Original Research Article

Simulation of Groundwater Level Using Recurrent Neural Network (RNN) in Raichur District, Karnataka, India

**Anandakumar[*], B. Maheshwara Babu, G.V. Srinivasa Reddy,
U. Satishkumar and Prasad Kulkarni**

*Department of SWE, College of Agricultural Engineering, UAS, Raichur-584104,
Karnataka, India*

*Corresponding author

A B S T R A C T

A proper design of the architecture of Artificial Neural Network (ANN) models can provide a robust tool in water resources modelling and forecasting. The performance of different neural networks in groundwater level forecasting was examined in order to identify an optimal ANN model for groundwater level forecast. The Devasugur nala watershed was selected for the study, located at northern part of Raichur district Karnataka and comes under middle Krishna river basin. Elman or Recurrent Neural Network (RNN) trained with Bayesian Regularization (BR), Levenberg Marquardt (LM) and Gradient Descent with Momentum and Adaptive Learning Rate Back propagation (GDX) algorithm models were developed. The results revealed that RNN with LM algorithm provided better prediction than the other models with highest correlation efficiency (0.8311) and lowest RMSE (0.9896) value during validation period. Overall it was observed that the ANN based algorithm was a better choice for the groundwater level forecasting.

Introduction

Groundwater usage in India accounts over 65 per cent for irrigation and 85 per cent for drinking water supplies. However, on current trends it is estimated that 60 per cent of groundwater sources will be in a critical state of degradation within the next twenty years. In the most seriously affected north-western states, recent satellite measurements indicate an average decline of 33 cm per year from 2002 to 2008 (Wyrwoll, 2012). Hence a clear understanding of various quantitative and qualitative aspects of groundwater resources in both space and time is an essential prerequisite for its judicious and sustainable management.

Almost all groundwater flow and transport models solve the relevant partial differential equation by finite difference or finite element method. These modelling methods are very much data and labour intensive and costly. So, in such cases empirical models remain an attractive alternative as these can provide useful results with less data and effort. In recent past, Artificial Neural Network (ANN) models are being applied increasingly to simulate the hydrological processes due to their better performance over the traditional

modelling techniques such as empirical models, statistical models (autoregressive, autoregressive moving average models) and physical based models.

ANN models are such 'black box' models with particular properties which are greatly suited to dynamic nonlinear system modeling. The advantages of ANN models over conventional simulation methods have been discussed in detail by French *et al.,* (1992). ANN applications in hydrology vary, from real-time to event based modeling. They have been used for rainfall-runoff modeling, precipitation forecasting and water quality modeling (Govindaraju and Ramachandra Rao, 2000). One of the most important features of ANN models is their ability to adapt to recurrent changes and detect patterns in a complex natural system. More concepts and applications of ANN models in hydrology have been discussed by the ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000). Neural networks have previously been applied for groundwater level prediction (Coulibaly *et al.,* 2001a; Coppola *et al.,* 2003; Daliakopoulos *et al.,* 2005; Nayak *et al.,* 2006 and Krishna *et al.,* 2008).

Materials and Methods

Study area description

Devasugur nala watershed was selected for the study which is located at northern part of Raichur district Karnataka. The observation well is located at Dept. of Mines and Geology premises with the latitude and the longitude 16° 12' 30" N, 77°21' 15" E. The monthly water table depth (below ground level) for 17 years *i.e.,* from January 1999 to March 2015 were collected from Department of Mines and Geology, Raichur, and used for analysis. The rainfall and meteorological data such as maximum and minimum temperature, relative humidity, wind speed and sunshine hour data were collected from Main Agricultural Research Station, Raichur, Karnataka and used for calculating the evapotranspiration using CROPWAT 8.0 which uses Penman-Monteith method.

