam planning, flood mitigation, operation of water reservoirs, distribution of drinking water and drainage water, hydropower generation in dry periods, planning of river transport and for many other purposes as reported by Salas et al. (1985) and Salas (1993).

No two hydroelectric systems in the world are alike but they are all different. The reasons for the differences are the natural differences in the watersheds, the differences in the manmade storage and release elements used to control the water flows, and the very many different types of natural and manmade constraints imposed on the operation of hydroelectric systems (Wood and Wollenberg, 1984). The stream flow data for River Sewa has been collected through a very sparse and distinct data acquisition networks. Also, lot of uncertainty is involved in stream flow data because of its non-stationary nature due to wet and dry periods over the year (Maier and Dandy, 2000).

Furthermore, the inherently non-linear relationship between input and output flow challenges stream flow forecasting processes. Several researchers have suggested different model-driven and data-driven approaches to predict stream flows. In traditional model-driven approaches, such as ARMA-type models, for fitting an ARMA type model to a historical time series the data need to be stationary (Hipel, 1985) and should follow normal distribution (Bender and Simonovic, 1994)

*Corresponding author. E-mail: veena@nitham.ac.in.

pattern. Otherwise, a technique such as differencing (Box and Pierce, 1970) is applied to induce stationarity and then through Box-Cox transformation, normally distributed patterns are achieved. Whereas, while developing artificial neural network models, the statistical distribution of the data does not have to be known (Box and Pierce, 1970) and non-stationarities in the data, such as trends and seasonal variation, are accounted for by the internal structure of the ANNs (Dandy and Maier, 1996). ANNs are suited to complex problems, where the relationships between the variables being modelled are not well understood. ANNs differ from the traditional approaches in the sense that, they belong to a class of data-driven approaches, as opposed to traditional model-driven approaches such as ARMA-type models.

Data-driven approaches have the ability to determine which model inputs are critical, so that there is no need for a prior knowledge about relationships between variables. They are relatively insensitive to noisy data, unlike ARMA-type models, as they have the ability to determine the underlying relationship between model inputs and outputs, resulting in good generalization ability. Model-driven approaches, on the other hand, require some understanding of the problem, as the model order has to be determined before the unknown model parameters can be estimated (Dandy et al., 1996). In addition, an ANN is a nonlinear mathematical structure capable of representing arbitrarily complex nonlinear processes, that relate the inputs and outputs data (Hsu et al., 1995) sets of any system.

The success with which ANN have been used to model dynamic systems in areas of science and engineering suggests that the ANN approach have become one of the commonly used and powerful alternative technique to deal with time series prediction in situation where explicit knowledge of the internal hydrologic sub-processes of the watershed is not required. Application of ANN to problems involving rainfall-runoff modeling (Hsu et al., 1995; Lorrai and Lorrai, 1995; Cheng and Noguchi, 1996; Smith and Eli, 1995) and weather (French et al., 1992; Jayawardena and Fernando 1996) and river flow prediction (Karunanithi et al., 1994; Raman et al., 1995; Raman and Chandramouli 1996) have been reported in the literature.

In an effort to address the difficulty and inherent uncertainty of forecasting for long term planning horizon, the ANN based approach to stream flow forecasting has been investigated, which uses a black-box approach, with little rationalization about possible interactions between input and output data sets (that is historical inflow series) taken from Sewa hydro electric project stage-II. In order to check the sensitivity of the data training, validation and testing data sets with three different proportions of ratio 60:20:20, 80:10:10 and 90:05:05 were analyzed. Then ANN generated results were evaluated using Mean Square Error and Regression R value in neural network fitting tool box in MATLAB 7.8.

Study area and data set

The present study focuses on stream flow forecasting in long-term reservoir operation scheduling for Sewa Hydroelectric Project Stage-II, a run-of-the river project, which fulfills the partial requirements of the irrigation in state of Jammu and Kashmir. The power house is located in a village called Mashka near the junction of Sewa and Ravi, The project will generate 533.52 million units in a 90% dependable year and also provide 120 MW peaking capacity in the power system of northern region. This plant has a small reservoir with maximum storage capacity of 0.9174 million cubic meters (MCM) and average storage capacity of 0.2234 MCM. Moreover, elevation is taken from above mean sea level through some level sensors, maximum reservoir level of this plant is 1200 m and average reservoir level measured is 1184 m. The project envisages 53 m high concrete gravity dam, a 10,020 m long head race tunnel and its power house will be equipped with 3*40 MW vertical Pelton turbine units with rated net head of 560 m. Its geographical coordinates having latitude 32° 36’ 38” N to 32° 41’ 00” N and longitude 75° 48’ 46” E to 75° 55’ 38” E is shown in Figure 1, referred from site (www.nhpcindia.com).

In this paper for model development, 16 years historical inflow data of the river out of available 18 years inflow data has been used. Two years inflow of Sewa River has been forecasted and validated by the remaining two years data. The historical time series data for stream flow forecasting is taken on duration of 10 days interval. There are 36 data in a given year. The inflows are as shown in Figure 2.

Neural network overview

Neural networks are inspired by nervous systems found in biological organisms. It is comprised of data processing units (neurons) connected via adjustable connection weights. Neurons are arranged in layers, an input layer, hidden layer(s), and an output layer. There is no specific rule that dictates the number of hidden layers. The function is largely established based on the connections between elements of the network. In the input layer, each neuron is designated to one of the input parameters. The network learns by applying a back-propagation algorithm which compares the neural network simulated output values to the actual values and calculates a prediction error. The error is then back propagated through the network and weights are adjusted as the network attempts to decrease the prediction error by optimizing the weights that contribute most to the error. The training or learning of the network occurs through the introduction of cycles of data patterns (epochs or iterations) to the network. One problem with neural network training is the tendency for the network to memorize the training data after an extended learning phase. If the network over learns the training data, it is more difficult for the network to generalize to a data set that was not seen by the network during training. Therefore, it is common practice to divide the data set into a learning data set, that is used to train the network and a validation data set that is used to test network performance. Figure 3 shows the representation of neural Network diagram with inputs ai, weights wi, hidden layer,

