Neural Comput & Applic (2010) 19:1251–1263
DOI 10.1007/s00521-010-0360-1

ORIGINAL ARTICLE

Hybrid neural modeling for groundwater level prediction

Nikunja B. Dash · Sudhindra N. Panda · Renji Remesan · Narayan Sahoo

Received: 8 June 2009 / Accepted: 30 March 2010 / Published online: 11 April 2010

Abstract The accurate prediction of groundwater level is important for the efficient use and management of groundwater resources, particularly in sub-humid regions where water surplus in monsoon season and water scarcity in non-monsoon season is a common phenomenon. In this paper, an attempt has been made to develop a hybrid neural model (ANN-GA) employing an artificial neural network (ANN) model in conjunction with famous optimization strategy called genetic algorithms (GA) for accurate prediction of groundwater levels in the lower Mahanadi river basin of Orissa State, India. Three types of functionally different algorithm-based ANN models (viz. back-propagation (GDX), Levenberg–Marquardt (LM) and Bayesian regularization (BR)) were used to compare the strength of proposed hybrid model in the efficient prediction of groundwater fluctuations. The ANN-GA hybrid modeling was carried out with lead-time of 1 week and study mainly aimed at November and January months of a year. Overall, simulation results suggest that the Bayesian regularization model is the most efficient of the ANN models tested for the study period. However, a strong correlation between the observed and predicted groundwater levels was observed for all the models. The results reveal that the hybrid GA-based ANN algorithm is able to produce better accuracy and performance in medium and high groundwater level predictions compared to conventional ANN techniques including Bayesian regularization model. Furthermore, the study shows that hybrid neural models can offer significant implications for improving groundwater management and water supply planning in semi-arid areas where aquifer information is not available.

Keywords Training algorithms · Prediction · Genetic algorithm · ANN · Winter · Groundwater · India

1 Introduction

Water scarcity in non-monsoon season and water logging during monsoon season are the two major problems of agricultural fields in the lower Mahanadi river basin in Orissa State, India. The basin is located in the coastal alluvial tract and, moreover, at the tail end of the canal irrigation system. The climate of the basin is sub-humid with average annual rainfall of 1,500 mm; two-thirds of which concentrates during monsoon season (June to September). Siltation and weed growth in the drainage line and at the mouth of drains cause drainage congestion in agricultural lands during monsoon season. Prolonged submergence of rice during monsoon season leads to crop failure. Modeling groundwater level with lead-time at different seasons of a year would help in alleviation of these problems. But for this purpose, fitting a physical model is not possible because of the unavailability of essential data that affect the physical system of the study area. That is why modeling approaches like artificial neural network (ANN) and genetic algorithm (GA) are looked for,

N. B. Dash · S. N. Panda
Agricultural and Food Engineering Department, Indian Institute of Technology, Kharagpur 721302, India

R. Remesan (✉)
Water and Environmental Management Research Centre, Department of Civil Engineering, University of Bristol, Lunsford House, Cantocks Close, Clifton, Bristol BS8 1UP, UK
e-mail: renji.remesan@bristol.ac.uk

N. Sahoo
Water Technology Center for Eastern Region, Bhubaneswar 751023, India

which require least amount of input data and produce results with reasonable accuracy.

A neural network model is a mathematical model whose architecture is essentially analogous to the learning capability of the human brain. The advantage of ANNs in engineering and practical applications lies in their ability to learn and capture information from data that describe the behavior of a real system [14]. But GA is an unorthodox search and optimization technique, which combines an artificial survival of the fittest with genetic operators to find an optimal solution from a pool of possible solutions. Starting with an initial population of chromosomes (solutions), genetic operators (selection, crossover and mutation operators) are applied repeatedly from one generation to the next to find a global optimal solution.

Maier and Dandy [20] reviewed 43 papers dealing with the use of neural network models for prediction and forecasting of water resources variables. ASCE Task Committee on Application of ANNs in Hydrology [1] examined the role of ANNs in various branches of hydrology. ASCE Task Committee on application of ANNs in Hydrology [2] presented description of various aspects of ANNs, some guidelines on their usages, a brief comparison of the nature of ANNs and other modeling philosophies in hydrology. The advantages of ANN models over conventional simulation methods have been discussed in detail by French et al. [11].

In recent years, ANNs have found many successful applications in the field of groundwater level modeling. Yang et al. [32, 34] simulated the fluctuations of water table depths and drain outflows well in real time with help of ANN modeling approach. Yang et al. [31] utilized ANN to predict water table elevations in subsurface-drained farmlands, as influenced by daily rainfall and potential evapotranspiration rates. Coulibaly et al. [6] calibrated three functionally different ANN models using a relatively short length of groundwater level records and other related hydro-meteorological data to simulate groundwater table fluctuation. Lallahem et al. [18] showed the feasibility of using ANN to estimate ground water level in piezometers implanted in an unconfined chalky aquifer. Daliakopoulosa et al. [9] simulated groundwater level with acceptable predictions up to 18 months ahead. Application of ANN to problem of irrigation and drainage was investigated by Yang et al. [33]. All these successful results have encouraged us to apply ANNs to the present case of modeling.

Many recent studies have efficiently used GA to solve various groundwater management problems [3, 24]. Bhattacharjya and Datta [4] have developed a simulation–optimization model by linking the ANN model with a GA-based optimization model for solving multiple objective saltwater management problems. Parasuraman and Elshorbagy [23] have used regular GA-trained ANN model for computing dynamics of high flows. Wu and Chau [30] have employed a GA-based ANN for flood forecasting. However, these reviews suggest that not much work has been carried out till date on the prediction of groundwater levels using hybrid modeling schemes. Thus, the main objective of the present study is to construct a hybrid neural model that combines two main artificial intelligence techniques viz. ANN and GA for efficient prediction of groundwater levels in the lower Mahanadi river basin of Orissa State, India. The study compared the hybrid model simulated results with that of conventional and functionally different ANN models viz. back-propagation (GDX), Levenberg–Marquardt (LM) and Bayesian regularization (BR). This paper organized as follows (1) Sect. 2: Study area and data description, (2) Sect. 3: Artificial neural network (ANN) and training algorithms, (3) Sect. 4: Genetic algorithm, (4) Sect. 5: ANN model development, (5) Sect. 6: ANN-GA hybrid model development, (6) Sect. 7: Performance evaluation of the models, (7) Sect. 8: Results and discussions and (8) Sect. 9: Conclusions.

2 Study area and data description

The study area is a part of the lower Mahanadi river basin extending over 600 km^2, covering from 19°40′ to 20°20′ north latitude and from 85°24′ to 86°37′ east longitude. Location map of the study area in the Mahanadi river basin is shown in the Fig. 1. The study area comprises of mostly agricultural fields, natural drains/rivers, artificial surface drains, and canals. Rice is the major crop grown in monsoon and non-monsoon season. In this area, 26 meteorological stations were selected for the present study with respect to availability and continuity of the required data. Groundwater level data for these 26 meteorological stations were collected for 10 years (1993–2002) from the Central Ground Water Board, Govt. of India, Bhubaneswar, Orissa. Monthly rainfall data for this period were collected from the Board of Revenue, Govt. of Orissa, Cuttack, Orissa.