Development of Elman or Recurrent neural network (RNN)

Deciding the suitable input vector is the one of the key parameter in ANN modelling, hence detailed correlation analysis between the independent and dependent variables were done. The correlation analysis helped to find out the possible input variable for the modelling, but it didn't give the exact lag values. The statistical parameters such as Auto-Correlation Function (ACF), Partial Auto-Correlation Function (PACF) and Cross-Correlation Function (CCF) were used to find out the significant lag values of input variables. Many researchers successfully used correlation analysis for selection of input variables (Nayak *et al.,* 2006 and Sasmita *et al.,* 2013).

The Elman Neural Network (ENN) is one type of the partial recurrent neural networks, which consists of a two-layer back propagation network with an additional feedback connection from the output of the hidden layer to its input. The advantage of this feedback path was that it allows the ENN to recognize and generate temporal patterns and spatial patterns. This means that after training, inter relations between the current input and internal states are processed to produce the output and to represent the relevant past information in the internal states. As a result, the ENN has been widely used in various fields which include classification, prediction, dynamic system identification (Fausett, 1994; Zealand *et al.,* 1999; ASCE, 2000 a, b; Maier and Dandy, 2000; Jain and Srinivasulu, 2004 and Sudheer *et al.,* 2002).

Elman Neural Network (ENN) has found numerous applications in such as time series prediction, system identification and adaptive control since it has powerful dynamic memories. Fully recurrent networks, introduced by Elman (1990), feed the outputs of the hidden layer back to itself. Partially recurrent networks start with a fully recurrent net and add a feed forward connection that bypasses the recurrence, effectively treating the recurrent part as a state memory. A typical recurrent network consisting of four input nodes, a hidden layer with three nodes and one output (Fig. 1).

Training the ANN

Determining the best values of all the weights is called training the ANN. In this study supervised training methodology was used to train the networks.

In a so-called supervised learning mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted so that the next iteration will produce a closer match between the desired and the actual output. Various learning methods for weight adjustments try to minimize the differences or errors between observed and computed output data.

The main objective of training (calibrating) a neural network is to produce an output vector $Y = (y_1, y_2, ..., y_p)$ that is as close as possible to the target vector (variable of interest or forecast variable) $T = (t_1, t_2, ..., t_p)$ when an input vector $X = (x_1, x_2, ..., x_p)$ is fed to the ANN. In this process, weight matrices W and bias vectors V are determined by minimizing a predetermined error function as indicated as

$$E = \sum_P \sum_p (y_i - t_i)^2 \quad (1)$$

Where, E is the error function; t_i is a component of the desired output T; y_i is the corresponding ANN output; p is the number of output nodes; and P is the number of training patterns.

The three most widely used training algorithms were used in order to identify the one which trains network more efficiently and those are Bayesian Regularization (BR), Levenberg Marquardt (LM) and Gradient Descent with Momentum and Adaptive Learning Rate Back propogation (GDX) algorithm.

Bayesian Regularization (BR) Algorithm

The Bayesian regularization is an algorithm that automatically sets optimum values for the parameters of the objective function. In the approach used, the weights and biases of the network are assumed to be random variables with specified distributions.

In order to estimate regularization parameters, which are related to the unknown variances, statistical techniques are being used. One feature of this algorithm is that it provides a measure of how many network parameters (weights and biases) are being effectively used by the network so that the function will not be over-fitted irrespective of the size of the network. Bayesian regularization has been effectively used in literature (Anctil *et al.,* 2004 and Coulibaly *et al.,* 2001a).

Levenberg-Marquardt (LM) Algorithm

The Levenberg–Marquardt method is a modification of the classic Newton algorithm for finding an optimum solution to a minimization problem. It uses an approximation to the Hessian matrix in the following Newton-like weight update

$$X_{k+1} = X_k - [J^T J + \mu I]^{-1} J^T e \quad (2)$$

Where, x the weights of neural network, J the Jacobian matrix of the performance criteria to be minimized, μ a scalar that controls the learning process and e the residual error vector. When the scalar μ is zero, Eq. (4) is just the Newton's method, using the approximate Hessian matrix. Levenberg–Marquardt has great computational and memory requirements and thus it can only be used in small networks (Maier and Dandy, 1998). Never the less, many researchers have been successfully using it (Anctil *et al.,* 2004; Coulibaly *et al.,* 2000 and Coulibaly *et al.,* 2001 a).