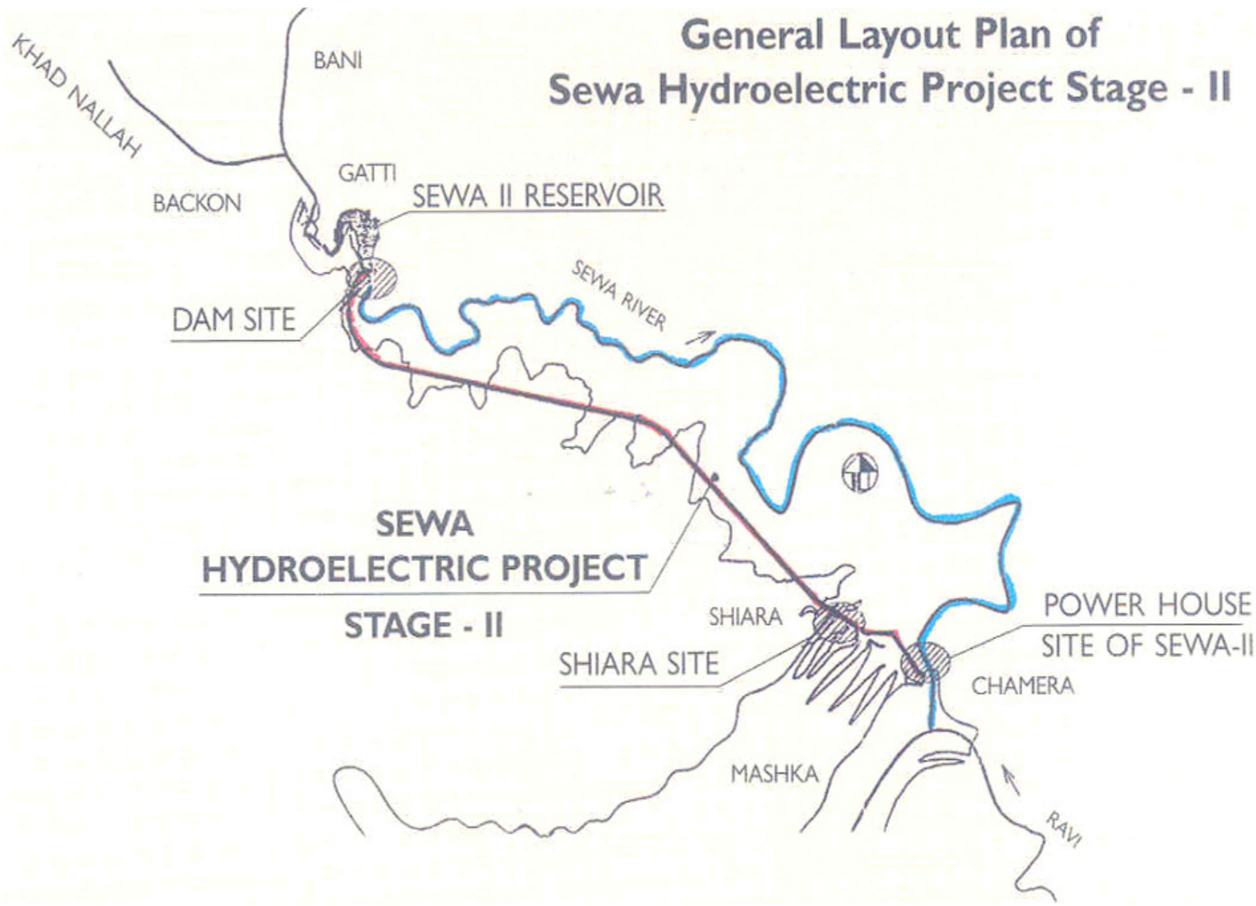

Figure 1. Location map.

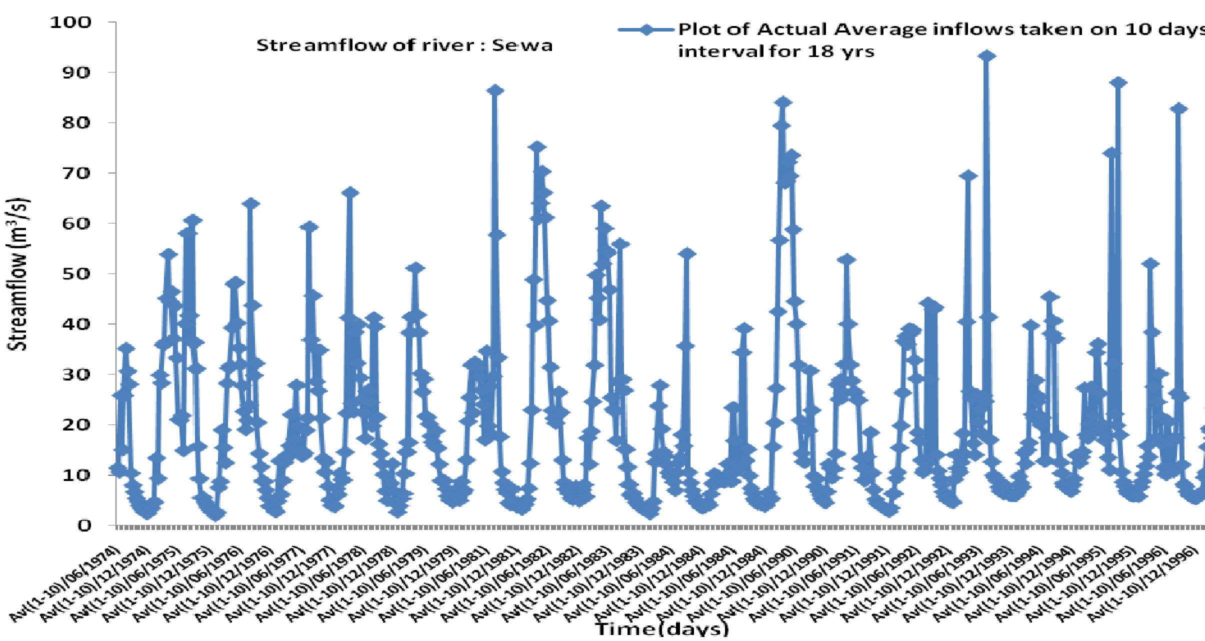

Figure 2. Plot of the river stream flow data originating from Himalayan region.