The groundwater level data were available for 2 months (i.e., November, and January) of a year during the study period. Antecedent information of aggregated rainfalls of different months, antecedent information of groundwater level, and groundwater level of previous year were taken as the three inputs in the modeling process, whereas groundwater level at the present time-step is taken as the only output. Groundwater level of previous time-step and the previous year were taken as two inputs to consider the effect of other important physical parameters, not being taken into account as inputs. Although this type of structure does not explicitly represent cause and effect relationships between input and output, still they predict with same or

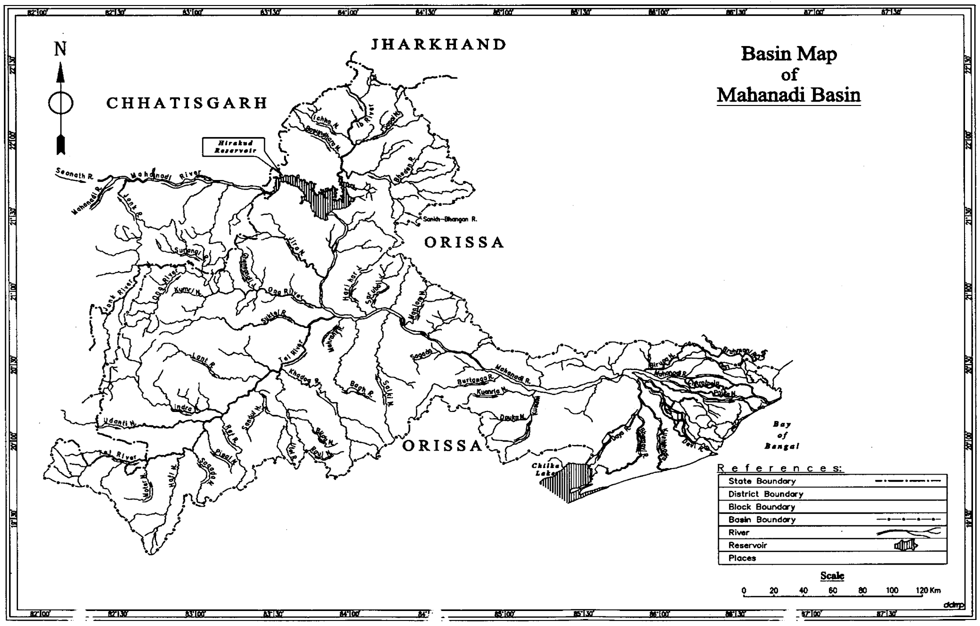

Fig. 1 Location map of the study area in the lower Mahanadi river basin, India

even better accuracy than cause-effect models for purposes of operational forecasting [13]. The groundwater level data for January and November months were considered for analysis in this study because of various reasons. The main reason was availability of continuous and reliable data for longer duration (10 years) from several observation points throughout the study region. The winter season remains in India generally from November to February and in some years it extends to month of March. The summer months in India lasts from about March to June. It is important to analyze the pre-summer groundwater fluctuation to make better water management guidelines to be adopted during severe summer months. The data analysis on early winter month November and later January months would give a better idea of influence of winter rains on groundwater replenishment. The success of major winter crops (e.g., wheat) is largely depending on the winter rains. Kelkar et al. [17] has noted the influence of high variability of winter rains on winter crop in India. They noted that winter rains are delayed and highly inconsistent in recent years because of that the wheat crop has been very poor for the last 10–15 years, and 15–20% of the agricultural land is left barren. The inconsistency in winter showers has made irritation and groundwater pumping inevitable for success of crop. This highlights the need for proper quantification of groundwater fluctuation during winter months to adopt better irrigation and water management guidelines. Groundwater data of November (early winter season) and January (mid or late winter month) have great significance both from agricultural and hydrological point of view in the study area. As winter cropping is one of the primary agricultural seasonal practices in the study area, a preliminary knowledge of the groundwater levels is of much importance to the farmers to provide irrigation guidelines. In addition, it gives an idea about effect of rainfall on the ground water level in the post-monsoon period. In both the months, the groundwater level data were collected at the end of the week, and rainfall data were collected in the beginning of the week. Thus, in both ANN and ANN-GA hybrid modeling, the lead-time for the prediction in the month of November and January is 1 week.

3 Artificial neural network (ANN) and training algorithms

Artificial neural network (ANN) is a massively parallel distributed information processing system that has certain performance characteristics resembling biological neural networks of human brain [15]. ANN is characterized by its architecture that represents the pattern of connection between nodes, its method of determining the connection weights, and the activation function [10].

The nodes were arranged in layers: an input layer, an output layer, and a hidden layer as shown in the Fig. 2. Each neuron in a specific layer is connected to many other neurons via weighted connections. The weights determine the strength of the connections between interconnected neurons. The neurons in the hidden and output layers transform their respective inputs to outputs through two separate stages. First, for each neuron, its input was multiplied by its corresponding weight. Secondly, the total sum of these products plus a constant called bias yielded the node output in the hidden and output layers.

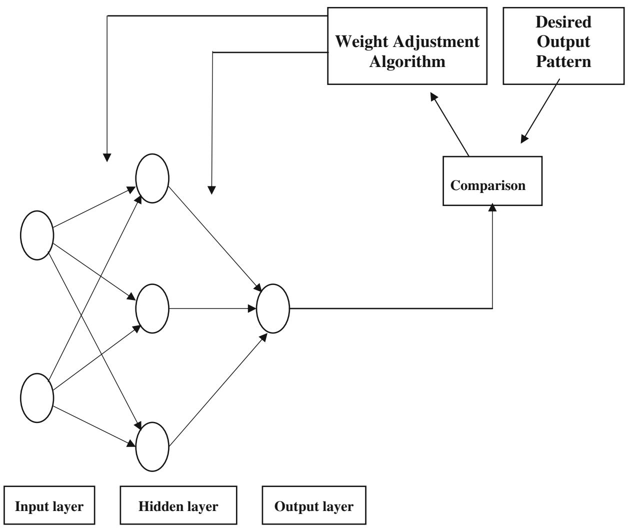

Fig. 2 Supervised training used in ANN modeling

3.1 Feed forward network

Feed forward network (FNN) is the most commonly used network in ANN modeling. Multilayer FNN can have more than one hidden layer. But it has been shown that ANNs with one hidden layer can approximate any function, given that sufficient degrees of freedom (i.e., connection weights) are provided [16]. Multilayer FNN training (supervised type of training) consists of providing input–output examples to the network and minimizing the objective function (i.e., error function) using either a first-order or a second-order optimization method. Here, the model output was compared with the desired output (which acts as a supervisor), and the error was calculated. This error was then back-propagated with the help of a weight-adjustment algorithm and the connection weights between the layers are adjusted so that in next iteration the error became less. This cycle continued till the predicted output become mostly equal to the observed output within the modeling constraints. This whole process of supervised training is presented in the Fig. 2.

