Journal of Hydrologic Engineering, ASCE, Vol. 10, No. 6, 2005, pp. 485-491

Comparison of several flood forecasting models in Yangtze River

K.W. Chau[1], C.L. Wu[2] and Y.S. Li[1]

[1]*Department of Civil and structural Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, People's Republic of China*

[2]*Changjiang Water Resources Commission, 430010, Wuhan, HuBei, People's Republic of China*

Abstract

In a flood-prone region, quick and accurate flood forecasting is imperative. It can extend the lead time for issuing disaster warnings and allow sufficient time for habitants in hazardous areas to take appropriate action, such as evacuation. In this paper, two hybrid models based on recent artificial intelligence technology, namely, genetic algorithm-based artificial neural network (ANN-GA) and adaptive-network-based fuzzy inference system (ANFIS), are employed for flood forecasting in a channel reach of the Yangtze River in China. An empirical linear regression model is used as the benchmark for comparison of their performances. Water levels at downstream station, Han-Kou, are forecasted by using known water levels at the upstream station, Luo-Shan. When cautious treatment is made to avoid overfitting, both hybrid algorithms produce better accuracy performance than the linear regression model. The ANFIS model is found to be the optimal, but entails a large number of parameters. The performance of the ANN-GA model is also good, yet requires longer computation time and additional modeling parameters.

Keywords: Flood forecasting model; Hybrid algorithms; Artificial neural networks; Genetic algorithms; Fuzzy inference system

1. Introduction

A quick and accurate flood forecasting is required, in particular in a flood-prone region, for the issue of disaster warnings in order to allow ample time for the evacuation of populations endangered by imminent rising water levels. Models for flood propagation in a channel reach can broadly be classified into two main categories: conceptually based models; and, empirical models based on system analysis or "black-box" approach. In conceptually based models, the flood propagation process is usually described by the de Saint Venant equations comprising partial differential equations of continuity and momentum. These equations are not amenable to analytical solutions. During the past few decades, many conceptually based numerical schemes have been proposed to solve the problem (Chau and Lee 1991a & 1991b). Whilst conceptually based models have advantages in describing the mechanisms of the hydrological process, they require large amount of data (for example, characteristics of terrain and river networks, rainfall, and runoff) for calibration. In many occasions, these data may be unavailable, or expensive and time consuming to collect. Sophisticated physical models may not be ideal for real-time forecasting due to the tremendous data requirement and the associated long computation time for model calibration. On the other hand, empirical models are based on an evidence of relationships manifested in historical records of input and output records without analyzing the internal structure of the physical process.

In many practical situations, the main concern is about making accurate and timely predictions at specific locations. Then a simple "black box" model is preferred in identifying a direct mapping between inputs and outputs. In recent years, many nonlinear approaches, such as, artificial neural network (ANN), genetic algorithm (GA), and fuzzy logic, have been used in solving flood forecasting problems.

Smith and Eli (1995) applied a back-propagation ANN model to predict discharge and time to peak over a hypothetical watershed. Tokar and Johnson (1999) compared ANN models with regression and simple conceptual models. Liong *et al.* (2000) employed an ANN approach for river stage forecasting in Bangladesh. Chau and Cheng (2002) performed a real-time prediction of water stage with ANN approach using an improved back propagation algorithm. Chau (2004a & b) employed particle swarm optimization in river stage forecasting and rainfall-runoff correlation. The ASCE Task Committee (2000) summarized the state-of-the-art applications of ANN in hydrology and posed some future directions.

The literature describing the application of GAs to hydrological problems is not abundant. Olivera and Loucks (1997) employed a GA to formulate operating rules for multireservoir systems. Wardlaw and Sharif (1999) evaluate a GA for optimal reservoir system operation. Cheng *et al.* (2002) applied a GA to calibrate conceptual rainfall-runoff models. Chau (2002) calibrated flow and water quality modeling using a GA.

Fuzzy logic has been employed in a variety of hydrological applications. Russell and Campbell (1996) developed some reservoir operating rules with fuzzy programming and made comparison with deterministic dynamic programming. Fortane *et al.* (1997) planned reservoir operations through fuzzy set theory. Yu *et al.* (2000) forecasted rainfalls with combined gray and fuzzy methods. Cheng and Chau (2001) applied a fuzzy iteration methodology for reservoir flood control operations. Dubrovin *et al.* (2002) used total fuzzy similarity for real-time reservoir operations. Tilmant *et al.* (2002) compared reservoir operating policies from fuzzy and nonfuzzy explicit stochastic dynamic programming. Ponnambalam *et al.* (2002) employed a fuzzy system to minimize variance of operation benefits for reservoir systems.

The objectives of this study are to use hybrid algorithms for flood forecasting in a channel reach of the Yangtze River. This paper is organized as follows: algorithms of the genetic algorithm-based artificial neural network (ANN-GA) and adaptive-network-based fuzzy inference system (ANFIS) models are introduced; data mining and division for flood forecasting analysis in Yangtze River are presented; prediction results based on these hybrid algorithms are compared with those of the linear regression (LR) model; and finally, conclusions are drawn.

2. Genetic Algorithm-Based Artificial Neural Network (ANN-GA)

An ANN is a form of artificial intelligence mimicking the functioning of the human brain and nervous system. It acquires knowledge through a learning process that involves finding an optimal set of weights for the connections and threshold values for the neurons. Its ability to “train” and “learn” the outputs from a given input renders it possible to simulate large-scale arbitrarily complex non-linear problems (Rumelhart *et al.* 1994). The most widely used ANN is the feed-forward network with a back-propagation algorithm, termed the back propagation network (BPN). In the forward pass, the output response at each neuron is computed from the weighed sum of its inputs and bias from neurons connected to it, using a predetermined activation function. A sigmoid type activation function is often used:

$$f(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

In the reverse pass, the weights in the network are modified by using the error value between the output of the network and the target outputs. The derivatives of the objective function with respect to the weights in the entire network are used to distribute the error to neurons in each layer. A gradient descent method is often used to move the weights in the direction in which the error declines most quickly. Training is stopped when the error is smaller that a preset value. Although an ANN is a flexible and powerful mapping tool, initialization of weights and biases has a significant effect on network performance. Inappropriate assigned weights and biases can lead to local convergence. Moreover, the traditional BPN with gradient descent learning algorithm is characterized by a surprisingly slow convergence and a long computational time.

On the other hand, a GA is an application of biological principles into computational algorithm and has been employed to attain the optimum solutions (Goldberg and Kuo 1987). A GA holds the ability of searching, yet may not necessarily lead to the best possible solution. A hybrid integration of these two algorithms may take advantages of the characteristics of both schemes. It can increase solution stability and improve performance of an ANN model, though at the expense of computational time. Hence, a genetic algorithm-based artificial neural network (ANN-GA) model is developed here. A GA is firstly employed to optimize initial parameters of ANN, prior to training by the conventional ANN.

The objective function of the GA sub-model used for initializing weights and biases is denoted by:

$$\min J(W,\theta) = \sum_{i=1}^{p} \left| Y_i - f(X_i, W, \theta) \right| \tag{2}$$

where W is the weight, θ is the bias or threshold value, i is the data sequence, p is the total number of training data pairs, X_i is the ith input data, Y_i is the ith measured data, and $f(X_i, W, \theta)$ represents simulated output. The goal of the GA sub-model is to ascertain optimal parameters so that accumulative errors between the measured data and simulated data are minimal. The overall flow chart of the ANN-GA model is shown in Figure 1, in which p_c is the crossover probability, p_m is the mutation probability, $G_{\max}$ is the maximum number of generation, and $N_{\max}$ is the population size.