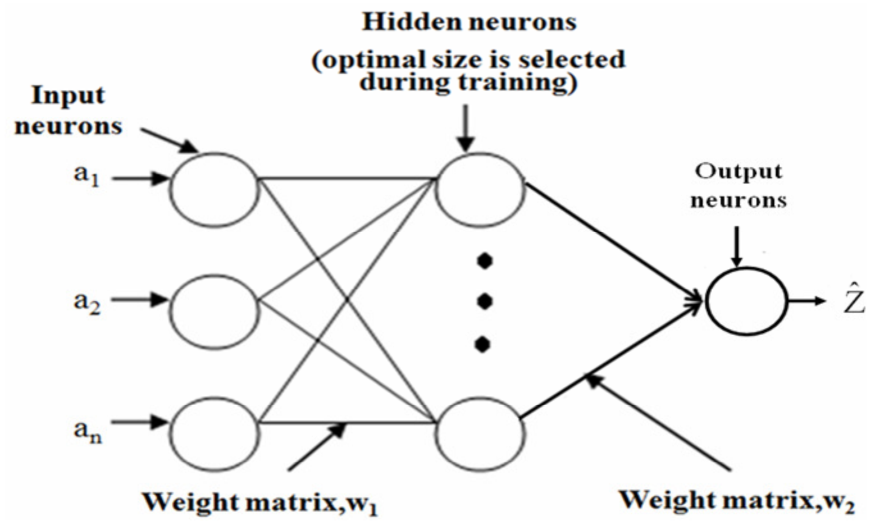

Figure 3. Feed-forward neural network model.

and an output $\hat{z}$ (Beale et al., 1996). In the present study, neural network fitting tool (nftool) of MATLAB 7.8 (Beale et al., 1996; Sivanandam et al., 2006) has been used. This research employed supervised learning where the target values for the output are presented to the network, in order for the network to update its weights. Supervised learning attempts to match the output of the network to values that have already been defined. After training network verification is applied in which only the input values are presented to the network so that the success of the training can be established, an algorithm that trains ANN 10 to 100 times faster than the usual back propa-gation algorithm is the Levenberg-Marquardt algorithm. While back propagation is a steepest descent algorithm, the Levenberg-Marquardt algorithm is a varia-tion of Newton's method (Hagan and Menhaj, 1994). In this paper, the Levenberg-Marquardt algorithm has been employed which is an approximation to Newton's method.

Prediction of stream flow data using ann

To forecast stream flow data for Sewa hydro electric project stage-II with neural network using Levenberg-Marquardt algorithm has been investigated in this paper. The three layers network structure is shown in Figure 3. To solve this problem, the network was trained by using Matlab neural networks module (nftool). For investigating the suitability of ANN, three ratios between training, validation and testing sets were considered that is, 60:20:20, 80:10:10 and 90:05:05. In order to check the sensitivity of neural network, initialization of connection weights, training, validation and testing operations have been performed with 5 independent random trials. From the simulation study which was carried out on three different ratios, it was found that lower proportions of ratio produced more accurate predictions. Although it requires many runs to converge, to get expected training, yet once the system was trained then it tests the remaining samples of data.

Figure 4 shows the performance accuracy employing different numbers of hidden nodes for 90:05:05 ratio set. From the graph, it can be realized that good prediction accuracy is achieved with 5 numbers of neurons in the hidden layer on fifth trial. It can be noted from the graph that with increase in number of neurons in the hidden layer, prediction accuracy also increases to some extent. In this study, number of neurons in the hidden layer was obtained via trial an error method.

Here, hyperbolic tansig function f(x) = 1/ (1 + exp(x)) is applied for the hidden layer, and the linear transfer function (purelin) is used in the output layer. Input data is applied after normalization process between -1 and +1. To evaluate neural network performance, initialization of connection weights, training, validation and testing has been performed with five independent random trials for weight initialization as listed in Table 1. The comparison of the mean squared error (MSE) values, indicates the average squared difference between outputs and targets, which is used to assess the network performance and is given as:

$$MSE = \frac{\sum_{m=1}^{M} (y_m - d_m)^2}{M} \qquad (1)$$

where, y_m and d_m are the network output, and the desired output at any sample 'm' respectively; and M is the length of the investigated data sets.

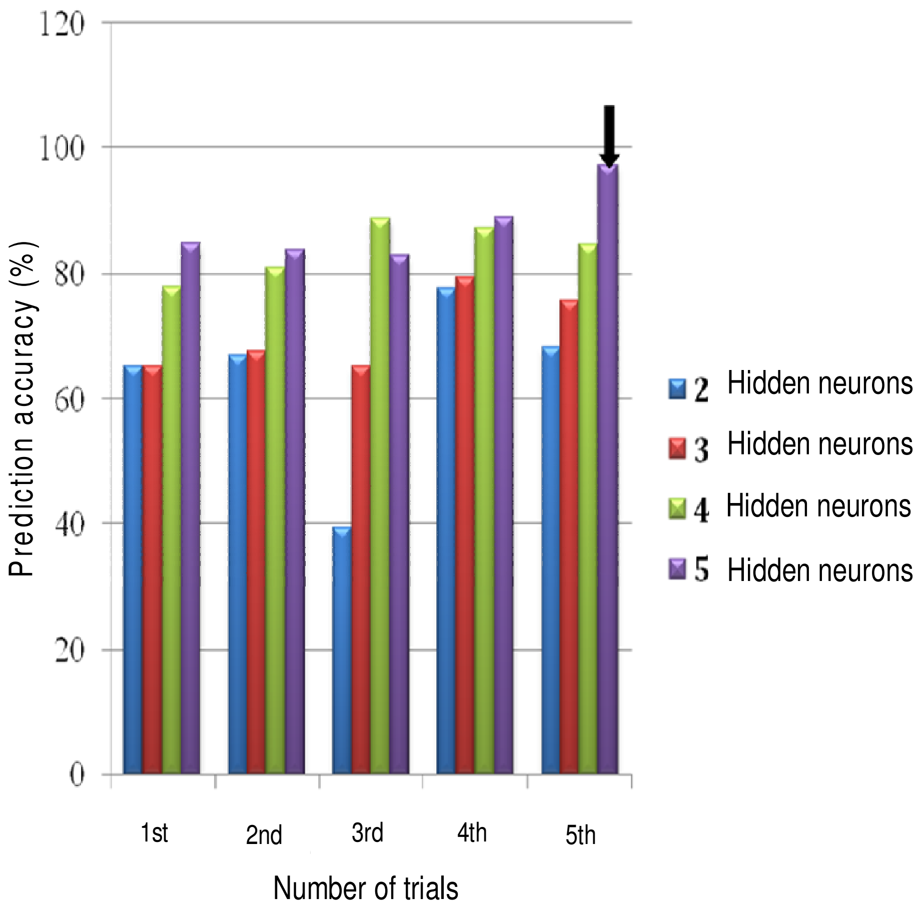

Figure 4. Prediction Accuracy versus number of trials with different hidden neurons for 90:05:05 ratio set

Table 1. Performance evaluation of training, validation and testing for 90:05:05 ratio set.