Gradient descent with momentum and adaptive learning rate back propogation Algorithm (GDX)

This method uses back propagation to calculate derivatives of performance cost function with respect to the weight and bias variables of thenetwork. Each variable is adjusted according to the gradient descent with momentum. Acting like a low pass filter, momentum allows the network to ignore small features in the error surface. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable. For each step of the optimization, if performance decreases the learning rate is increases. This is probably the simplest and most common way to train a network (Haykin, 1999).

Network Architecture

The network geometry is generally highly problem oriented in order to get optimal network geometry trial and error procedure is adopted. The numbers of nodes in the input layer were decided based on the inputs to the model. The number of hidden neurons in the

network, which is responsible for capturing the dynamic and complex relationship between various input and output variables, was identified by various trial and error methods. The trial and error procedure started with one hidden neuron initially, and it has been increased up to 20 neurons. For each set of hidden neurons, the network was trained with input datasets in batch mode to minimize the mean square error at the output layer.

The transfer functions of hidden and output layers have been considered as log sigmoid and pure linear respectively in the both training and validation of the ANN model.Various internal parameters used in the ANN model like learning rate, momentum coefficient, scalar μ and combination of transfer functions for hidden and output layer were also found out by trial and error. MATLAB 2012a software was used for analysis.

Performance evaluation of ANN Models

The whole data length is divided into two sets based on statistical properties of the time series such as mean and standard deviation, in that one is used for calibration (training) and another for validation of ANN model. The performance during calibration and validation is evaluated by using statistical parameters. They are Correlation coefficient (CC) Coefficient of Efficiency (CE), Root Mean Square Error (RMSE), Explained Variance (EV) and regression analysis.

Results and Discussion

The ACF and PACF of groundwater level, the CCF between monthly mean evapotranspiration with monthly mean water table depth and the CCF between monthly total rainfalls with monthly mean water table depth were calculated and presented from Figure 2–5.

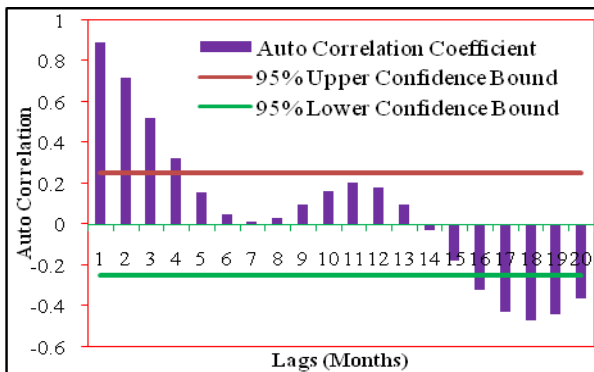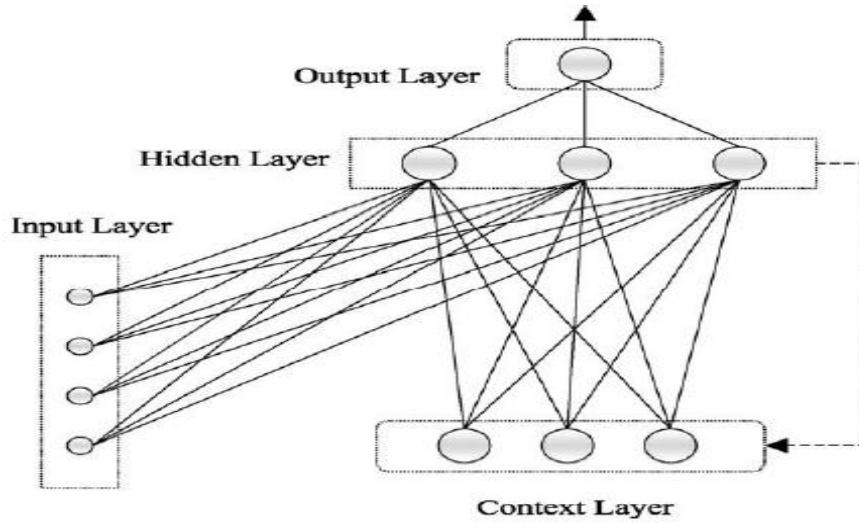**Fig.1** A typical elman or recurrent neural network