3. Adaptive-Network-Based Fuzzy Inference System (ANFIS)

Fuzzy logic and fuzzy set theory have been widely used to simulate the ambiguity and uncertainty in decision making. The key ideas for fuzzy logic are to allow for quantities to be partial truth rather than having to be either crisply “true” or “false” (Zadeh and Kacprzyk 1992). The degree of “belongingness” to a set or category can be described numerically by a membership function, with range from 0 to 1. The function may be triangular-shaped, trapezoidal-shaped, bell-shaped, etc. In general, fuzzy logic programming can be used in two ways: to model the behavior of a human expert; and, to map a set of outputs to a set of inputs in a fuzzy inference method. In order to model the thinking of a human expert, input variables are specified by category, such as “low”, “high”; and fuzzy rules are developed on the basis of the expert’s knowledge and experience. In the present study, however no expertise is available and the number of membership functions assigned arbitrarily.

In this sense, the fuzzy inference is similar to an ANN, both with the goals to identify the transformation of a set of inputs to the corresponding set of outputs through training. An ANN tends to operate more in a “black box” manner, whereas a fuzzy logic system is more transparent due to the incorporation of the expert’s knowledge and experience into the inference process. Jang (1993) classified the fuzzy inference systems into three types in accordance with the nature of fuzzy reasoning and fuzzy if-then rules. In the present study, Takagi and Sugeno’s fuzzy if-then rules are employed (Takagi and Sugeno 1985). The output of each rule is a linear combination of input variable plus a constant term whilst the final output is the weighted averaged of each rule’s output. Figure 2 shows a sample fuzzy reasoning with two input variables.

The fuzzy rule base is set up by combining all categories of variables. The following illustrates a sample case comprising three input variables and a single output variable. Each input variable (x , y , and z) is divided into three categories. For simplicity, equally spaced triangular membership functions are assigned. The categories are “low,” “medium,” and “high.” The number of rules in a fuzzy rule base is c^n , where c is the number of categories per variable and n the number of variables. The optimal number of categories can be selected through performance comparison. The rule base takes the form of an output $o_{i,j,k}$ for each combination of category i , of input variable x , category j , of input y , and category k , of input variable z . In this case, there are 27 rules in total. Part of the rule sets are illustrated as follows:

If x is low, y is low, and z is low then the output $o_{1,1,1} = a_1x + b_1y + c_1z + d_1$;

If x is low, y is low, and z is medium then the output $o_{1,1,2} = a_2x + b_2y + c_2z + d_2$;

If x is low, is low, and z is high then the output $o_{1,1,3} = a_3x + b_3y + c_3z + d_3$;

…

If x is high, y is high, and z is high then the output $o_{3,3,3} = a_{27}x + b_{27}y + c_{27}z + d_{27}$;

where $a_{(\cdots)}, b_{(\cdots)}, c_{(\cdots)}$, and $d_{(\cdots)}$ are parameters of fuzzy output functions. In this study, these parameters are determined through training.

For each rule triggered, memberships of x , y , and z are computed. The result of a specific T-norm operation will then provide the weight $w_{i,j,k}$ to be assigned to the corresponding output $o_{i,j,k}$. Multiplication operation is used in this study. Finally, the outputs from all triggered rules are combined to give a single weighted average output

$$o = \frac{\sum w_{i,j,k} \cdot o_{i,j,k}}{\sum w_{i,j,k}} \tag{3}$$

In order to develop a FIS model for forecasting, some parameters, including t_1 , t_2 and t_3 of each triangular membership function and a_i , b_i , c_i and d_i of the consequence part of each rule, have to be determined by learning. An ANFIS is the system when ANN is employed to train these FIS parameters. A fuzzy system based on hybrid algorithms can improve the intelligence of systems working in uncertain, imprecise, and noisy environments and can achieve faster convergence. It possesses the features of both the neural networks (learning abilities, optimization abilities, and connectionist structures) and the fuzzy control systems (human like “if-then” rule thinking and ease of incorporating expert knowledge).

Figure 3 shows the ANFIS configuration employed in this study. The parameters defining the shape of the membership functions are identified by the back-propagation learning algorithm, whereas the consequent parameters for each rule are identified by the least-squares method. In Figure 3, $w_{t-2,1}$ $w_{t-1,1}$ and $w_{t,1}$ are degrees of membership functions of X_{t-2}, X_{t-1}, and X_t , respectively; $w_{1,1,1}$ is the product by $w_{t-2,1}$ $w_{t-1,1}$ and $w_{t,1}$, and subscripts i, j, and k of $w_{i,j,k}$ and $o_{i,j,k}$ are integers in the range [1,3]; and, l is from 1 to 27.

4. Analysis, Results and Discussions

4.1 Study area

The channel reach studied is in the middle stream of the Yangtze River, which is the largest river in China. It passes through Wuhan city, which is the capital of Hubei province (see Figure 4). The flow of the Yangtze River is quite unsteady, and exhibits a seasonal behavior. The flow is low during the winter months and the peak flow occurs during August and September. A hydrological year is often classified into a flooding period and a non-flooding period, which are from June to October and from November to next May, respectively. The water level at Luo-Shan station can be as low as 17.3m during the non-flooding period and as high as 31.0m during the flooding period. The averages of water levels are 20.8m and 27.1m during the non-flooding and flooding periods, respectively. The purpose of this study is to predict water levels of the downstream station, Han-Kou, by known water levels of the upstream station, Luo-Shan. The lateral inflow is neglected since it is very small in comparison with the discharge of the main stream.

4.2 Data mining

The travel time of flood between Luo-Shan and Han-Kou is determined to be about 24hrs using Muskingum method. In other words, the flood at Han-Kou has a phase lag of approximately one day with that at Luo-Shan. A mapping of D points in a time series of spacing Δ apart, $X(t-(D-1)\Delta),\cdots,X(t-\Delta),X(t))$, can be generated to predict a future value $Y(t+p)$. In the present study, it is initially set that $p=1$ day and $\Delta=1$ day whilst the optimal value of D is found on the basis of the correlation. An integer value from 1 to 4 is assigned to D for testing. The data used for modeling are daily averages for water levels of Luo-Shan and Han-Kou stations. Analysis has been done on the appropriateness of working with average water level values with the possibility of averaging out the peaks and low levels. It is found from the data that the water levels do not vary so rapidly so that water levels at shorter intervals (at hourly or 3 hourly) are not required.

4.3 Division of data

If the output fits the training data too closely, it simulates the noise in addition to the underlying function. In such case, a problem occurs and the model overfits the training data but does not fit well to new data. Smith (1993) proposed three methods to solve this problem: limit the number of hidden nodes; discourage the network from using large weights; and, limit the number of training epochs. Shahin et al. (2002) suggested dividing the data into three subsets. In this study, the data are randomly divided into three sets: training, testing, and validation. Whilst 75% of the data are used for training, 25% are used for validation. The training data are further divided into 2/3 for the training set and 1/3 for the testing set.

In the present study we extract 1456 input-output data pairs of the following format from the data record:

$$[X(t-2), X(t-1), X(t);\; Y(t+1)] \tag{4}$$

which represents the relationship between water levels at Luo-Shan during the past three days and the water level at Han-Kou for the next day. It is ensured that data used for training, testing and validation represent the same population so that there is no need to extrapolate beyond the range of their training data. Table 1 shows the statistical parameters, including the mean, standard deviation, minimum, maximum, and range, for the training, testing, and validation sets, respectively.