Number of hidden layer neurons	Operation	MSE Best	MSE Worst	MSE Average	R Best	R Worst	R Average	Standard deviation MSE	Standard deviation R
2	Training	9.67E+01	1.49E+02	1.24E+02	8.68E-01	6.46E-01	7.40E-01	2.41E+01	9.09E-02
2	Validation	2.34E+00	4.07E+02	1.06E+02	1.00E+00	1.00E+00	1.00E+00	1.71E+02	5.48E-07
2	Testing	1.14E+00	1.76E+02	5.14E+01	1.00E+00	1.00E+00	1.00E+00	7.28E+01	4.47E-07
3	Training	4.59E+01	1.53E+02	8.13E+01	9.87E-01	7.20E-01	8.84E-01	4.44E+01	1.02E-01
3	Validation	5.01E+00	3.80E+01	2.00E+01	1.00E+00	1.00E+00	1.00E+00	1.26E+01	4.47E-07
3	Testing	1.06E+00	3.44E+01	1.74E+01	1.00E+00	1.00E+00	1.00E+00	1.29E+01	5.48E-07
4	Training	6.04E+00	6.75E+01	2.76E+01	9.94E-01	9.26E-01	9.75E-01	2.63E+01	2.80E-02
4	Validation	1.81E+00	3.06E+01	1.44E+01	1.00E+00	1.00E+00	1.00E+00	1.13E+01	0.00E+00
4	Testing	4.46E-01	5.96E+01	1.98E+01	1.00E+00	1.00E+00	1.00E+00	2.64E+01	4.47E-07
5	Training	4.27E-01	8.51E+01	2.07E+01	1.00E+00	8.44E-01	9.65E-01	3.61E+01	6.78E-02
5	Validation	1.63E-01	6.06E+01	1.86E+01	1.00E+00	1.00E+00	1.00E+00	2.41E+01	5.48E-07
5	Testing	4.24E-01	8.87E+01	2.98E+01	1.00E+00	-1.00E+00	6.00E-01	3.55E+01	8.94E-01

The standard deviation of a sample of observations is the square root of the average of the squared deviations about their mean (James et al., 1994)

$$\text{Standard deviation} = \sqrt{\frac{\sum_{m=1}^{M}(y_m - \bar{y})^2}{M}} \qquad (2)$$

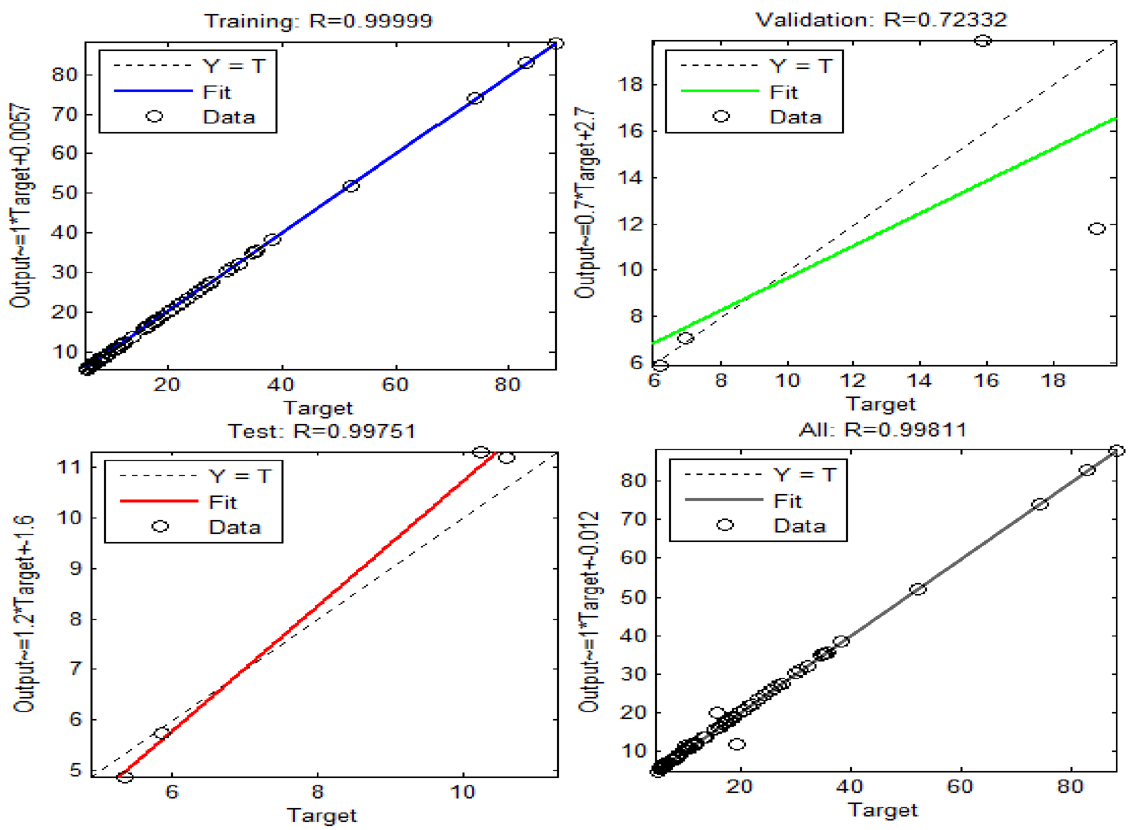

Figure 5. Regression plots for actual and forecasted results by feed-forward neural network model for training, validation, testing samples and all data set for 90:05:05 ratio set.

The standard deviation is a measure that summarizes the amount by which every value within a dataset varies from the mean. Effectively, it indicates how tightly the values in the dataset are bunched around the mean value. When the values in a dataset are pretty tightly bunched together, the standard deviation is small. When the values are spread apart, the standard deviation will be relatively large. The standard deviation is usually presented in conjunction with the mean and is measured in the same units. In Table 1, standard deviation and correlation coefficient R values provide how well model is close to actual values. In other words, it provides a measure of how well future outcomes are likely to be predicted by the model. Hence, it is desired that correlation coefficient R, values to be very high that is, close to 1. The performance was evaluated in terms of the correlation coefficient R, computed as

$$R = \sqrt{\frac{\sum_{m=1}^{M}(y_m - \bar{y})^2 - \sum_{m=1}^{M}(y_m - \hat{y}_m)^2}{\sum_{m=1}^{M}(y_i - \bar{y})^2}} \tag{3}$$

where, y_m = the observed dependent variable, $\hat{y}_m$ = the fitted dependent variable for the independent variable x_m $\bar{y}$ = mean, x_m = the independent variable in the m [th] trial, $\sum_{m=1}^{M}(y_m - \bar{y})^2$ represents total sum of squares, while, $\sum_{m=1}^{M}(y_m - \hat{y}_m)^2$ represents residual sum of squares