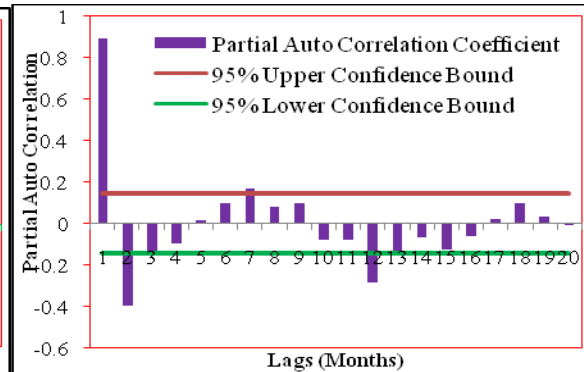**Fig. 2 ACF of groundwater level for Raichur observation well**

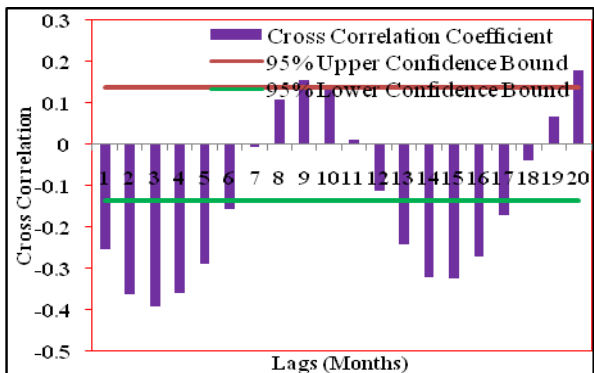**Fig. 3PACF of groundwater level for Raichur observation well**

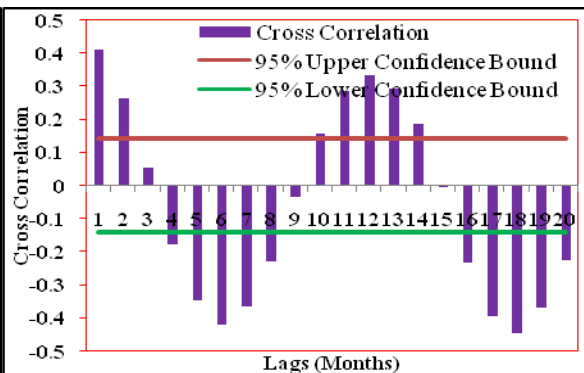**Fig. 4 CCF between RF and GWL for Raichur observation well**