4.4 Evaluation criteria for model performance

The performance of the predictions resulting from training, testing, and validation is evaluated by the following measures for goodness-of-fit: RMSE (root mean square error) and CC (coefficient of correlation):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{p}[(X_m)_i - (X_s)_i]^2}{p}} \tag{5}$$

$$CC = \frac{\sum_{i=1}^{p}[(X_m)_i - (\overline{X}_m)_i][(X_s)_i - (\overline{X}_s)_i]}{\sqrt{\sum_{i=1}^{p}[(X_m)_i - (\overline{X}_m)_i]^2[(X_s)_i - (\overline{X}_s)_i]^2}} \tag{6}$$

where subscripts m and s = the measured and simulated water levels, respectively; p = total number of data pairs considered; $\overline{X}_m$ and $\overline{X}_s$ = mean value of the measured and simulated data, respectively. RMSE furnishes a quantitative indication of the model error in units of the variable, with the characteristic that larger errors receive greater attention than smaller ones. The qualitative evaluation of the model performance is made in terms of the coefficient of correlation between the measured and simulated data.

4.5 Results of linear regression (LR) model

Since the basic characteristics of the Yangtze River remain unaltered in years, there exists certain correlation between the upstream (Luo-Shan) and the downstream (Han-Kou) water levels. A LR model is the simplest and well-developed representation of a causal, time-invariant, relationship between an input function of time and the corresponding output function. Hence, a LR model is developed as the benchmark for comparison in flood forecasting. In order to avoid overfitting, four regression models are initially developed with D value ranging from 1 to 4, with results as shown in Table 2. Table 3 shows the correlation amongst the input variables, which might result in multi-collinearity problem (i.e., imprecise regression parameter estimates due to highly correlated independent variables). The existence of surplus variables might lead to overfitting of the training set, but at the same time lowering the performance of the validation set. This is demonstrated for the case when D = 4 in Table 2. The adopted prediction model, based on the linear regression with D = 3, is expressed as follows:

$$X_{t+1} = 1.441X_t - 0.823X_{t-1} + 0.393X_{t-2} - 5.073 \qquad (7)$$

The option of D = 1 is equally good and which also support the principle of parsimony. In order to verify this, a comparison of the performance of the models was made by taking D = 1. No significant lowering of performance has been observed.

4.6 Results of ANN-GA model

Under the same basis of comparison, three inputs and one output are applied to the ANN-GA model. The input and output data are normalized to a range from 0 to 1, corresponding to the minimum and the maximum water levels, respectively. In this study, a three-layer network is used. In order to determine the optimal network geometry that holds for good generalization, ANN-GA models with different number of nodes ranging from 1 to 7 in the hidden layer are trained based on a trial and error procedure. The performances for training set and testing set at different training epochs are recorded. Training is stopped when the error learning curve of the testing set starts to increase and that of the training set is still decreasing. The performances for the testing set against different number of hidden nodes are shown in Figure 5. The same analysis has been attempted by dividing the data set afresh again and analysis repeated. The reproducibility of the same result has been ensured by repeating it 5 to 6 times. The optimal ANN-GA architecture adopted is 3-3-1. Figure 6 shows the prediction results and absolute errors for the validation data set with the ANN-GA model.

4.7 Results of ANFIS model

An ANFIS model is adopted to forecast the downstream water levels, with the same input-output data pairs as those of the LR and ANN-GA models. In general, more number of categories will furnish higher accuracy, but with the disadvantage of larger rule bases as well as more computation time. For example, with five fuzzy categories for each of three variables, this would involve a set of 125 (i.e. 5^3) rules, which are too much to allow patterns to be easily discerned. The parameters from premises and consequences are increased significantly and the computational time is rather long whilst the performance might only be improved slightly. Similar to that as described in the ANN-GA model, a testing set is adopted in order to overcome overfitting. Experiments are undertaken in order to select the appropriate number of variable categories, with results as shown in Table 4. The number of parameters increases remarkably by 495 from 2 to 5 categories whilst the training time increases significantly from 2.2s to 1219s. Although more subspaces for the ANFIS model will generally result in better performance, cautious treatment should also be made to avoid overfitting. When both the computational time and RMSE_vali are considered, an optimal number of categories of 3 (i.e., low, medium, and high) is adopted, with their fuzzy membership functions shown in Figure 7.

4.8 Discussions

Figure 8 depicts a qualitative comparison of the performance of the LR, ANN-GA, and ANFIS models in forecasting 1-day lead time water levels at Han-Kou by the upstream water levels at Luo-Shan station during the past three days for the validation set using an absolute error indicator. It can be seen that, amongst them, the fluctuation of absolute error for the LR model is the largest. The fluctuation is smallest for the ANFIS model with more data concentrated near the zero axis. Since the comparison between ANN-GA or ANFIS (both with non-linear nature) and LR model is not fair, a non-linear statistical fitting (NLSF) in dependent variable is also added. Table 5 represents a quantitative comparison of their performance, using indicators including RMSE_tra RMSE_vali, training time, and number of parameters. Whilst the results from all models may be satisfactory in terms of RMSE_vali and RMSE_tra, the ANFIS model is better in accuracy comparatively. The performance between LR and NLSF does not differ too much. Moreover, the ANFIS model consumes less training time than ANN-GA model and thus appears better suited to flood forecasting environment. Some possible explanations on the differences in their performances are suggested as follows.

Whilst the LR model can only fit a linear function to input-output data pairs, the ANN-GA model is capable of contorting itself into a complex form to accommodate the temporal changes of the input-output data pairs. The coupling of ANN and GA can improve the performance since it takes advantage of the local optimization of ANN and the global optimization of GA. It is reasonable that an ANN-GA model with 16 parameters is more flexible than LR model with 4 parameters. This is analogous to an anticipated better performance of a power or polynomial function than that of a simple linear function.

Similar to the ANN-GA model, the ANFIS model is also able to handle non-linear and complicated problems. In this ANFIS model, the mapping space of input variables to output variable is classified into 27 subspaces, and each subspace is described by a linear model (i.e., consequent part of a rule). It is natural that a phase-space linear function from the ANFIS model outperforms a LR model. Furthermore, a larger number of parameters used in the ANFIS model than its counterparts of LR is another factor contributing to the more rigorous performance.

The exhibition of a better forecast for the ANFIS model than the ANN-GA model can be explained by their difference in nature. It is demonstrated that the local approximation approach of the ANFIS model is capable of better mapping the connectivity of input-output data pairs than the global approximation approach of the ANN-GA model. Moreover, the ANN-GA model requires more training time than the ANFIS model since GA is a time consuming searching tool. However, with the fast development of computer technology, the computational time will not be a constraint.

5. Conclusions

In the present study, when cautious treatment is addressed to avoid overfitting problems, both ANN-GA and ANFIS models produce accurate flood predictions of the channel reach between Luo-Shan and Han-Kou stations in the Yangtze River. These “black-box” models avoid the complication of traditional physical model, in particular the necessity to glean site-specific parameters. Amongst them, the ANFIS model is the best in terms of the simulation performance and appears better suited to flood forecasting environment. However, it requires a larger amount of parameters in comparison with the benchmark LR model. The ANN-GA model adequately combines the advantage of ANN (i.e., fast convergence and local optimization) with the advantage of GA (i.e., global searching ability). However, it costs the longest computation time. Some additional parameters, such as crossover probability and mutation probability, are required for the GA component. The ANN-GA and ANFIS models could be considered as complement to conventional models and hence are worthy tools.

Acknowledgement

This research was supported by the Internal Competitive Research Grant of Hong Kong Polytechnic University (A-PE26).