3.2 Back-propagation algorithm

The term back-propagation refers to the process by which derivatives of network error, with respect to network weights and biases, can be computed. In back-propagation algorithm, the weights are moved in the direction of the negative gradient, i.e., in the direction in which the performance function decreases most rapidly. Thus, as the training proceeded, the weights were constantly adjusted toward the minimum. Figure 3 represents the schematic representation of a multilayer network for back-propagation

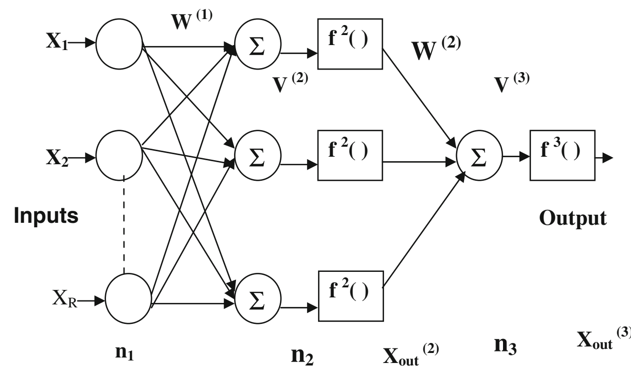

Fig. 3 Schematic representation of back-propagation algorithm

training. Here, in multilayer network, the serial number of the layers was used to determine the superscript on the weight matrices.

Output of individual node after summation operation,

$$V = W_1X_1 + W_2X_2 + \cdots W_RX_R + b \tag{1}$$

Output of individual node after passing through transfer function,

$$X_{\text{out}} = f(V) = f(W_1X_1 + W_1X_1 + \cdots W_RX_R + b) \tag{2}$$

For output layer

Weight change in output (second) layer,

$$\Delta W_{ji}^{(3)} = -\mu^{(3)}\frac{\partial E_q}{\partial W_{ji}^{(3)}} \tag{3}$$

Since (3) was not an explicit function of the weight in the hidden layer, the first partial derivatives of E_q were evaluated with respect to the weights using the chain rule and were moved in the steepest descent direction. In the back-propagation training, minimization of E was attempted using the steepest descent method and computing the gradient of the error function by applying the chain rule on the hidden layers of the FNN [26].

$$\Delta W_{ji}^{(3)} = -\mu^{(3)}\frac{\partial E_q}{\partial V_j^{(3)}}\frac{\partial V_j^{(3)}}{\partial W_{ji}^{(3)}} \tag{4}$$

where

$$\frac{\partial V_j^{(3)}}{\partial W_{ji}^{(3)}} = \frac{\partial}{\partial W_{ji}^{(3)}}\left(\sum_{h=1}^{n_2} W_{ji}^{(3)} X_{\text{out},h}^{(2)}\right) = X_{\text{out},i}^{(2)} \tag{5}$$

$$\begin{aligned}\frac{\partial E_q}{\partial V_j^{(3)}} &= \frac{\partial}{\partial V_j^{(3)}}\left\{\frac{1}{2}\sum_{h=1}^{n_2}\left[d_{qh} - f\left(V_h^{(3)}\right)\right]^2\right\}\\ &= -\left[d_{qj} - f\left(V_j^{(3)}\right)\right]g\left(V_j^{(3)}\right)\\ &= -\left[d_{qj} - X_{\text{out},h}^{(3)}\right]g\left(V_j^{(3)}\right)\\ &= -\delta_j^{(3)}\end{aligned} \tag{6}$$

So,

$$\Delta W_{ji}^{(3)} = \mu^{(3)} \delta_j^{(3)} X_{\text{out},i}^{(2)} \tag{7}$$

$$W_{ji}^{(3)}(K+1) = W_{ji}^{(3)}(K) + \mu^{(3)} \delta_j^{(3)} X_{\text{out},i}^{(2)} \tag{8}$$

Briefly, back-propagation algorithm can be described as follows:

For hidden layer

Weight change in hidden (first) layer,

$$\Delta W_{ji}^{(2)} = -\mu^{(2)} \frac{\partial E_q}{\partial W_{ji}^{(2)}} \tag{9}$$

$$\Delta W_{ji}^{(2)} = -\mu^{(2)} \frac{\partial E_q}{\partial V_j^{(2)}} \frac{\partial V_j^{(2)}}{\partial W_{ji}^{(2)}} \tag{10}$$

where

$$\frac{\partial V_j^{(2)}}{\partial W_j^{(2)}} = \frac{\partial}{\partial W_{ji}^{(2)}} \left(\sum_{h=1}^{n_1} W_{ji}^{(2)} X_{\text{out},h}^{(1)} \right) = X_{\text{out},i}^{(1)} \tag{11}$$

$$\begin{aligned} \frac{\partial E_q}{\partial V_j^{(2)}} &= \frac{\partial}{\partial X_{\text{out},j}^{(2)}} \left\{ \frac{1}{2} \sum_{h=1}^{n_3} \left[d_{qh} - f\left(\sum_{p=1}^{n_2} W_{hp}^{(3)} X_{\text{out},p}^{(2)} \right) \right]^2 \right\} \frac{\partial X_{\text{out},j}^{(2)}}{\partial V_j^{(2)}} \\ &= -\left[\sum_{h=1}^{n_3} \left[\left(d_{qh} - X_{\text{out},h}^{(3)} \right) g\left(V_h^{(3)} \right) \right] W_{hj}^{(3)} \right] \times g\left(V_j^{(2)} \right) \\ &= -\left(\sum_{h=1}^{n_3} \delta_h^{(3)} W_{hj}^{(3)} \right) g\left(V_j^{(2)} \right) \\ &= -\delta_j^{(2)} \end{aligned} \tag{12}$$

So,

$$\Delta W_{ji}^{(2)} = \mu^{(2)} \delta_j^{(2)} X_{\text{out},i}^{(1)} \tag{13}$$

$$W_{ji}^{(2)}(K+1) = W_{ji}^{(2)}(K) + \mu^{(2)} \delta_j^{(2)} X_{\text{out},i}^{(1)} \tag{14}$$

$$W_{ji}^{(s)}(k+1) = W_{ji}^{(s)}(k) + \mu^{(s)} \delta_j^{(s)} X_{\text{out},j}^{(s)} \tag{15}$$

For output layer

$$\delta_j^{(s)} = \left(d_{qh} - X_{\text{out},j}^{(s)} \right) g\left(V_j^{(s)} \right) \tag{16}$$

For hidden layer

$$\delta_j^{(s)} = \left(\sum_{h=1}^{n_{s+1}} \delta_j^{(s+1)} W_{hj}^{(s+1)} \right) g\left(V_j^{(s)} \right) \tag{17}$$

where μ is the learning coefficient; K is the epoch; q is the pattern feed to the network; S is any layer; g is the first derivative of transfer function; δ is the incremental error; W_{ji} is the weight between jth target neuron and ith source neuron. However, it was very easy for the training process to be get trapped in a local minimum during simple back-propagation training process. This problem was prevented by adding a momentum term to the weight change to permit larger learning rates and using a variable (adaptive) learning parameter so that chances of being trapped in local minima decreases considerably. This modified algorithm is known as back-propagation algorithm with momentum and variable learning rate or GDX algorithm.

3.3 Levenberg–Marquardt algorithm

The Levenberg–Marquardt algorithm is a second-order optimization algorithm and less easily trapped in local minima than other optimization algorithms [5]. But it has great computational and memory requirements and thus it can only be used in small networks [19]. In this algorithm, Hessian matrix (H) is approximated in terms of Jacobian matrix (J) following Newtonian update in following ways.

$$x_{k+1} = x_k - \left[J^T J + \mu I \right]^{-1} J^T e \tag{18}$$

$$J^T J = H \tag{19}$$

$$J = \nabla E \tag{20}$$

$$J^T e = g \tag{21}$$

where x is the weight of neural network; μ is the learning parameter that controls the learning process, e is a matrix of network errors, and g is the gradient of error surface.