Correlation coefficient R, is a measure of the explanatory power of the model. Here for best model chosen values of R is 0.9999, 0.72332 and 0.99751 for training, validation and testing respectively as shown in Figure 5. As per the model, 99% of variation in dependent variable has been explained by independent variable. In Figure 5, the dashed line is the perfect fit line where outputs and targets are equal to each other. The circles are the data points and the coloured line represents the best fit between outputs and targets. Here, it is important to note that circles gather across the dashed line therefore, the outputs are not far from targets. Figure 6 depicts the training, validation and testing mean square error values for Levenberg-Marquardt algorithm with 5 number of neurons in the hidden layer. In this network, minimum MSE for best model in case of training is 0.427453, validation is 14.6178 and for testing is 0.423842. Here,

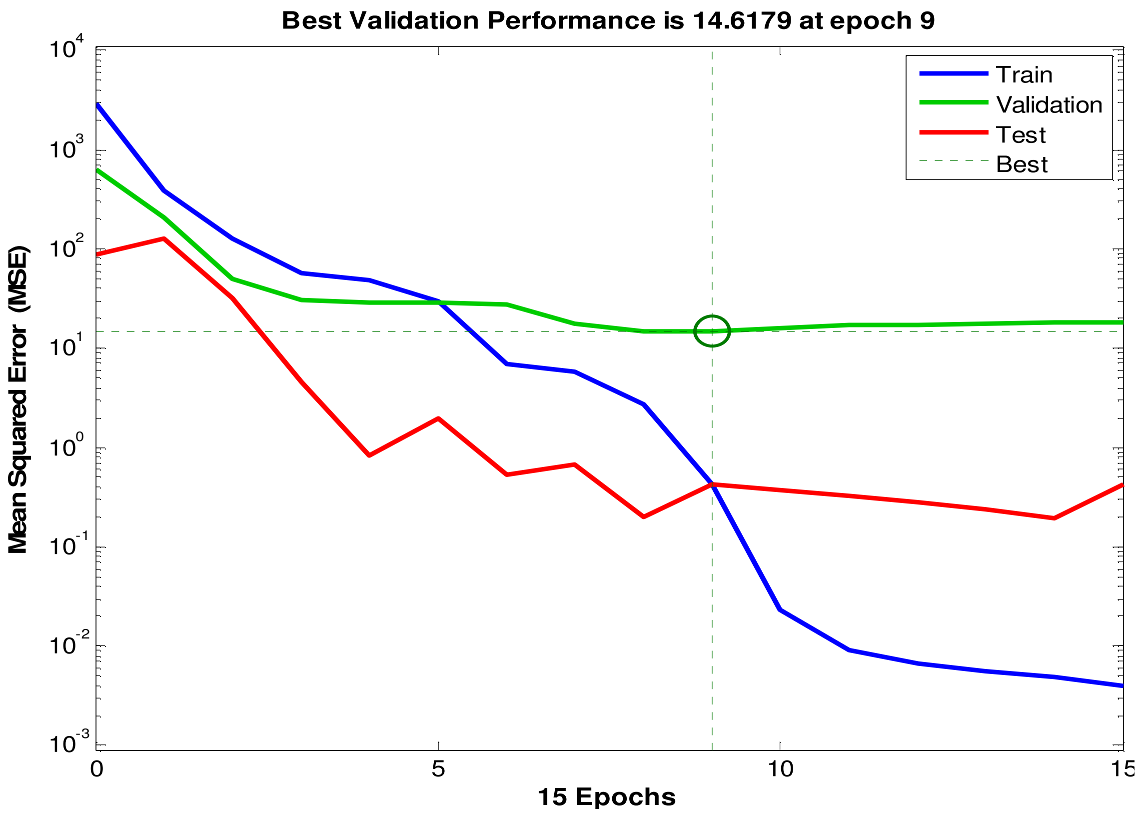

Figure 6. Training, validation and testing mean square errors for Levenberg-Marquardt algorithm with 5 neurons for 90:05:05 ratio set

the training set is used for computing the gradient and the neural network weights. The errors obtained from the validation set are monitored during the training. In this work, the number of input-output data pairs in the validation set is chosen to be 5% of the full training set (Jason et al., 2009; Erik et al., 2010). When the network is starting to over fit the data from the training set, the errors obtained from the validation set usually start to increase. When the validation error has increased for a specified number of iteration, the training stops and the weights and biases at the minimum of the validation error is returned, whereas testing provides an independent measure of network performance during and after training. For updating Levenberg-Marquardt algorithm, the scalar quantity µ called adaptive parameter has been used. Also, performance function which is the sum of squared errors between the target outputs and the network's simulated outputs (as is typical in training feed forward networks) is proportional to µ (Hagan and Menhaj, 1994).

Consequently, µ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function is reduced at every iteration of the algorithm. This algorithm appears to be the fastest method for training moderate-sized feed forward neural networks.

Variation of the gradient error, value of µ and validation error are shown in Figure 7. Moreover, stopping of training process is shown here at epoch 15, along with gradient as it reaches to minimum. Error plot between actual and predicted results by neural network model has been shown in Figure 8. It is seen that 96% of the errors accumulate between -2 and +2, and most of them are smaller than 2%. This indicates that, the network is performing very well. In general, the location of the error is dependent on the data used to train the network and on the initial conditions of the output layer (Haykin, 1994).

Comparison of forecast made by ANN prediction models with three different proportions of data ratio

In order to choose the best prediction model for forecasting, three different proportions of data ratio 60:20:20, 80:10:10 and 90:05:05 were considered. Five different random trials with two, three, four and five number of neurons in the hidden layer have been conducted by using Levenberg-Marquardt algorithm in nftool box in MATLAB 7.8. Throughout the analysis, where training, validation and testing was performed several times to choose the best model with best fit and minimum MSE. Each of the training session was carried