References

Chau, K.W. (2002). “Calibration of flow and water quality modeling using genetic algorithm.” *Lecture Notes in Artificial Intelligence*, Vol. 2557, 720-720.

Chau, K.W. (2004a). “River stage forecasting with particle swarm optimization.” *Lecture Notes in Computer Science*, Vol. 3029, 1166-1173.

Chau, K.W. (2004b). “Rainfall-runoff correlation with particle swarm optimization algorithm.” *Lecture Notes in Computer Science*, Vol. 3174, 970-975.

Chau, K.W., and Cheng, C.T. (2002). “Real-time prediction of water stage with artificial neural network approach.” *Lecture Notes in Artificial Intelligence*, Vol. 2557, 715-715.

Chau, K.W., and Lee, J.H.W. (1991a). “Mathematical modelling of Shing Mun River network.” *Advances in Water Resources*, Vol. 14, No. 3, 106-112.

Chau, K.W., and Lee, J.H.W. (1991b) “A microcomputer model for flood prediction with applications.” *Microcomputers in Civil Engineering*, Vol. 6, No. 2, 109-121.

Cheng, C.T., and Chau, K.W. (2001). “Fuzzy iteration methodology for reservoir flood control operation.” *Journal of the American Water Resources Association*, Vol. 37, No. 5, 1381-1388.

Cheng, C.T., Ou, C.P., and Chau, K.W. (2002). “Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration.” *Journal of Hydrology*, Vol. 268, No. 3, 72-86.

Dubrovin, T., Jolma, A., and Turunen, E. (2002). “Fuzzy model for real-time reservoir operation.” *Journal of Water Resources Planning and Management, ASCE*, Vol. 128, No. 1, 66-73.

Fortane, D.G., Gates, T.K., and Moncada, M. (1997). “Planning reservoir operations with imprecise objectives.” *Journal of Water Resources Planning and Management, ASCE*, Vol. 123, No. 3, 154-162.

Goldberg, D.E., and Kuo, C.H. (1987). “Genetic algorithms in pipeline optimization.” *Journal of Computing in Civil Engineering, ASCE*, Vol. 1, No. 2, 128-141.

Govindaraju, R.S. (2000). “Artificial neural networks in hydrology. II: hydrologic application.” By the ASCE Task committee on Application of ANNs in Hydrology. *Journal of Hydrologic Engineering, ASCE*, Vol. 5, No. 2, 124-137.

Jang, J.S.R., (1993). “ANFIS: adaptive-network-based fuzzy inference system.” I*EEE Trans. Systems Man Cybernet*, Vol. 23, No. 3, 665-685.

Liong, S.Y., Lim, W.H., and Paudyal, G.N. (2000). “River stage forecasting in Bangladesh: neural network approach.” *Journal of Computing in Civil Engineering, ASC*E, Vol. 14, No. 1, 1-8.

Olivera, R., and Loucks, D.P. (1997). “Operating rules for multireservoir systems.” *Water Resources Research*, Vol. 33, No. 4, 839-852.

Ponnambalam, K., Karray, F., and Mousavi, S.J. (2002). “Minimizing variance of reservoir systems operations benefits using soft computing tools.” *Fuzzy Sets and Systems*, Vol. 139, No. 2, 451-461.

Rumelhart, D.E., Widrow, B., and Lehr, M.A. (1994). “The basic ideas in neural networks.” *Communications of the ACM*, Vol. 37, No. 3, 87-92.

Russell, S.O., and Campbell, P.F. (1996). “Reservoir operating rules with fuzzy programming.” *Journal of Water Resources Planning and Management, ASCE*, Vol. 122, No. 3, 165-170.

Shahin, M.A., Maier, H.R., and Jaksa, M.B. (2002). “Predicting settlement of shallow foundations using neural networks.” *Journal of Geotechnical and Geoenviromental Engineering, ASCE*, Vol. 128, No. 9, 785-793.

Smith, M. (1993). *Neural Networks for Statistical Modeling*, Van Nostrand Reinhold, New York.

Smith, J., and Eli, R.N. (1995). “Neural-network models of rainfall-runoff process.” *Journal of Water Resources Planning and Management, ASCE*, Vol. 121, No. 6, 499-508.

Takagi, T., and Sugeno, M. (1985). “Fuzzy identification of systems and its applications to modeling and control.” *IEEE Transactions on Systems Man and Cybernetics*, Vol. 15, No. 1, 116-132.

Tilmant, A., Vanclosster, M., Duckstein, L., and Persoons, E. (2002). “Comparison of fuzzy and nonfuzzy optimal reservoir operating policies.” *Journal of Water Resources Planning and Management, ASCE*, Vol. 128, No. 6, 390-398.

Tokar, A.S. and Johnson, P.A. (1999). “Rainfall-runoff modeling using artificial neural networks.” *Journal of Hydrologic Engineering, ASCE*, Vol. 4, No. 3, 232-239.

Wardlaw, R., and Sharif, M. (1999). “Evaluation of genetic algorithms for optimal reservoir system operation.” *Journal of Water Resources Planning and Management, ASCE*, Vol. 125, No. 1, 25-33.

Yu, P.S., Chen, C.J., and Chen, S.J. (2000). “Application of gray and fuzzy methods for rainfall forecasting.” *Journal of Hydrologic Engineering, ASCE*, Vol. 5, No. 4, 339-345.

Zedeh, L., and Kacprzyk, J., eds. (1992). *Fuzzy Logic for the Management of Uncertainly*. John Wiley, New York, N.Y.

Table 1. Statistical parameters for training, testing, and validation sets

Model variables and data sets	Statistical parameters				
	Mean	Standard Deviation	Min.	Max.	Range
X_{t-2}(m)					
Training set	23.42	3.73	17.37	31.04	13.67
Testing set	23.38	3.75	17.35	30.93	13.58
Validation set	23.41	3.71	17.39	30.96	13.57
X_{t-1}(m)					
Training set	23.40	3.74	17.35	30.96	13.61
Testing set	23.43	3.73	17.37	31.04	13.67
Validation set	23.41	3.68	17.37	30.80	13.43
X_t(m)					
Training set	23.45	3.73	17.37	31.04	13.67
Testing set	23.42	3.64	17.39	30.96	13.57
Validation set	23.39	3.77	17.35	30.93	13.58
Y_{t+1}(m)					
Training set	18.61	3.73	12.21	25.69	13.48
Testing set	18.59	3.76	12.20	25.71	13.51
Validation set	18.62	3.75	12.26	25.70	13.44

Table 2. Performance comparison for different values of D in LR model

	Training set		Validation set	
D	RMSE	CC	RMSE	CC
4	0.235	0.9880	0.238	0.9960
3	0.238	0.9880	0.237	0.9960
2	0.241	0.9880	0.243	0.9958
1	0.242	0.9880	0.244	0.9958

Table 3. Correlation amongst input variables

CC	X_{t-3}	X_{t-2}	X_{t-1}	X_t
X_{t-3}	1.000	0.973	0.985	0.989
X_{t-2}		1.000	0.977	0.965
X_{t-1}			1.000	0.987
X_t				1.000

Table 4 Performance comparison for different categories in ANFIS model

Number of categories	RMSE_tra(m)	RMSE_vali(m)	Training_time(s)	Number of parameters*
2	0.211	0.215	2.2	50
3	0.204	0.214	49.4	135
4	0.193	0.267	1074	292
5	0.193	0.193	1219	545