3.4 Bayesian regularization algorithm

Bayesian regularization algorithm is a modification of the Levenberg–Marquardt algorithm to produce networks that generalize well. It reduces the difficulty in determining the optimum network architecture by computing the optimal regularization parameters in an automated fashion, thus prevents the problem of over-fitting and eliminates the guesswork required in determining the optimum network size. It is also known as automated regularization algorithm and provides a measure of how many network parameters (i.e., weights and biases) are being effectively used by the network. This effective number of parameters should remain approximately the same, no matter how large the total number of parameters in the network becomes. Bayesian regularization algorithm has been effectively used in research [5–8].

4 Genetic algorithm

Genetic algorithm (GA) is a powerful optimization technique that is based upon the underlying principles of natural evolution and selection. GA has the ability to search effectively for an optimum in several directions simultaneously. Another advantage of GA is that they can be parallelized easily [25]. The methodology of GA involves three basic operations, e.g., selection, crossover, and mutation [12].

Selection (who survives) of individuals was based on the fitness of the individuals with respect to an objective function. Individuals with high fitness values (i.e., better solutions to the problem) will have a higher probability of surviving and entering the mating population. The selection of the parents relied on the most popular *roulette-wheel selection method*. In this method, an individual with better fitness value was associated with a larger roulette-wheel slot size, which corresponds to a better chance to survive or to reproduce than the others. In crossover, best material from two selected parent strings is exchanged and a new offspring is created. Although reproduction and crossover effectively search and recombine extant notions, occasionally they may lose some potentially useful genetic material. The mutation operator protects against such an irrecoverable loss. In this process, mutation operator randomly flips (alters) some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small (e.g., 0.001). Thus, the mutation (random genetic changes) process helps to overcome trapping at local optima.

In genetic algorithm, each iteration is known as a *generation* and the entire set of generations is known as *run*. In each generation, in GA produces a new set of possible solutions for a given problem. First, the process of GA-based search was initiated by creation of a random set of initial populations. Then, performance of the strings, called as fitness, was evaluated with the help of some functions, representing the constraints of the problem. Depending on the fitness of the chromosomes, they were selected for a subsequent genetic process. After selection of the population strings, the crossover and mutation operations were carried out. Offspring produced after crossover and mutation operations form the next population to be evaluated. The generation of evolution was repeated until a desired termination criterion was reached.

5 ANN model development

Maier and Dandy [20] outlined the steps that should be followed in the development of ANN models, discussed the options available to modelers at each of these steps, and highlighted the issues that should be considered. In order to improve network performance, many factors like determination of adequate model inputs, data division and preprocessing, the choice of suitable network architecture, selection of network internal parameters, the stopping criteria and model testing need careful addressing. Data pre-processing is necessary before they are applied to the ANN to ensure all variables received equal attention during the training process. However, in this particular study, Matlab computing software [21] was used for the development of ANN and ANN-GA models.

Model development consists of two parts, training and testing. Seventy percent of the total data, i.e., data from the year 1993 to 1999, were used for training and thirty percent of the total data, i.e., data from the year 1999 to 2002, were used for testing. During training, in the beginning, the RMSE (root mean square error) generally decreased as number of nodes in the hidden layer increased and then RMSE started to increase when the model began to overfit. To avoid over-fitting, the early stopping criteria was used. In early stopping method, the best parameter set was selected and the process of training stops at a hidden node number when the RMSE of the training set started to increase. After completion of training and testing, results from different algorithm-based models were compared in terms of various PEC (performance evaluation criteria) and the best model was selected.

Various internal parameters used in ANN networks were chosen following trial–error method (trade-off between model accuracy and model complexity), which were as follows: number of epoch = 1,000, goal = 0.001, momentum coefficient = 0.8. In addition, optimum combinations of transfer functions in the hidden and output layer were determined and are presented in the Table 1.

6 ANN-GA hybrid model development

An individual run in GA is presented in the Fig. 4. Number of neurons in the hidden layer of the best ANN model was used in ANN-GA modeling for both months of the year. The values and types of optimal GA parameters were chosen by trial and error method as given in the Table 2. Then, the process of training was carried out using eight different generations, i.e., from 3 to 10 each with 50, 100, and 150 numbers of populations. After that, models were tested and evaluated using five different performance evaluation criteria. Then, the best model predicted by ANN in each month of a year was compared with respective

Table 1 Optimum combination of transfer functions in the hidden and output layer

Time-step	GDX algorithm	LM algorithm	BR algorithm
November	tansig-purelin	tansig-tansig	logsig-purelin
January	tansig-purelin	tansig-purelin	logsig-purelin

tansig Tangent sigmoid, *purelin* purely linear, *logsig* logistic sigmoid

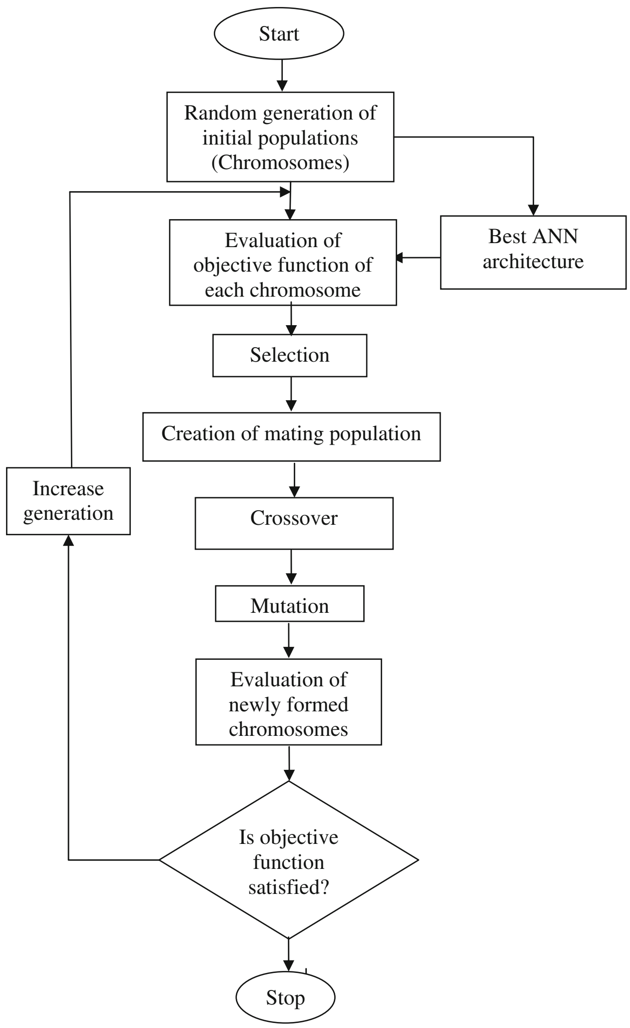

Fig. 4 An individual run in genetic algorithm-based hybrid modeling approach

optimal ANN-GA model to decide the best model for the months under consideration.

7 Performance evaluation of the models

A variety of PEC that could be used for evaluation and inter-comparison of different models were proposed by WMO [29] and Nash and Sutcliffe [22]. They were grouped as graphical and numerical performance indicators. For the present study, scatter plot criteria (of the simulated versus observed) was used as the graphical indicator where as numerical indicators suitable for the present study are the correlation coefficient (R), Nash–Sutcliffe coefficient (E), index of agreement (IOA), root mean square error (RMSE), mean absolute error (MAE). These are given by the following equations.