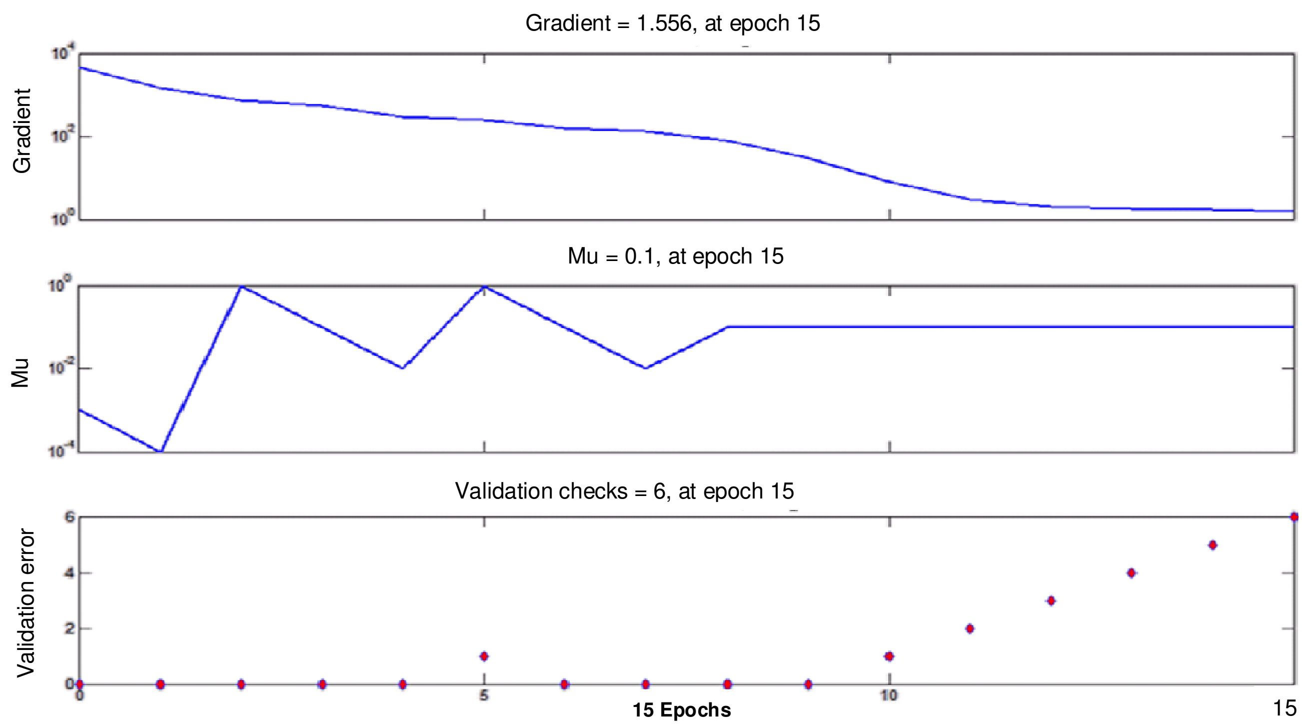

Figure 7. Variation of the gradient error, μ and validation error for 90:05:05 ratio set.

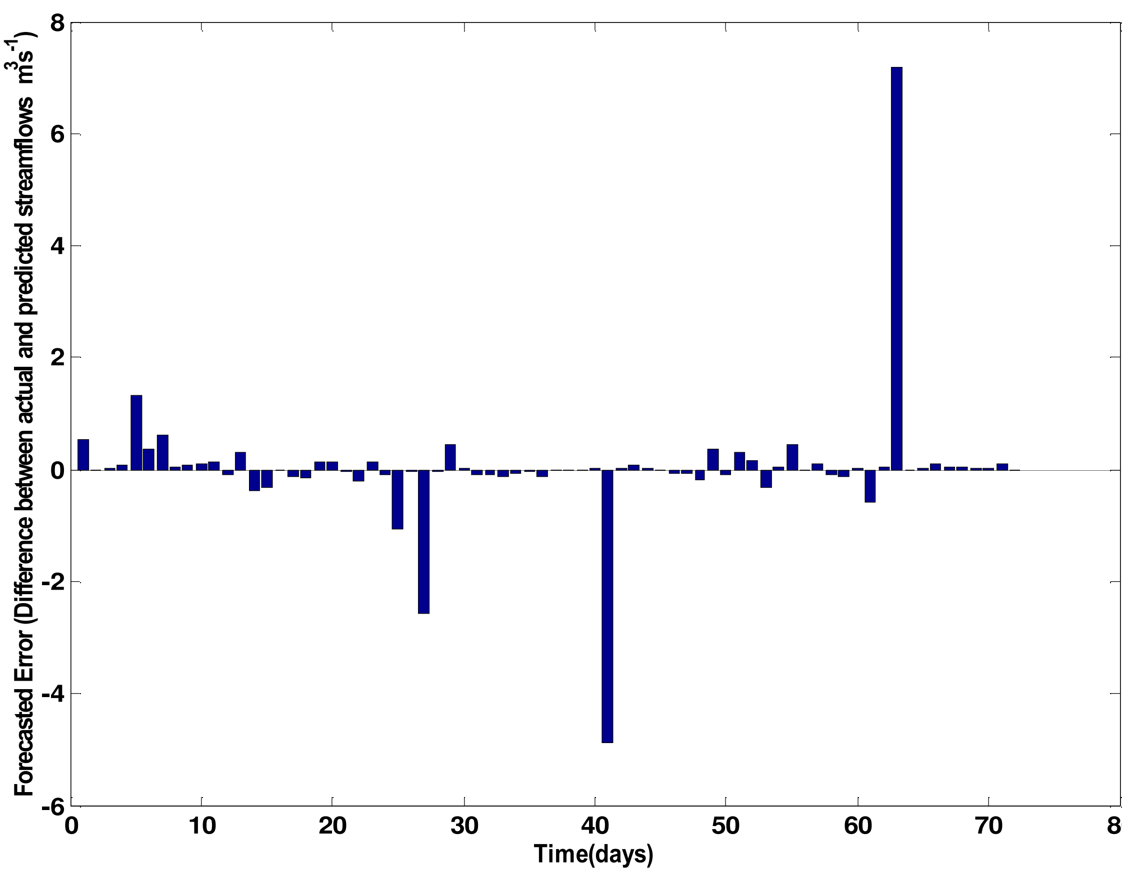

Figure 8. Error plot between actual and predicted flow results by neural network model for 90:05:05 ratio set