Fig. 5 CCF between ET and GWL for Raichur observation well

Fig.6 Comparison of actual and predicted water table depth for RNN-BR model during calibration and validation period for Raichur well
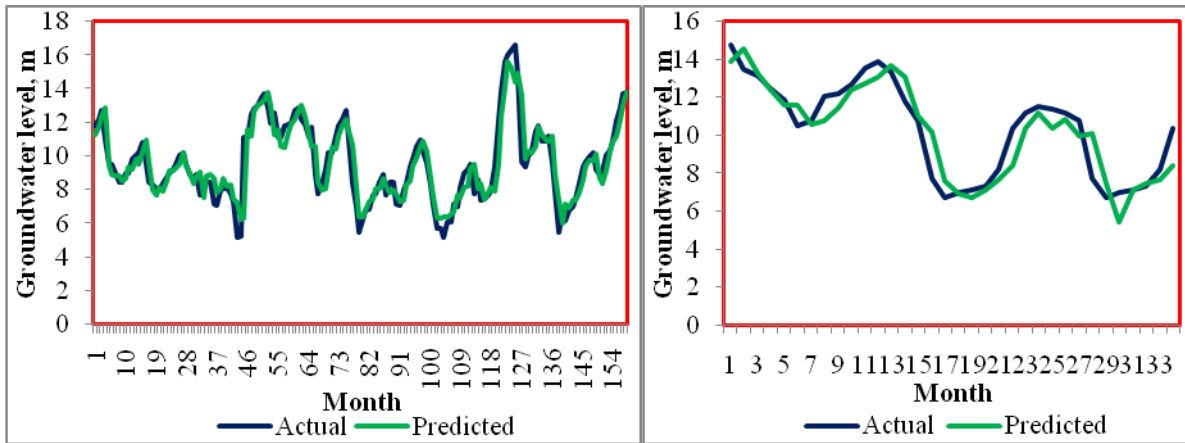

Fig.7 Comparison of actual and predicted water table depth for RNN-LM model during calibration and validation period for Raichur well
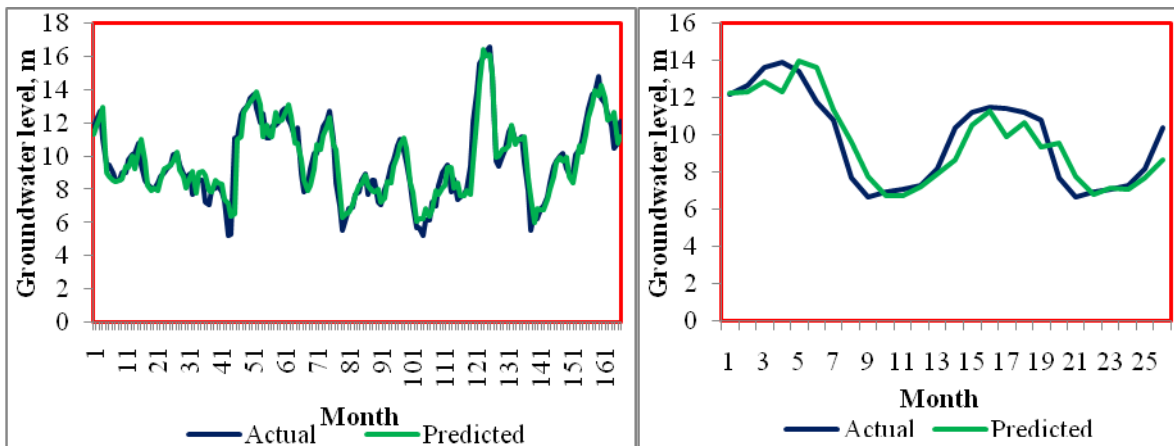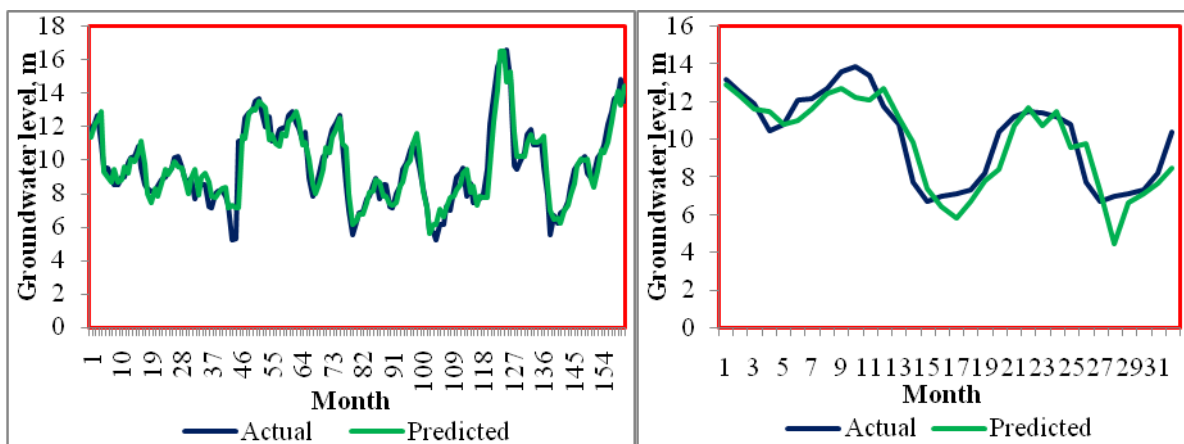

