Water Resour Manage (2013) 27:4469–4492
DOI 10.1007/s11269-013-0420-2

Prediction of Urmia Lake Water-Level Fluctuations by Using Analytical, Linear Statistic and Intelligent Methods

Hossein Kakahaji · Hamed Dehghan Banadaki · Abbas Kakahaji · Abdulamir Kakahaji

Received: 28 February 2013 / Accepted: 13 August 2013 /
Published online: 31 August 2013

Abstract Undoubtedly, the most significant factor with wise decision making and designing hydrological structures along the lake coasts is an accurate model of lake level changes. This issue becomes more and more important as recent global climate changes have completely reformed the behavior of traditional lake level fluctuations. Subsequently, estimating lake levels becomes more important and at the same time more difficult. This paper deals with modeling lake level changes of Lake Urmia located in north-west of Iran, in terms of both simulator and predictor models. According to this, two traditional simulator models based on water budget are developed which benefit from most effective components on water budget namely precipitation, evaporation, inflow and the lake level antecedents, as model inputs. Most famous linear modeling tools, Autoregressive with exogenous input (ARX) and Box-Jenkins (BJ) models are employed with the same mentioned inputs for prediction purpose. In addition, two other methods that are, Multi-Layer Perceptron (MLP) neural network and also Local Linear Neuro-Fuzzy (LLNF) are applied to investigate capability of intelligent nonlinear methods for lake level changes prediction. All models performances are indicated using both graph and numerical illustrations and results are discussed. Comparative results reveal that the intelligent methods are superior to traditional models for modeling lake level behavior as complex hydrological phenomena.

H. Kakahaji (✉)
Sama Technical and Vocational Training College, Islamic Azad University, Tehran Branch, Tehran, Iran
e-mail: h.kakahaji@srbiau.ac.ir

H. D. Banadaki
Department of Electrical Engineering, Islamic Azad University, Ashkezar Branch, Yazd, Iran
e-mail: h.dehghan@srbiau.ac.ir

A. Kakahaji
Department of Civil Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran
e-mail: a.kakahaji@gmail.com

A. Kakahaji
Water Resources Management Company of Iran, Tehran, Iran
e-mail: amirkakahaji@yahoo.com

Keywords Urmia Lake · Lake level · Simulation · Prediction · PNN · LLNF

1 Introduction

Descending trend of Lake Uremia's water level as one of the most significant lakes of Iran and the second largest salt water lake on the earth goes to the extent that in addition to conservationist groups, has also attracted the attention of others interested in this unique ecosystem.

Wetlands can be considered as greatest biologically diverse of all ecosystems, serving as home to a wide variety of plant and animal life. In addition, wetlands play various roles in the environment, principally water sanitization, flood control, and shoreline stability. Lake Urmia has been shrinking for a long time; in 1996 the lake water surface elevation was reported as 1276.7 m and ever since it has had a decreasing trend. Unfortunately, it has approximately shrunk by 60 % and has converted to a national concern. Various opinions have been expressed about any possible factors and their influence on this dropping trend. Undoubtedly, demographic developments, anthropogenic, agricultural developments, water resources management and dams, natural and climatic factors all effect on current situation and cause exacerbating this trend.

Modeling (including simulator and predictor models) of the water level fluctuations of Urmia Lake, which is an endangered ecosystem and a nonlinear complex natural phenomenon, can help in attenuating the damages caused by natural hazards such as drought and flood (Roshan et al. 2013; Tabari et al. 2013). Evidently, devising a model as accurate as possible can lead to a reliable decision supporting system for trading off between agricultural, industrial, and other needs and ecosystem requirements. For instance, as a real problem, Artemia (a kind of shrimp), which is the only aquatic species living in the lake, needs a minimum amount of water to survive. A precise predictor model can determine if such an amount is possible to be supplied. If not, then based on the current priorities and outcomes of the model, a wise and sustainable decision can be made.