*including all t_1, t_2, t_3 and a_i, b_i, c_i, d_i

Table 5 Performance comparison for different models

Model	RMSE_tra(m)	RMSE_vali(m)	Training time(s)	Number of parameters
LR	0.238	0.237		4
NLSF	0.241	0.236		4
ANN-GA	0.213	0.226	135	16
ANFIS	0.204	0.214	49	135

Figure Captions

Figure 1. Flow chart for the ANN-GA model

Figure 2. Fuzzy reasoning

Figure 3. Configuration of the ANFIS model

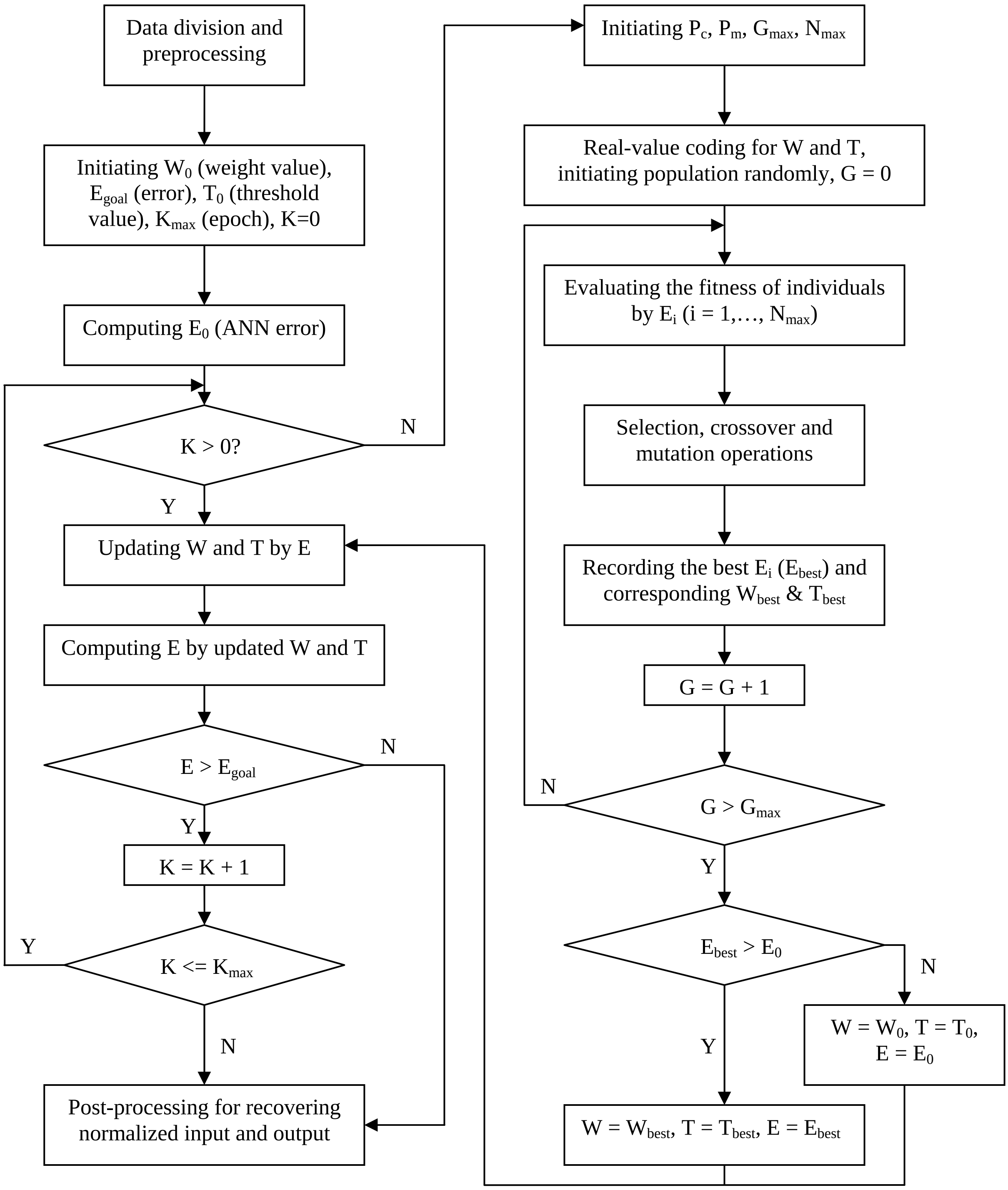
Figure 4. The study area

Figure 5. Performance of ANN-GA models against different numbers of nodes in hidden layer

Figure 6. Prediction results and absolute errors for validation data set with the ANN-GA model

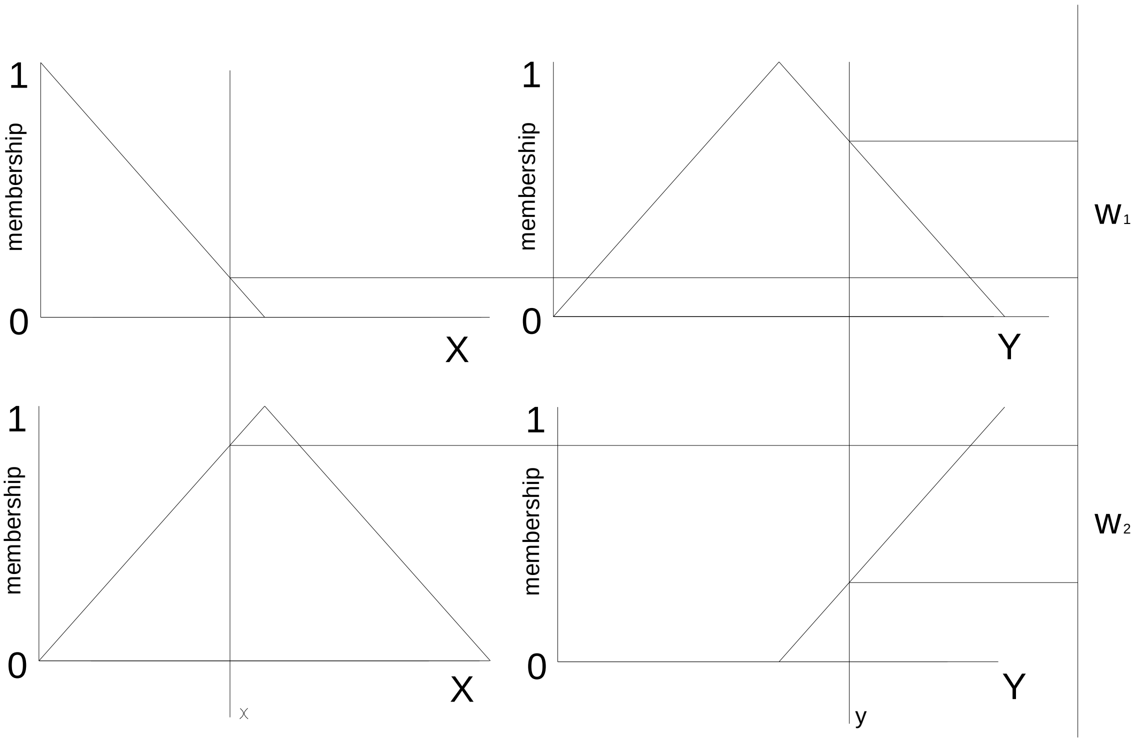
Figure 7. Optimized fuzzy membership functions