Table 2 GA parameters used in ANN-GA modeling

Options	Type or value
PopulationType	doubleVector
PopInitRange	[2 × 1 double]
EliteCount	2
CrossoverFraction	0.80
MigrationDirection	Forward
MigrationInterval	20
MigrationFraction	0.20
TimeLimit	Inf
FitnessLimit	−Inf
StallGenLimit	Inf
StallTimeLimit	Inf
PlotInterval	1
CreationFcn	@gacreationuniform
FitnessScalingFcn	@fitscalingrank
SelectionFcn	@selectionstochunif
CrossoverFcn	@crossoverscattered
MutationFcn	@mutationgaussian
Display	Iter
OutputFcns	@gaoutputgen
Vectorized	Off

$$R = \frac{\sum_{i=1}^{n} (O_i - \overline{O}_i)(P_i - \overline{P}_i)}{\sqrt{\sum_{i=1}^{n} (O_i - \overline{O}_i)^2 \sum_{i=1}^{n} (P_i - \overline{P}_i)^2}} \quad (22)$$

$$E = 1 - \frac{\sum_{i=1}^{n} (O_i - P_i)^2}{\sum_{i=1}^{n} (O_i - \overline{O}_i)^2} \quad (23)$$

$$\text{IOA} = 1 - \frac{\sum_{i=1}^{n} (O_i - P_i)^2}{\sum_{i=1}^{n} \left[\left| P_i - \overline{O}_i \right| + \left| O_i - \overline{O}_i \right| \right]^2} \quad (24)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (O_i - P_i)^2} \quad (25)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |O_i - P_i| \quad (26)$$

where n = total number of data sets; O_i = observed groundwater level for the ith data set; P_i = predicted groundwater level for the ith data set; $\overline{O}_i$ = mean of observed groundwater level for the ith data set; $\overline{P}_i$ = mean of predicted groundwater levels for the ith data set.

Correlation coefficient (R) is a measure of how well the variation in the output is explained by the targets. 'R' value equal to one implies a perfect fit between the outputs and the targets. Smith [27] suggested the following guide for values of $|R|$ between 0.0 and 1.0:

$|R| \geq 0.8$: Strong correlation exists between two sets of variables;

$0.2\langle|R|\langle 0.8$: Correlation exists between the two sets of variables; and
$|R| \leq 0.2$: Weak correlation exists between the two sets of variables.

Nash–Sutcliffe coefficient (E) represents the percentage of the initial uncertainty explained by the model. Its value varies between -1 and 1. The closer the value to 1, the better is the model performance. Index of agreement (IOA) is an adaptation of the Nash–Sutcliffe coefficient. The alteration to the denominator seeks to penalize differences in the mean predicted and mean observed values. Due to the squaring of the error terms in all three measures, they are considered overly sensitive to outliers in the data set. The RMSE is the most popular measure of error and has the advantage that large errors receive much greater attention than small errors. RMSE can give a quantitative indication of the model error in terms of a dimensioned quantity. It indicates the discrepancy between the observed and predicted values. A RMSE value close to zero indicates better performance of the model. The best fit between the observed and predicted values, which is unlikely to occur, would have RMSE value equal to 0. In case of mean absolute error (MAE), the absolute value of the error terms are considered rather than square of the terms. Thus, in contrast with RMSE, MAE eliminates the emphasis given to large errors. Both RMSE and MAE are desirable when the evaluated output data are smooth or continuous [28].

8 Results and discussion

8.1 Development and comparison of ANN and ANN-GA hybrid models in November

The training and testing results of different models in terms of their PEC in the month of November are presented in the Table 3. From training results of ANN-based models, it was observed that GDX, LM, BR models trained best with 2, 3, and 2 numbers of hidden nodes, respectively, but LM model trained better than other ANN models as reflected by its better values of PEC. From testing results of ANN models, it is observed that, for BR model, the values of R, E, IOA, RMSE, and MAE are 0.892, 0.794, 0.939, 0.586, and 0.367, respectively, which was found to be better than respective values of other ANN models. Thus, BR model was selected as the best ANN model in November, due to its better performance in both training and validation phase. In terms of modeling efficiency (Nash–Sutcliffe coefficient), one could classify that the effectiveness of training algorithms in ANN-based groundwater level prediction is in the form of BR > GDX > LM with efficiency values 79.4, 78.5, and 75.6%, respectively. In the Table 3, one could also note that the learning ability of LM model is higher than that of GDX and BR.

Table 3 Performance evaluation statistics from different models for November

Model	DPM	R	E	IOA	RMSE	MAE
During training						
GDX	$N = 1$	0.832	0.692	0.901	0.674	0.401
LM	$N = 3$	0.847	0.717	0.911	0.646	0.378
BR	$N = 1$	0.831	0.690	0.900	0.677	0.402
ANN-GA	$P = 50, G = 5$	0.835	0.706	0.905	0.666	0.398
During testing						
GDX	$N = 1$	0.887	0.785	0.936	0.599	0.383
LM	$N = 3$	0.872	0.756	0.932	0.638	0.396
BR	$N = 1$	0.892	0.794	0.939	0.586	0.367
ANN-GA	$P = 50, G = 5$	0.899	0.795	0.943	0.502	0.336

The results obtained from the ANN-GA hybrid model during training phases have shown that the model trained best with a population size of 50 and 5 generations of iterations. The results obtained during testing phase show the values of R, E, IOA, RMSE, and MAE as 0.899, 0.795, 0.943, 0.502, and 0.336, respectively. It was observed that ANN-GA hybrid model produced better result for almost all statistical indices considered in this study, in comparison with the best ANN model (i.e., ANN model with BR algorithm). Both BR model and the optimal ANN-GA model have shown R-value greater than 0.89 in the scatter plots (Fig. 5). The Fig. 5 showed a strong correlation between the observed and predicted groundwater levels (GLs) for both the models. But higher degree of correlations in GLs was observed for the ANN-GA model, in comparison with best ANN model (BR type).

In this study, groundwater levels were classified into three classes viz. low, medium, and high. The low class having the water level ranges of 0–1.5 m; while the other groundwater level ranges like 1.5–3.0 m and greater than 3.0 m were considered as medium and high, respectively. From the comparison between the observed and the predicted GLs for the month of November at 26 stations in 2000–2002 (in Fig. 6), it was observed that the best ANN model predicted low and medium GLs accurately. The best ANN model (BR model) failed to predict the higher values, whereas the ANN-GA hybrid model predicted both medium and high GLs accurately with some overestimation in low GLs. It was also observed that both the BR and ANN-GA hybrid model successfully predicted the peaks of the GLs during 3 years of testing in most of the stations. However, except some disparity in low GLs, the ANN-GA hybrid model consistently produced better results than the BR model throughout the modeling period. Thus, from both statistical and graphical comparison, ANN-GA-based