Fig.8 Comparison of actual and predicted water table depth for RNN-GDX model during calibration and validation period for Raichur well
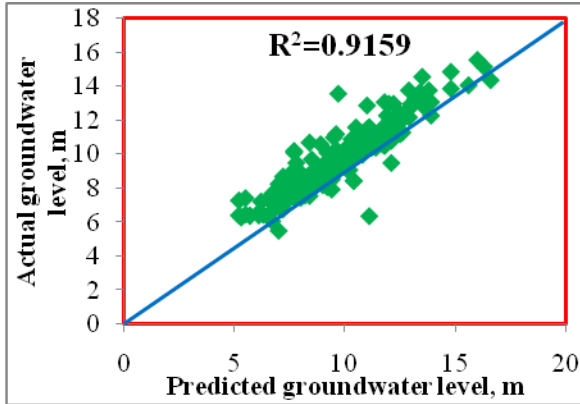
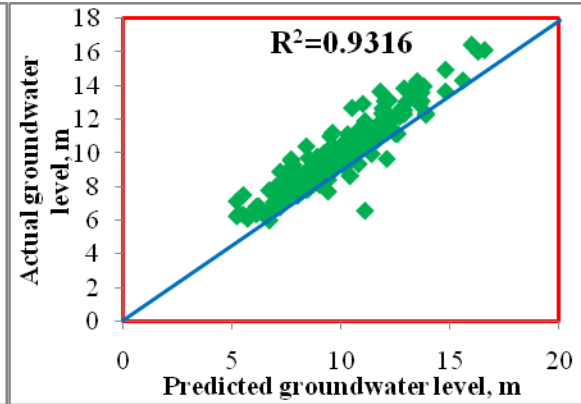**Fig. 9 Regression graph for RNN-BR model for Raichur observation well**

Fig. 10 Regression graph for RNN-LM model for Raichur observation well

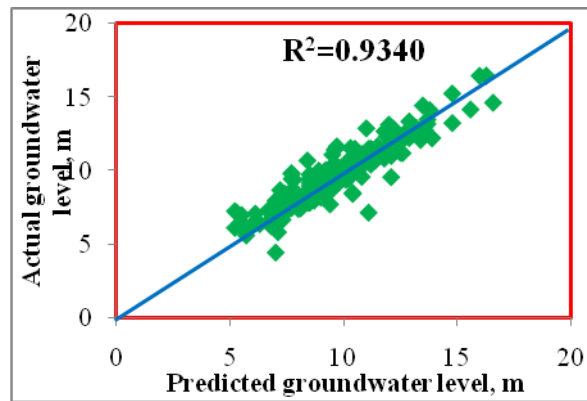**Fig.11** Regression graph for RNN-GDX model for Raichur observation well

Table.1 Goodness of fit statistics of different algorithms for developed ANN models

ANN with Training Algorithm	ANN Architecture	CC Cal	CC Val	CE Cal	CE Val	RMSE Cal	RMSE Val
RNN-BR	3-5-1	0.9159	0.9143	0.8389	0.8301	0.9332	1.0196
RNN-LM	3-5-1	0.9361	0.9136	0.8764	0.8306	0.8311	0.9896
RNN-GDX	3-10-1	0.934	0.9052	0.8752	0.7866	0.845	1.0861

The higher correlation values of ACF of Groundwater Level (GWL), PACF (GWL), CCF of Rainfall (RF) with (GWL) and the CCF of Evapotranspiration (ET) with (GWL) were 0.89 for one month lag time, 0.89 for one month lag time, -0.39 for three month lag time and 0.41 for three month lag time respectively.