Figure 8. Comparison of absolute errors with different models



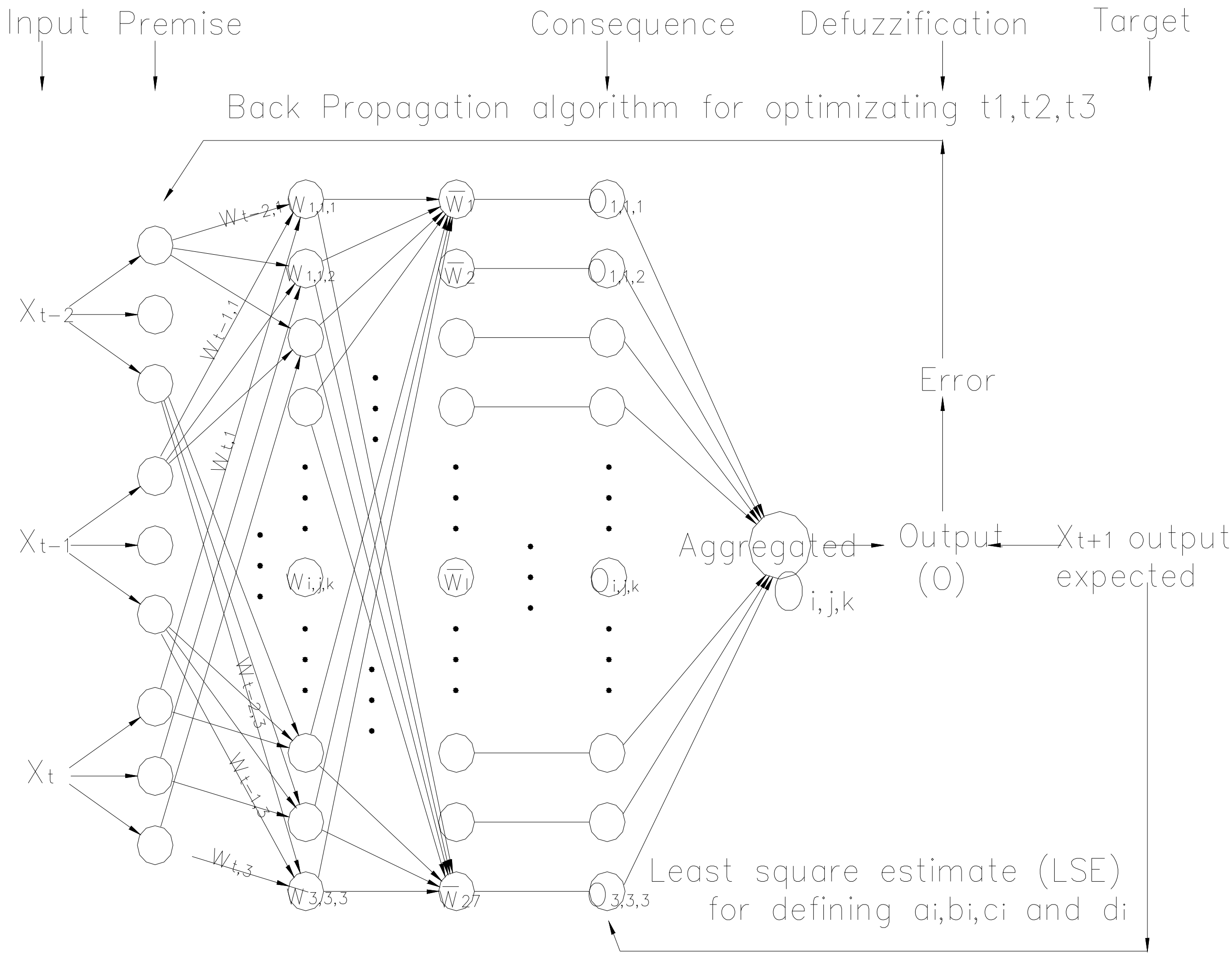
Data division and preprocessing
Initiating W_0 (weight value), E_{goal} (error), T_0 (threshold value), K_{max} (epoch), K=0
Computing E_0 (ANN error)
K > 0?
N
Y
Updating W and T by E
Computing E by updated W and T
E > E_{goal}
N
Y
K = K + 1
K <= K_{max}
Y
N
Post-processing for recovering normalized input and output
Initiating P_c, P_m, G_{max}, N_{max}
Real-value coding for W and T, initiating population randomly, G = 0
Evaluating the fitness of individuals by E_i (i = 1,…, N_{max})
Selection, crossover and mutation operations
Recording the best E_i (E_{best}) and corresponding W_{best} & T_{best}
G = G + 1
N
G > G_{max}
Y
E_{best} > E_0
N
Y
W = W_0, T = T_0, E = E_0
W = W_{best}, T = T_{best}, E = E_{best}

Figure 1.



$$\begin{aligned} o_1 &= a_1 x + b_1 y + d_1 \\ o_2 &= a_2 x + b_2 y + d_2 \end{aligned} \Rightarrow o = \frac{w_1 o_1 + w_2 o_2}{w_1 + w_2} = \overline{w_1} o_1 + \overline{w_2} o_2$$

Figure 2.



Input
Premise
Consequence
Defuzzification
Target
Back Propagation algorithm for optimizating t1,t2,t3
Error
Aggregated
Output
(O)
Xt+1 output
expected
Least square estimate (LSE)
for defining ai,bi,ci and di

Figure 3.