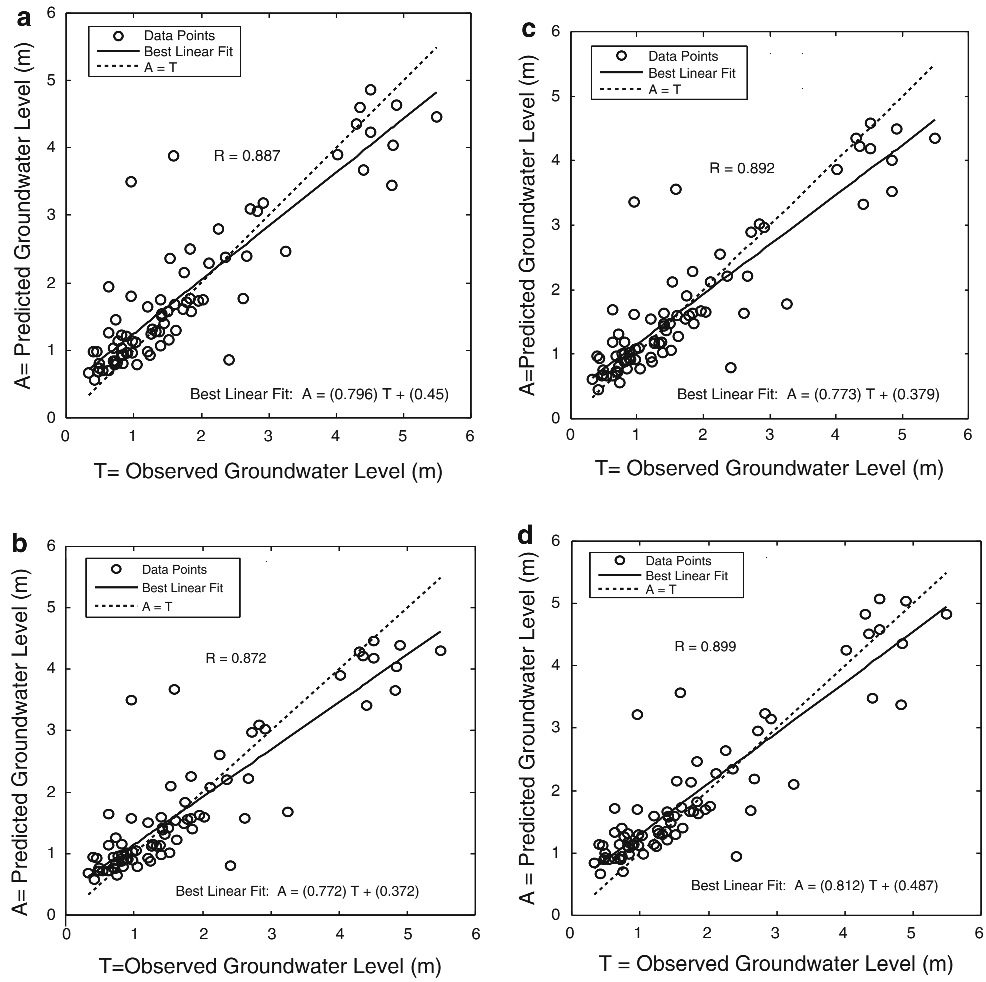

Fig. 5 Scatter plots of observed and predicted groundwater levels for November at 26 stations in 2000–2002. **a** GDX model, **b** LM model, **c** BR model, **d** ANN-GA model

optimum model with a population size of 50 and 5 generations of iterations was selected as the best model for the prediction of GLs during November.

8.2 Development and comparison of ANN and ANN-GA hybrid models in January

The results in terms of the various statistical measures during training and testing for the month of January are presented in Table 4. From training results of ANN-based models, it was observed that GDX, LM, and BR models trained best with 4, 4, and 3 number of hidden nodes, respectively. Likewise the analysis in the previous section, the LM model trained better than other ANN models as reflected by its better values of PEC. Though the learning ability of LM model is better, it failed to give a consistent performance during testing phase in comparison with BR algorithm–based ANN model. It can be noticed from the Table 4 that the testing results for BR model have *R*, *E*, IOA, RMSE, and MAE values as 0.862, 0.699, 0.919, 0.661, and 0.441, respectively, which was found to be better than respective values of other ANN models. As we observed in month November, once again the study identified BR model as the best ANN model for modeling groundwater levels in the month of January. In general, the performance of all models during January month was relatively less than that of November month. The modeling efficiency (Nash–Sutcliffe coefficient multiplied by 100) evaluated during test phase showed values like 67.7, 67.9, and 70.4% for ANN algorithms like GDX, LM, and BR, respectively.

ANN-GA hybrid model has produced better results when the population size of 50 and 4 generations of iterations were used. Among all the evaluated models, best values of the *R*, *E*, IOA, RMSE, and MAE were observed for ANN-GA model. These values obtained from ANN-GA model were 0.867, 0.704, 0.929, 0.630, and 0.440, respectively, during the testing process. It was also observed that ANN-GA hybrid model produced better PEC values in comparison with the best ANN model (BR). The performance of each model is displayed in graphical scatter plot form in the Fig. 7. Higher correlation (*R*-value more than 0.80) was observed in both ANN-GA and BR model with significantly higher value associated with optimal

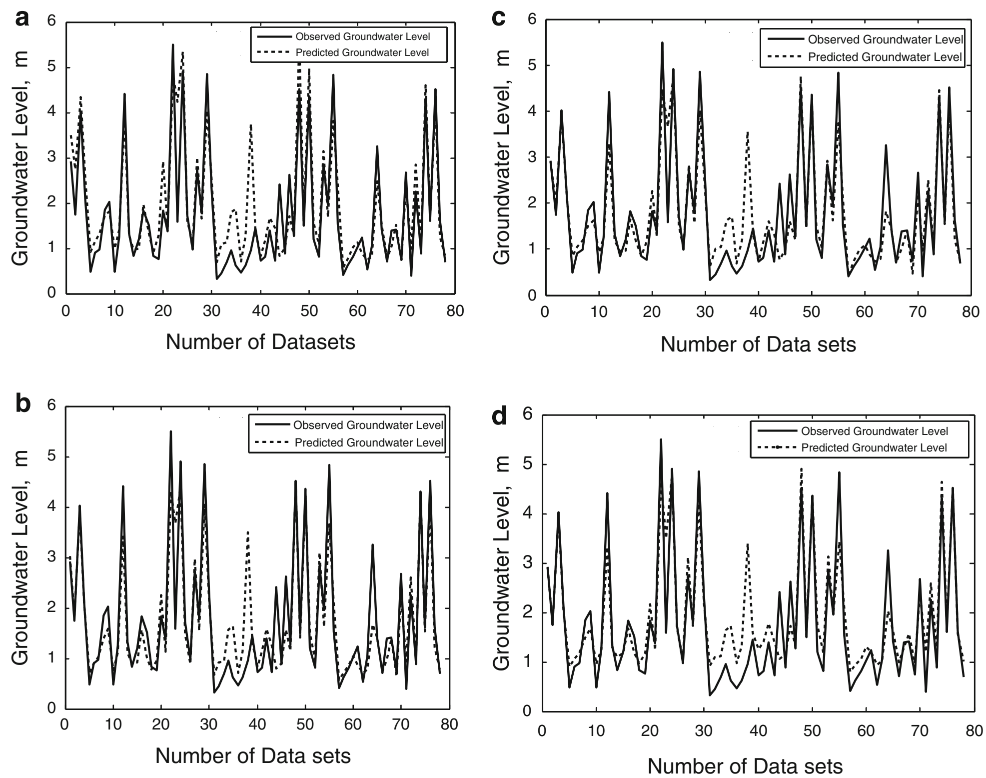

Fig. 6 Observed and predicted groundwater levels for November at 26 stations in 2000–2002. **a** GDX model, **b** LM model, **c** BR model, **d** ANN-GA model