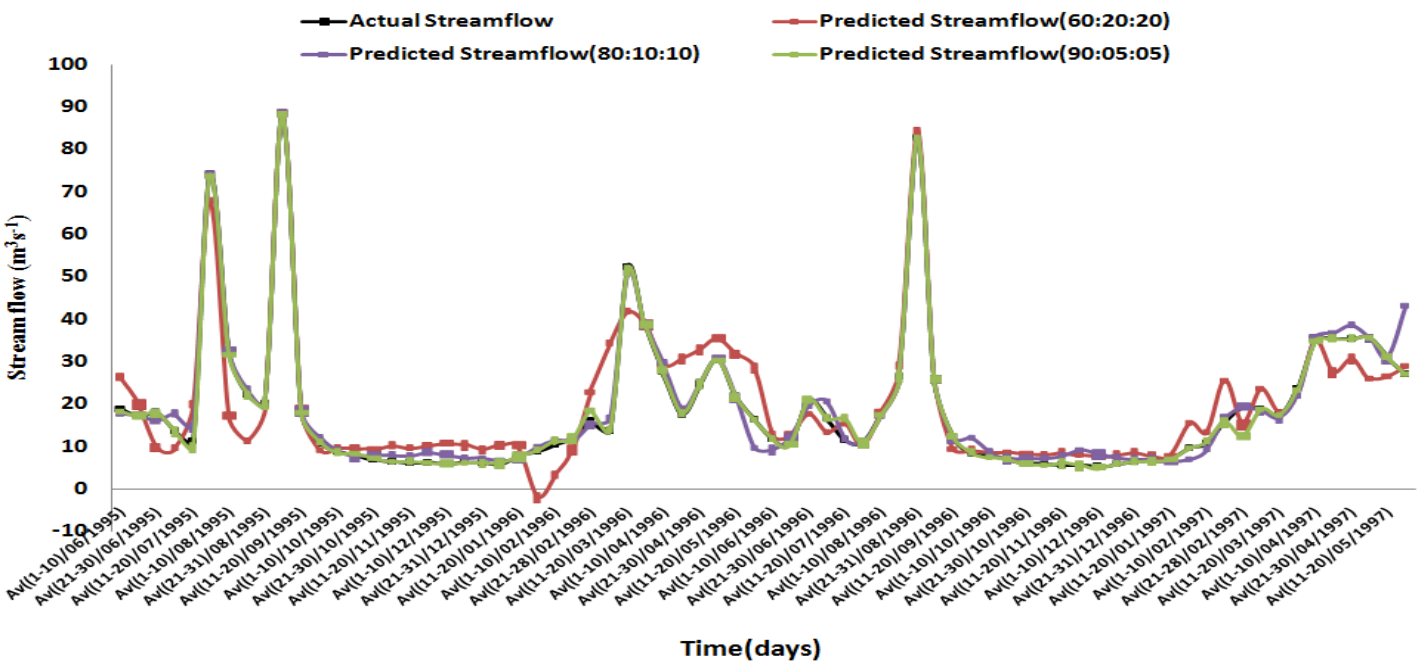

Figure 9. Comparison of best forecast results given by three different proportions of ratio with five neurons in the hidden layer.

Table 2. Various percentage of forecast errors in ANN models with three different proportions of ratio.

Ratio	PME (%)	PMAE (%)	PMSE (%)	PRMSE (%)	Accuracy (%)
60:20:20	-0.07548477	5.62249E-05	0.004103	0.064051	67.3111
80:10:10	-0.05354715	3.98847E-05	0.002064	0.045436	87.3334
90:05:05	0.001448051	0.025580012	1.51E-06	0.001229	97.1651

out with different initial weights and in each of the cases best prediction was obtained with 5 neurons in the hidden layer. Stream flow forecast results for 2 years with three different proportions of data ratio, that is 60:20:20, 80:10:10 and 90:05:05 have been shown in Figure 9. Table 2 presents various percentage forecast errors in ANN models with three different proportions of ratio as illustrated subsequently. If $\hat{z}_t$ is forecast data and z_i is actual data, then forecast error is $e_i = z_i - \hat{z}_t$. Here n is the length of investigated data for 2 years. Different types of forecast error:

1. Percentage mean error $PME = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{e_i}{z_i} \right) \times 100$

2. Percentage mean absolute error $PMAE = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{|e_i|}{z_i} \right) \times 100$

3. Percentage mean squared error $PMSE = \frac{1}{n} \left(\sum_{i=1}^{n} \frac{e_i^2}{z_i^2} \right) \times 100$

4. Percentage root mean squared error

$$PRMSE = \sqrt{\frac{1}{n} \left(\sum_{i=1}^{n} \frac{e_i^2}{z_i^2} \right)} \times 100$$

5. Accuracy = 100-Sum (abs (actual data-forecast data) / actual data*100) / n

The forecast error is less in lower proportion ratio of ANN prediction model as compared to other ratios. Though the selected model has been applied to forecast the inflows into Sewa River emanating from Himalayan region, yet this model can easily be modified to forecast the inflows into any reservoir system. This long-term forecasting of inflows would be of help in evaluating the performance of different operating policies for their adaptability and to check their suitability. Also, these long-term forecasts can be used as an effective input to the decision support for real-time operation of reservoir systems. Thus the ANN prediction model helps in effective utilization of available water, especially in a multipurpose perspective. Since it was found that, there is no need for frequent updating of the parameters, it can be used for forecasting the seasonal flows as well. The comparison between the

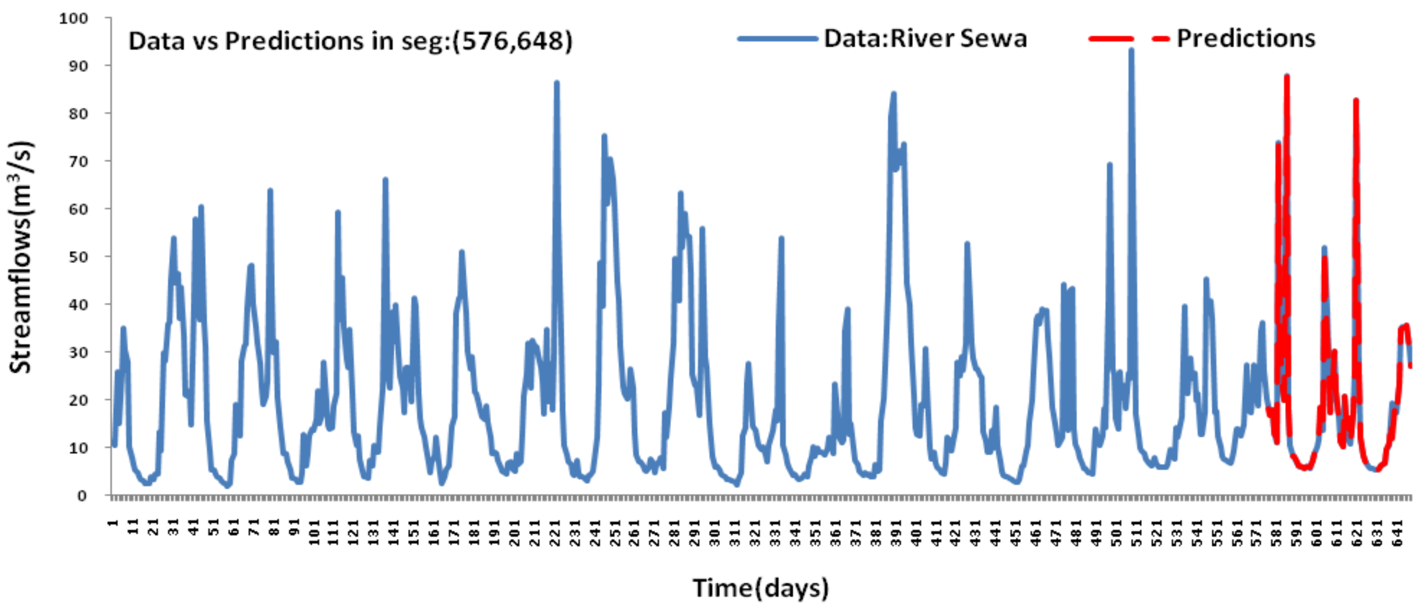

Figure 10. ANN prediction model obtained from 90:05:05 data set ratio by applying Levenberg-Marquardt algorithm.

forecasted inflows with that of the actual flows has been shown in Figure 10 which clearly reveals that neural network model has tracked the actual historical inflows data closely.