Comparison of Neural Networks

In order to identify the best suitable model to forecast the groundwater level in the observation well located at the study area, ANN models were compared. Elman or Recurrent Neural Network (RNN) was used for simulation and modelling of groundwater

3364

level fluctuation. Three different training algorithms namely Bayesian Regularization (BR), Gradient Descent with Momentum and Adaptive Learning Rate Back Propagation (GDX) and Levenberg-Marquardt (LM) were used to train the RNN. Hence three models namely RNN-BR, RNN-LM and RNN-GDX were developed for each observation well individually.

The data from January-1999 to March-2015 (195 months) was considered for the development of the model. Out of 195 months dataset, 192 month dataset was available for analysis considering maximum of three month time lag for rainfall series. The whole dataset was divided into two sets for the calibration and validation of the ANN models.The model with the most correlation coefficient, with the highest correlation efficiency and with least root mean square error in both calibration and validation period have been selected as a best fit model and the selected best fit model. The performance of the different ANN models during calibration and validation with the input combination derived from statistical procedure were indicated (Table 1).

The values of statistical parameters of different ANN models (Table 1) indicates coefficient of efficiency (CE) which evaluates how far the network would explain the total variance of data was low (0.7866) during validation period for the RNN-GDX model and was high (0.8764) during calibration period for the RNN-LM model. The variation of root mean square error (RMSE) statistics, a measure of residual variance which illustrated the global goodness of fit between the computed and observed water table depths was low (0.8311) during calibration period for the RNN-LM model and was high (1.0861) during validation period for the RNN-GDX model. The correlation coefficient (CC) which evaluated the linear correlation between the computed and observed water

table depths was high (0.9361) during calibration period for RNN-LM model and low (0.0.9052) during validation period for RNN-GDX model. The explained variance (EV) was high (0.9361) for RNN-LM and low (0.9087) for RNN-BR models during the calibration period.

The key parameters in the selection of the best ANN model were the correlation coefficient (Fig. 6 to 8) and regression analysis between actual and predicted water table depth during the calibration and validation period (Fig. 9 to 11).

Comparison of actual and predicted water table depth computed during the calibration and validation period shown in Figure 6 to 8. Regression analysis between actual and predicted groundwater level for the different observation wells located at the Raichur for the different ANN models presented in Figure 9 to 11 and by observing the figures it was confirmed that the predicted water table depths followed the actual water table pattern in all the models both in calibration as well as in validation period but RNN-BR (Fig. 6) was unable to follow the actual water table pattern at 48th, 102th, 103th 125th and 139th month during calibration and during initial periods as well as at 17th and 18th month in the validation and RNN-GDX (Fig. 4) at 46th, 104th, 108th, 128th, 136th and 137th month during calibration and in validation it was indicated lot of variations during initial periods. The RNN-LM model was found the much closer relationship between predicted and the actual water table pattern in both calibration and validation period with less or no variations. Figure 9 to 11 indicated the regression analysis between actual and predicted groundwater level, for the RNN-LM model all the actual and predicted groundwater level values lies near to the straight line as compared to the other models, Hence RNN-LM model was selected as best ANN

model for the forecasting of water table depthof the observation well located at Raichur.