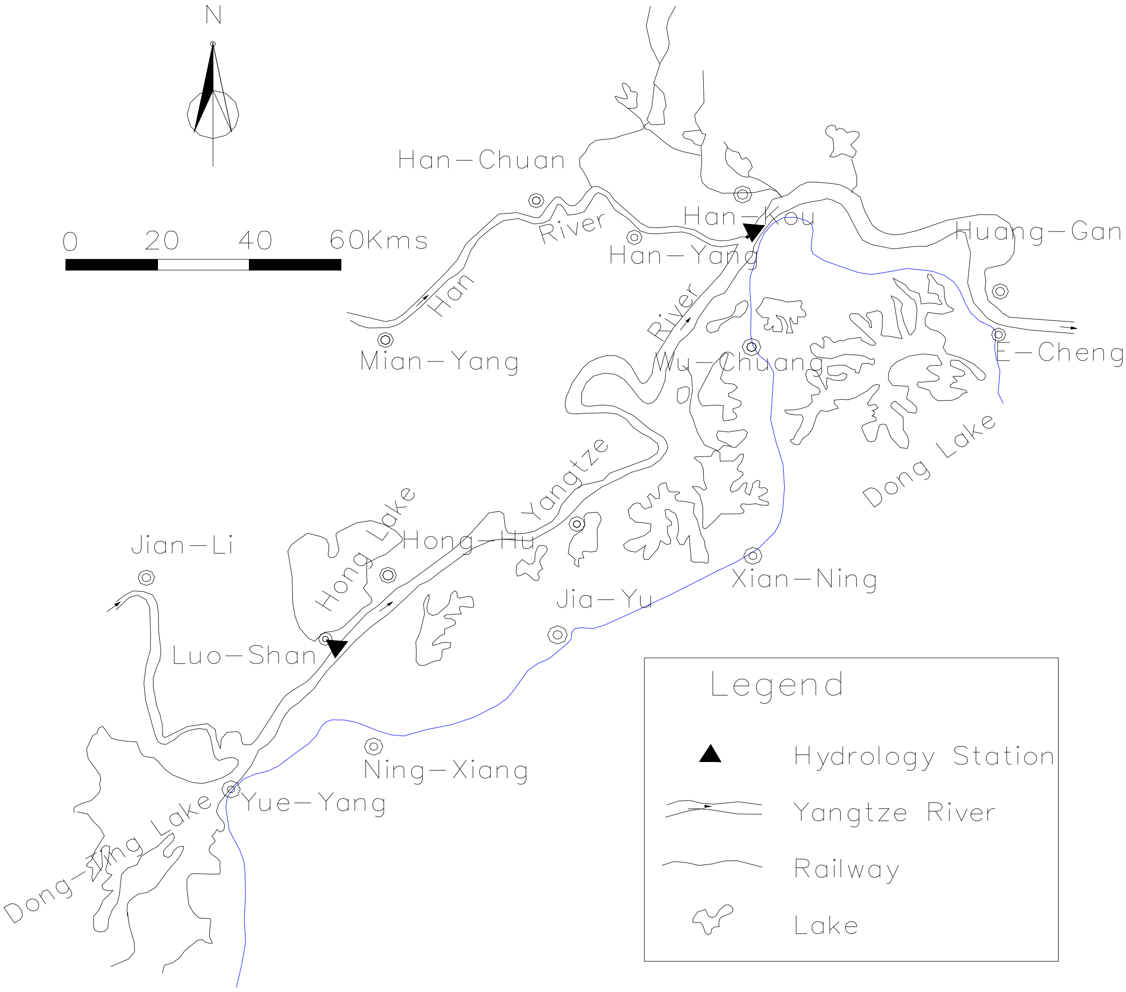

N
0
20
40
60Kms
Han-Chuan
Han River
Han-Yang
Han-Kou
Huang-Gan
Mian-Yang
River
Wu-Chuang
E-Cheng
Yangtze
Dong Lake
Hong Lake
Hong-Hu
Jian-Li
Jia-Yu
Xian-Ning
Luo-Shan
Ning-Xiang
Yue-Yang
Dong-Ting Lake
Legend
Hydrology Station
Yangtze River
Railway
Lake

Figure 4.

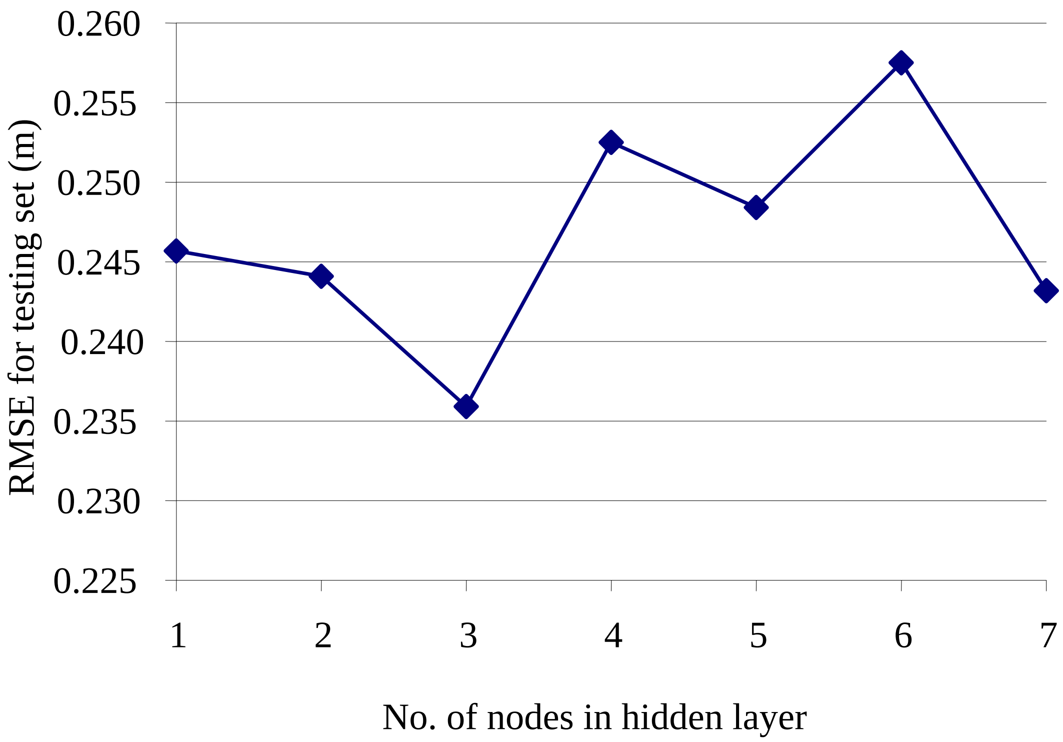

0.260
0.255
0.250
0.245
0.240
0.235
0.230
0.225
RMSE for testing set (m)
1
2
3
4
5
6
7
No. of nodes in hidden layer

Figure 5.

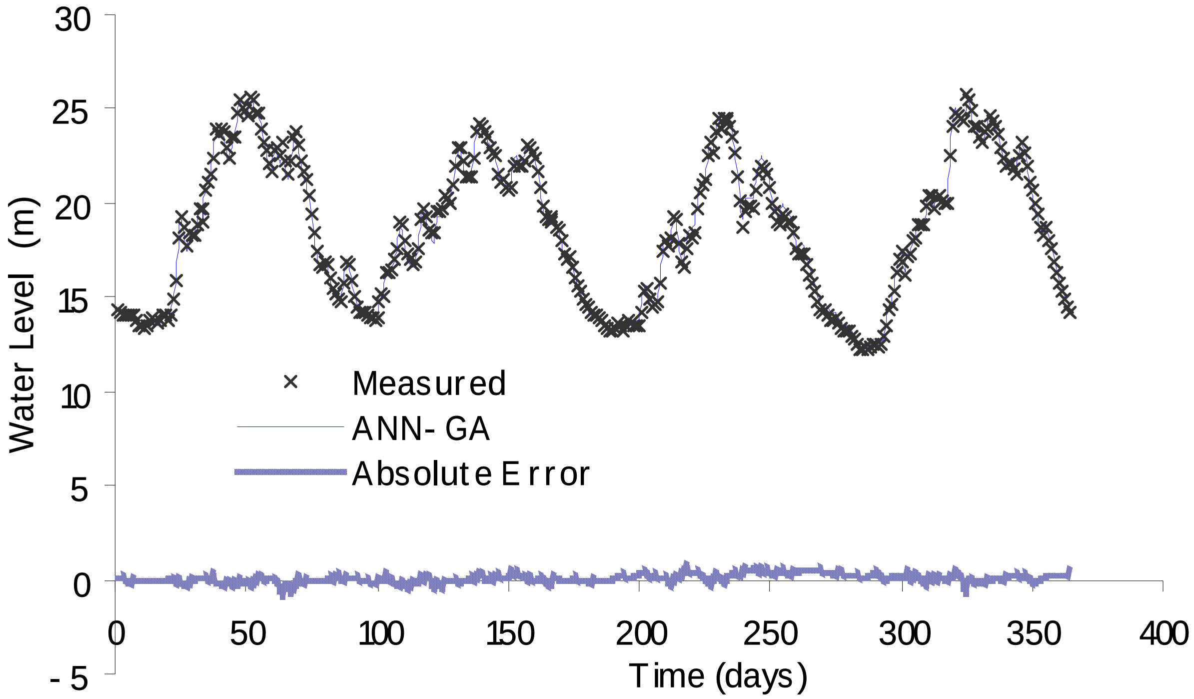

30
25
20
15
10
5
0
- 5
Water Level (m)
0
50
100
150
200
250
300
350
400
Time (days)
Measured
ANN- GA
Absolute Error

Figure 6.