Conclusions

Stream flow forecasting in long term reservoir operation scheduling has been presented in this paper and the study indicates that, ANN prediction model has been found to be the workable tool for forecasting the inflows correctly especially in the long-term. In order to check the sensitivity of the ANN prediction model, three different proportions of ratio were analyzed, that is 60:20:20, 80:10:10 and 90:5:5 for training cross validation and testing. It was found from the experiment that lower proportion of ratio gives better result with higher accuracy. The data for training, validation and testing are chosen randomly from the given data set. Predicted flow results are quite cohesive to the exact data flows. The forecast inflows would be of help in evaluating the optimal real time reservoir operation policies and the generated synthetic series of ten days inflows can be used to provide a probabilistic framework for reservoir design and also can be used as an effective input to the decision support system for real-time operation of reservoir systems that results in increased power production and enhanced revenue earnings in the process of planning and management of a water resources system.

ACKNOWLEDGEMENT

The authors are grateful for the kind support rendered by the Electronics division of BHEL Bangalore, India for carrying out this work.

REFERENCES

Beale M, Demuth BH, Hagan MT (1996). Neural Network Design. PWS Publishing Company., Boston, Massachusetts, U.S.A. ISBN-13:9780534943325 ISBN: 0534943322.

Bender M, Simonovic S (1994). Time-Series Modeling for long range stream flow forecasting, J. Water Resour. Plann. Manage., 120(6): 857-870.

Bhushan B (2005). ABC of ABT, A Primer on Availability Tariff, Power Grid Corporation of India,http://www.nrldc.org/docs/abc_abt.pdf

Box GEP, Pierce DA (1970). Distribution of the autocorrelations in Autoregressive Moving Average Time Series Models, J. Am. Stat. Assoc., 65(332): 1509-1526.

Cheng X, Noguchi M (1996). Rainfall-runoff modelling by neural network approach. Proc of Int.Conf. On Water Resources and Environment Research: Towards the 21st Century, 29-31October, Kyoto, Japan, 2: 143-150.

Christensen GS, Soliman SA (1988). Optimization of the production of hydroelectric power systems with a variable head. J Optimization Theory Applications, Springer Netherlands, ISSN 0022-3239, 58(2): 301-317.

Dandy G, Maier H (1996). Use of Artificial Neural Networks for Real Time Forecasting of Water Quality. Proceedings of Int. Conf. on Water Resource and Environment Research (ICWRER), Vol 2, Oct. 29 -31, Kyoto, Japan.

Deshmukh SR, Doke DJ, Nerkar, YP (2008). Optimal Generation Scheduling under ABT using Forecasted Load and Frequency. Proceedings of POWERCON 2008 and 2008 IEEE Power India Conference (Joint International Conference), 12-15 October, pp.1-6.

Erik O, Hans JZ, Hajime S (2010). Macrocell Path-Loss Prediction Using Artificial Neural Networks, IEEE Trans. On Vehicular Technol., 59(6): 2735- 2747.

French MN, Krajewski WF, Cuykendall RR (1992). Rainfall forecasting in space and time using a neural network. J. Hydrol., 137(1-4):m 1-31.

Hagan MT, Menhaj MB (1994). Training feed forward networks with the Marquardt algorithm. IEEE Trans. Neural Networks, 5(6): 989-993.

Haykin S (1994). Neural Networks: A Comprehensive Foundation.

NewYork: Macmillan Publishing Company. ISBN 0-02-352781-7
Hipel KW (1985). Time Series Analysis in Perspective, Water Resour. Bull., 21(4): 609-623.
Hsu K, Gupta HV, Sorooshian S (1995). Artificial neural network modeling of the rainfall-runoff process. Water Resour. Res., 31(10): 2517-2530.
James EB, Gerald MB, David LR (2009). Elementary statistics for geographers. 3rd edition, Guilford Press. pp.186-187.
Jason H, Orlando DJ, Martin TH, (2009). Spurious Valleys in the Error Surface of Recurrent Networks-Analysis and Avoidance, IEEE Trans. Neural Networks, 20(4): 686 – 700.
Jayawardena AW, Fernando DAK (1996), Use of artificial neural networks in estimating evaporation, Proceedings of the International Conference on Water Resources and Environment Research: Towards the 21st Century, Kyoto, Japan, October 29-31, 1: 141-147.
Karunanithi N, Grenney WJ, Whitley D, Bovee K (1994). Neural Networks for River Flow Prediction, Journal of Computing in Civil Eng., ASCE, 8(2): 201-220.
Lorrai M, Sechi GM (1995). Neural nets for modelling rainfall-runoff transformations. Water Resour. Manage., 9: 299-313.
Maier H, Dandy G, (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, Environmental Modelling Software, 15: 101-124.
Raman H, Chandramouli V (1996).Deriving a general operating policy for reservoir using Neural networks, J. Water Resour. Plann Manage. ASCE, 122(5): 342-347.
Raman H, Sunil Kumar N (1995).Multivariate modelling of water resources time series using artificial neural networks, J. Hydrological Sci., 40(2): 145-163.
Salas JD (1993) Analysis and modeling of hydrologic time series, Handbook of Hydrology, D.R. Maidment, ed., Mc Graw-Hill., New York, pp. 19.1–19.71.
Salas JD, Tabios GQ, P Bartolini (1985). Approaches to multivariate modeling of water resources time series, Water Resour. Bull., 21(4): 683-708.
Sivanandam SN, Sumathi S, Deepa SN (2006). Introduction to Neural Networksusing MATLAB 6.0, Tata McGraw-Hill.
Smith J, Eli RN (1995). Neural network models of rainfall-runoff process, J. Water Resour. Plann. Manage., 121(6): 499-508.
Wood AJ, Wollenberg BF (1984). Power Generation, Operation, and Control. 2nd edition, John Wiley & Sons., New York. pp. 209-250.