The optimum ANN based model proposed in this study shows very promising results. The three ANN models were developed and the effectiveness of the models developed for the study area was assessed using statistical indicators as well as visual comparison of observed and predicated groundwater level. The RNN-LM were found to be most efficient for monthly groundwater level forecasting for the study area, followed by RNN-BR and RNN-GDX models. Overall it can be concluded that ANN technique can be used effectively for the prediction of groundwater level.

References

ASCE Task Committee on application of artificial neural networks in hydrology. 2000a. Artificial neural networks in hydrology-I: preliminary concepts. *J. Hydrol. Engg.*, 5(2): 115-123.

ASCE Task Committee on application of artificial neural networks in hydrology. 2000b, artificial neural networks in hydrology-II: hydrologic applications. *J. Hydrol. Engg.*, 5(2): 124-137.

Chiew, F. H. S., Stewardson, M. J. and McMahonn, T. A., 1993, Comparison of six rainfall-runoff modeling approaches. *J. Hydrol.*, 147: 1-36.

Coppola, E., Szidarovszky, F., Poulton, M. and Charles, E., 2003, Artificial neural network approach for predicting transient water levels in a multilayered groundwater system under variable state, pumping and climate conditions. *J. Hydrol. Engg.*, 8(6): 348-360.

Coulibaly, P., Anctil, F. and Bobee, B., 2000, Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *J. Hydrol.*, 230: 244-257.

Coulibaly, P., Anctil, F., Aravena, R. and Bobee, B., 2001a, Artificial neural network modeling of water table depth fluctuations. *Water Resour. Res.*, 37(4): 885-896.

Daliakopoulos, I. N., Coulibaly, P. and Tsanis, I. K., 2005, Groundwater level forecasting using artificial neural network. *J. Hydrol.*, 309: 229-240.

Elman, J. L., 1990, Finding structure in time. *CognitiveSci.*, 14: 179-211.

Fausett, L., 1994, *Fundamentals of Neural Networks*. Prentice Hall Publisher, Englewood cliffs, N.J., p. 150-280.

French, M. N., Krajewski, W. F. and Cuykendall, R. R., 1992, Rainfall forecasting in space and time using neural network. *J. Hydrol.,* 137: 1-31.

Govindaraju, R. S. and Ramachandra Rao, A., 2000, *Artificial neural network in hydrology*. Kluwer Academic Publishing, Amsterdam, Netherlands, p. 68-73.

Haykin, S., 1999, *Neural networks, A Comprehensive Foundation*. Prentice Hall Publishers, Englewood Cliffs, N.J., p. 115-122.

Jain, A. and Srinivasulu, S., 2004, Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques. *Water Resour. Res.*, 40: 1-12.

Krishna, B., SatyajiRao, Y. R. and Vijaya, T., 2008, Modelling groundwater levels in an urban coastal aquifer using artificial neural networks. *Hydrol. Process*, 22: 1180-1188.

Maier, H. R. and Dandy, G. C., 1998, Understanding the behavior and optimizing the performance of back-propagation neural networks: an

empirical study. *Environ. Model. Softw.,* 13: 179-191.

Nayak, P. C., SatyajiRao, Y. R. and Sudheer, K. P., 2006, Groundwater level forecasting in a shallow aquifer using artificial neural network approach. *Water Resour. Manage.*, 20: 77-90.

Sasmita, S. and Madan, K. J., 2013, Application of hybrid neural network in modeling water table fluctuation. *J. Groundwater Research.* 2(1): 9-19.

Sudheer, K. P., Nayak, P. C. and Ramasastri, K. S., 2002, Improving peak flow networks: A neural network modeling approach for hydrological processes. *Water Resour. Res.*, 42: 1-14.

Wyrwoll, P., 2012, India's groundwater crisis. *Global Water Forum*, Canberra, Australia, p. 1228

Zealand, C. M., Burn, D. H. and Simonovic, S.P., 1999, Short term stream flow forecasting using artificial neural networks. *J. Hydrol.*, 214(4): 32-48.