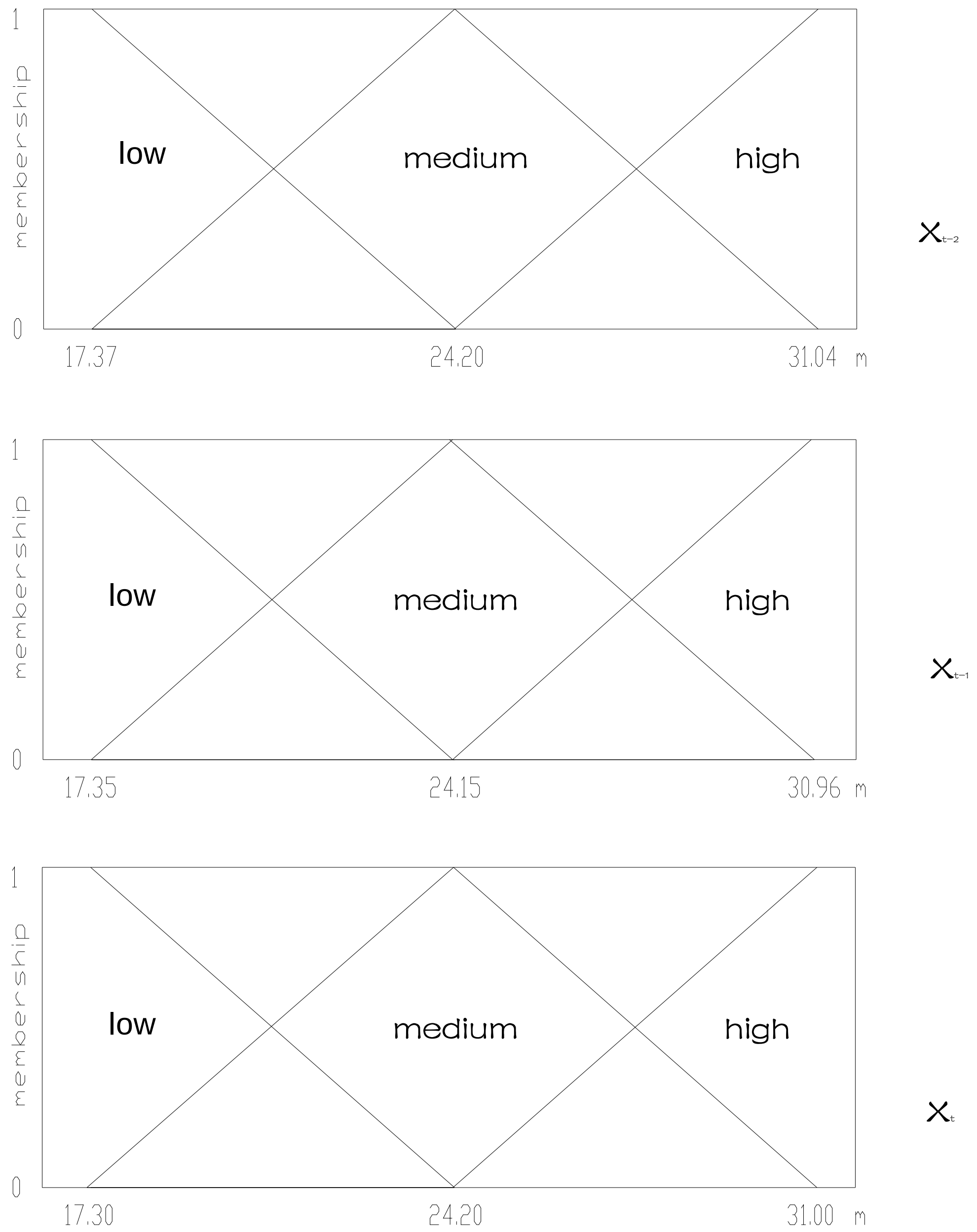

1
membership
low
medium
high
0
17.37
24.20
31.04 m
X_{t-2}
1
membership
low
medium
high
0
17.35
24.15
30.96 m
X_{t-1}
1
membership
low
medium
high
0
17.30
24.20
31.00 m
X_t

Figure 7.

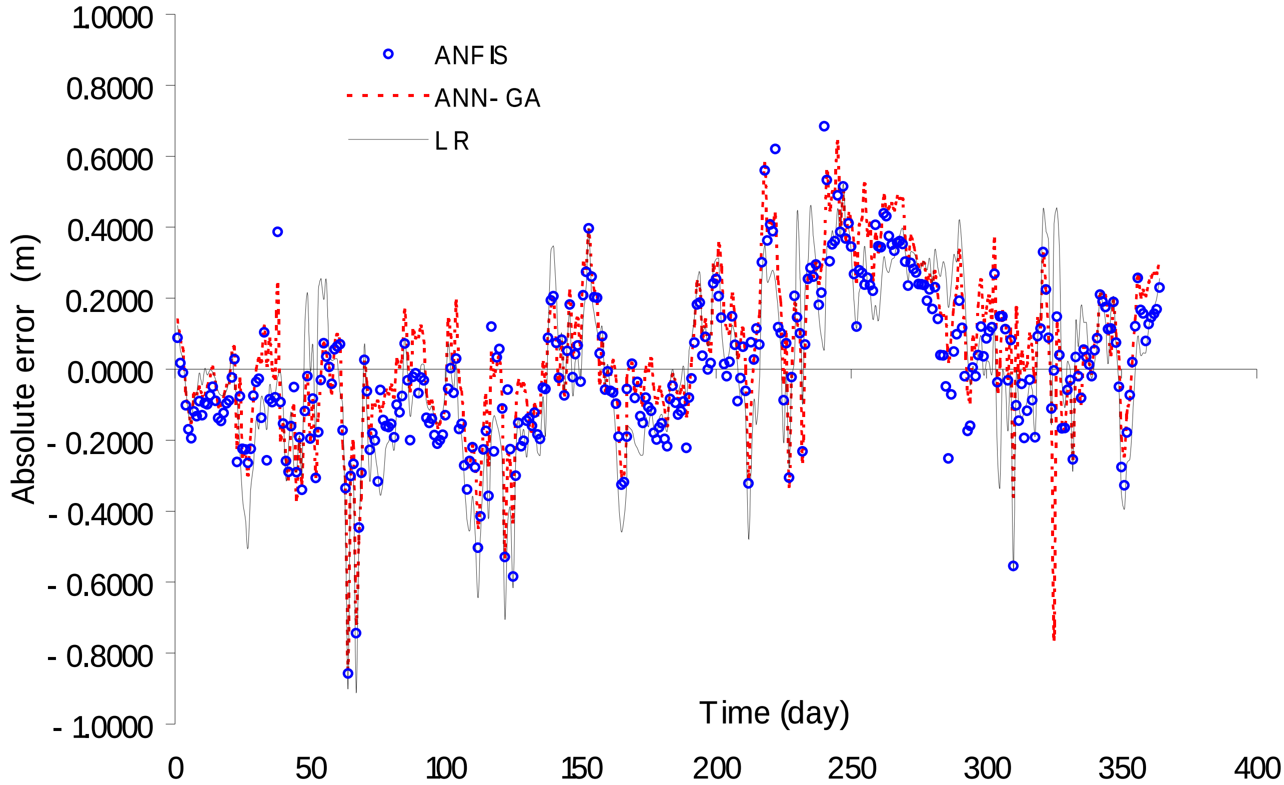

1.0000
0.8000
0.6000
0.4000
0.2000
0.0000
- 0.2000
- 0.4000
- 0.6000
- 0.8000
- 1.0000
Absolute error (m)
ANFIS
ANN- GA
LR
Time (day)
0
50
100
150
200
250
300
350
400

Figure 8.