Table 4 Performance evaluation criteria from different models for January

Model	DPM	*R*	*E*	IOA	RMSE	MAE
During training						
GDX	$N = 4$	0.905	0.819	0.948	0.471	0.315
LM	$N = 4$	0.922	0.849	0.958	0.430	0.288
BR	$N = 3$	0.913	0.834	0.953	0.451	0.306
ANN-GA	$P = 50, G = 4$	0.920	0.838	0.954	0.437	0.291
During testing						
GDX	$N = 4$	0.856	0.677	0.916	0.689	0.466
LM	$N = 4$	0.859	0.679	0.918	0.682	0.458
BR	$N = 3$	0.862	0.699	0.919	0.661	0.441
ANN-GA	$P = 50, G = 4$	0.867	0.704	0.929	0.630	0.440

ANN-GA structure. The performance of the evaluated models during 2000–2002 is given in graphical line plot in Fig. 8. These figures confirm that the best ANN model (BR model) was able to predict the flow values very accurately. It is worth mentioning that the best ANN model (BR model) was able to predict the GLs for all magnitude ranges (e.g., low, medium, and high) with reasonable accuracy. The ANN-GA hybrid model predicted medium and high GLs accurately but over predicted low GLs. It was also observed that both the BR and hybrid ANN-GA model successfully predicted the peaks of the GLs during 3 years of testing period in most of the stations. The performance of ANN-GA was more consistent with the exception of some mismatch in low GLs. Thus, from both statistical and graphical comparison, ANN-GA-based optimum model with a population size of 50 and 4 generations of iterations was selected as the best model for the prediction of GLs during January. The IOA values were higher for LM-based models during training phase (for both January and November month) because of its exhibited higher learning (training) abilities. But the ANN-GA model gave better index of agreement during validation phases of both the months during the study period. In general, the experiments explained in this case study showed the capability of ANN-GA model to make consistent predictions of groundwater levels than that of the BR model.

9 Conclusions

The present paper dealt with the groundwater level prediction of non-monsoon season (winter months like November and January) over Mahanadi river basin region in India using predictors like antecedent information of aggregated rainfall and antecedent information of groundwater level. The study has comprehensively compared and contrasted state-of-art functional algorithms generally used in ANN models viz. back-propagation (GDX), Levenberg–Marquardt (LM) and Bayesian regularization (BR). The study has adopted an ANN-GA hybrid model for reliable

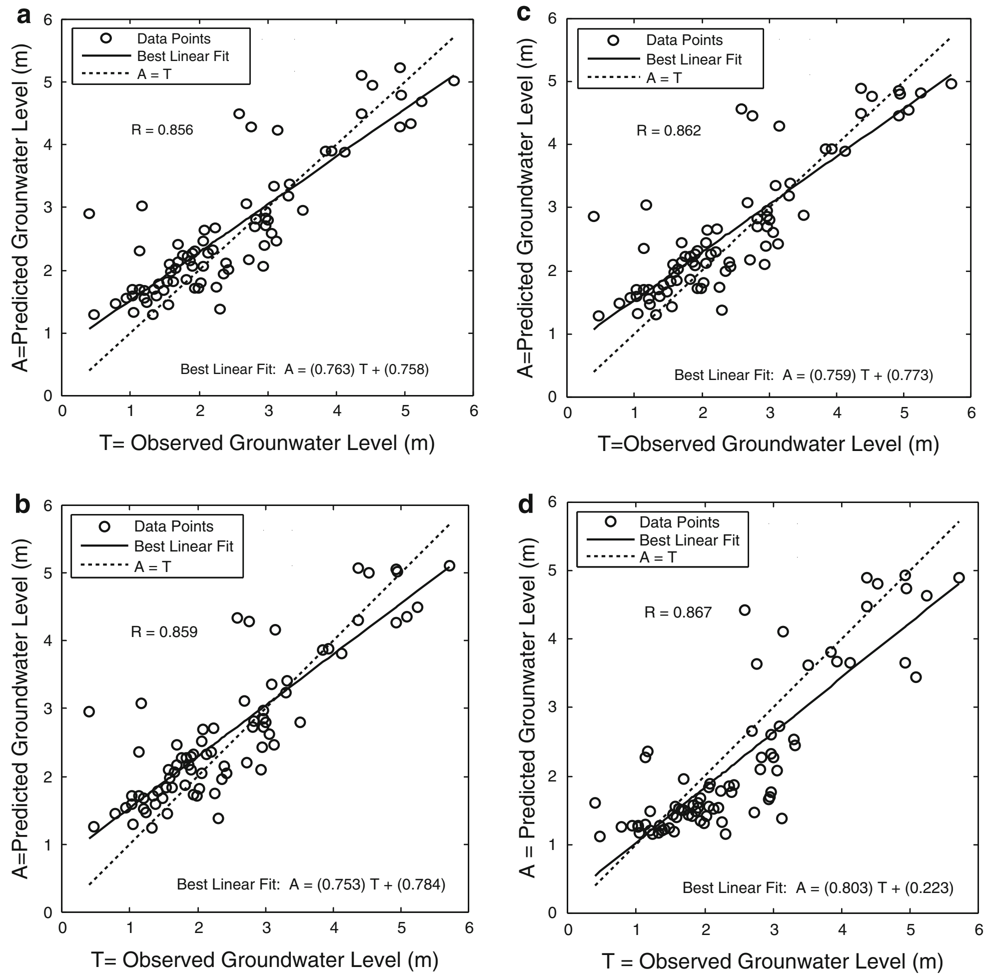

Fig. 7 Scatter plots of observed and predicted groundwater levels for January at 26 stations in 2000–2002. **a** GDX model, **b** LM model, **c** BR model, **d** ANN-GA model

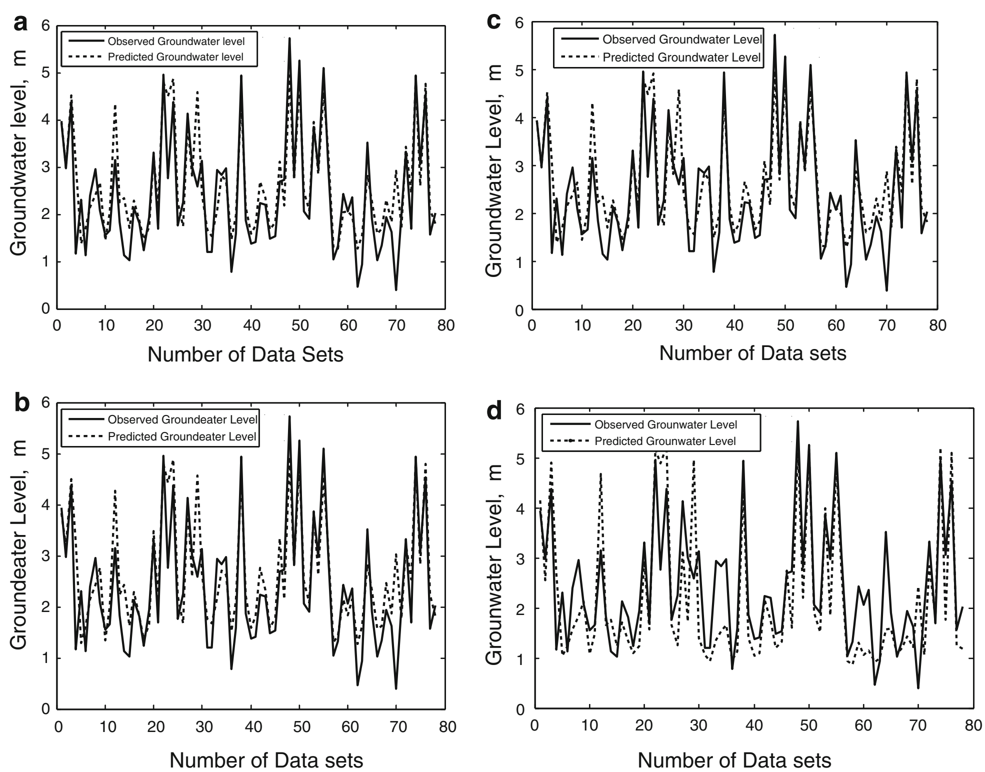

Fig. 8 Observed and predicted groundwater levels for January at 26 stations in 2000–2002. **a** GDX model, **b** LM model, **c** BR model, **d** ANN-GA model

prediction of winter groundwater level fluctuations and compared the results with that of above-mentioned algorithm-based ANN models. The study has acknowledged the better learning ability of LM algorithm–based ANN model, but the BR model beat the LM model with consistence performance on both training and validation phase. The proposed hybrid ANN-GA model has outperformed the best ANN model (BR-based ANN) with higher consistency and reliability in predicting higher values in groundwater fluctuation for both January and November months during the study period. However, the optimal ANN-GA model has shown some difficulty in capturing lower values of groundwater levels, which was successfully done by BR-based ANN model.

The significant advantage of ANN-GA models is that they can provide reliable predictions of medium and high groundwater levels even from a short record of data. This study results are very promising for improving water supply planning and water management in semiarid areas like India, where groundwater modeling data deficiency is a big issue. As long as the issues like lack of aquifer system information and lack of lengthy historical data persist, there is scope for these artificial intelligence models. Less cost and simplicity are another attraction of these models than conventional physically based models. Further improvements in prediction are possible if one could add more inputs like evapotranspiration, geometric and geographical factors, and groundwater withdrawal. However, the findings of the study reported in this paper usefulness of ANN-GA hybrid modeling approach in empirical groundwater level forecasting especially lower Mahanadi river basin of Orissa.

Acknowledgments This study was made possible as required groundwater level data were obtained from the Regional Director, Central Ground Water Board, Govt. of India, Bhubaneswar, and the required rainfall data were obtained from the Director, Board of Revenue, Cuttack.

References

1. ASCE Task Committee on Application of Neural Networks in Hydrology (2000) Artificial neural network in hydrology I: preliminary concepts. J Hydrol Energy ASCE 5(2):115–123
2. ASCE Task Committee on Application of Neural Networks in Hydrology (2000) Artificial neural network in hydrology II: hydrologic application. J Hydrol Energy ASCE 5(2):124–137
3. Bhattacharjya RK, Datta B (2005) Optimal management of coastal aquifer using linked simulation optimization approach. Water Resour Manag 19(3):295–320
4. Bhattacharjya RK, Datta B (2009) ANN-GA-based model for multiple objective management of coastal aquifers. J Water Resour Plann Manage 135(5):314–322
5. Coulibaly P, Anctil F, Bobee B (2000) Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. J Hydrol 230(3–4):244–257
6. Coulibaly P, Anctil F, Aravena R, Bobee B (2001) Artificial neural network modeling of water table depth fluctuations. Water Resour Res 37(4):885–896
7. Coulibaly P, Anctil F, Bobee B (2001) Multivariate reservoir inflow forecasting using temporal neural networks. J Hydrol Energy 9–10:367–376
8. Coulibaly P, Bobee B, Anctil F (2001) Improving extreme hydrologic events forecasting using a new criterion for artificial neural network selection. Hydrol Process 15:1533–1536
9. Daliakopoulosa IN, Coulibaly P, Tsanis IK (2005) Groundwater level forecasting using artificial neural networks. J Hydrol 309:229–240
10. Fausset L (1994) Fundamentals of neural networks. Prentice Hall, Englewood Cliffs
11. French MN, Krajewski WF, Cuykendall RR (1992) Rainfall forecasting in space and time using a neural network. J Hydrol 137:1–31
12. Goldberg DE (2000) Genetic algorithms in search, optimization and machine learning. Addision-Wesley, Reading
13. Gautam MR, Watanabe K, Saegus H (2000) Runoff analysis in humid forest catchment with artificial neural network. J Hydrol 235:117–136
14. Govindaraju RS, Rao AR (2000) Artificial neural networks in hydrology. Kluwer Academic Publishers, Amsterdam, The Netherlands
15. Haykin S (1994) Neural networks—a comprehensive foundation. Macmillan, New York
16. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
17. Kelkar U, Narula KP, Sharma VP, Chandna U (2008) Vulnerability and adaptation to climate variability and water stress in Uttarakhand State, India. Global Environ Change 18:564–574
18. Lallahem S, Mania J, Hani A, Najjar Y (2005) On the use of neural networks to evaluate groundwater levels in fractured media. J Hydrol 307:92–111
19. Maier HR, Dandy GC (1998) Understanding the behavior and optimizing the performance of back-propagation neural networks: an empirical study. Environ Model Softw 13:179–191
20. Maier HR, Dandy GC (2000) Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. Environ Model Softw 15:101–124
21. Mathworks, Inc. (2004) Neural network and genetic algorithm toolbox for Matlab, http://www.mathworks.com
22. Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I—a discussion of principles. J Hydrol 10:282–290
23. Parasuraman K, Elshorbagy A (2007) Cluster-based hydrologic prediction using genetic algorithm-trained neural networks. J Hydrol Eng 12(1):52–62
24. Rao SVN, Kumar S, Shekhar S, Chakraborty D (2006) Optimal pumping from skimming wells. J Hydrol Energy 11(5):464–471
25. Rojas R (1996) Neural networks: a systematic introduction. Springer, Berlin
26. Rumelhart DE, Hinton GE, William RJ (1986) Learning internal representation by error propagation. In: Rumelhart DE, McClelland JL (eds) Parallel distributed processing. MIT Press, Cambridge
27. Smith GN (1986) Probability and statistics in civil engineering: an introduction. Collins, London
28. Twomey JM, Smith AE (1997) Validation and verification. Artificial neural networks for civil engineers: fundamentals and applications. In: Kartam N, Flood I, Garrett JH (eds) ASCE, New York, pp 44–64
29. World Meteorological Organization (1975) Inter-comparison of conceptual models used in operational hydrological forecasting.

World Meteorological Organization. Technical report no 429, Geneva, Switzerland
30. Wu CL, Chau KW (2006) A flood forecasting neural network model with genetic algorithm. Int J Environ Pollut 28(3–4):261–273
31. Yang CC, Prasher SO, Lacroix R, Sreekanth S, Patni NK, Masse L (1997) Artificial neural network model for subsurface-drained farmlands. J Irrig Drain Eng ASCE 123(4):285–292
32. Yang CC, Prasher SO, Lacroix R (1996a) Applications of artificial neural networks in subsurface drainage system design. In Proceedings of ASAE, Computers and Electronics in Agriculture, pp 932–940
33. Yang CC, Prasher SO, Lacroix R (1996) Applications of artificial neural networks to land drainage engineering. Trans ASAE 39(2):525–533
34. Yang CC, Prasher SO, Lacroix R (1996) Applications of artificial neural networks to simulate water-table depths under subirrigation. Can Water Resour J 21(1):27–44