From another perspective, the lake may be considered as a reservoir system which needs to be controlled and optimized: for controlling, an appropriate dynamic model of the process (i.e. the lake) must be embedded in the control loop. For optimization, a reliable dynamic model of the lake can be incorporated in the form of constraints or objective functions of an optimization model (Ostadrahimi et al. 2012; Afshar et al. 2011).

Among the previous studies, an improved wavelet model was proposed for hydrologic time series prediction by Sang (2013). The method was applied to four hydrologic cases which results indicated the better performance compare to pure wavelet model. Marce et al. (2004) provided an Adaptive Neuro-Fuzzy Inference System (ANFIS) model for estimating fluvial nutrient loads in watersheds under time varying human impact, and reported that ANFIS outperformed models based on rating curves and ratio estimators. Khu et al. (2001) and Babovic and Keijzer (2002) used Genetic Programming (GP) to runoff prediction and results showed promising accuracy compared with Auto Regression (AR) and Kalman filter as the updating models. Also, Muttil and Liong (2001), Liong et al. (2002) and Aytek and Alp (2008) addressed use of GP for creating rainfall-runoff models. Ghorbani et al. (2010) applied the GP for modeling sea level at the Hillary Boat Harbour and performance compared with output data obtained by Artificial Neural Network (ANN).

Jones et al. (2001) investigated historical lake levels and recent climate changes at three closed lakes in Western Victoria Australia, and they believed diminishing the rate of precipitation to evaporation causes slaking water budget. Other related works can be found in literature (Kadıoğlu et al. 1999; Bengtsson and Malm 1997). Astushi et al. (2004) tried to find the influence of global climate changes on water budget in Lake Biakal, southern Siberia during the past 100,000 years. They inferred that more water was supplied to the lake within warm stages. Altunkaynak (2007) employed ANN for producing 1 month ahead predictive model and applied it to water-level changes prediction of Lake Van in Turkey. The results were compared with Autoregressive Moving Average Model with exogenous inputs (ARMAX) model. Vaziri (1997) applied ANN and Autoregressive Integrated Moving Average (ARIMA) model for predicting Caspian Sea water surface and claimed that ANN underestimated the sea level by 3 cm whereas the ARIMA overestimated it by 3 cm. The research introduced by Güldal and Tongal (2010) used Recurrent Neural Network (RNN) and ANFIS for predicting water levels in the Lake Egirdir which is one of the biggest fresh water lakes of Turkey. They concluded that both model have good ability to prediction lake level changes. The ability of ANFIS method to prediction of sea level variations at a tide gauge site in the Hillarys Boat Harbour, Western Australia investigated in (Shiri et al. 2011). Daily water level changes of Lake Iznik in Turkey were predicted using ANFIS, Gene Expression Programming (GEP), and ANN models in (Kisi et al. 2012). The results obtained by the GEP approach indicated better performance than ANFIS and ANN.

Hassanzadeh et al. (2012) investigated main factors which cause reducing the Lake Urmia water level. They determined biggest contributors to descending trend are changes in inflows, constructing dams, and less precipitation, which are responsible for the problem of 65 %, 25 % and 10 %, respectively. Talebizadeh and Moridnejad (2011), in an engrossing research, developed ANN and ANFIS models to predict the Urmia Lake level changes. They found that the results of ANFIS model are superior to ANN. In other works conducted on the Urmia Lake, Kavehkar et al. (2011) examined the performance of GP compare to ANN in orther to prediction aim and noted that the GP perform marginally better for most of the cases. Karimi et al. (2012) used two intelligent methods, that is, ANFIS and GEP, for predicting water level fluctuations of Urmia Lake. The findings indicated that GEP surpassed the ANFIS model (see section 5 for more details). Exploiting ANFIS and ANN models for predicting sea level in Darwin Harbor Australia was subject of another research (Karimi et al. 2013). The outcomes expressed the same performance for both mentioned models. In this paper six models are developed include two mathematical models (water balance and multiple correlation equation), two dynamic linear models (ARX and BJ), as well as a couple of intelligent models (MLP and LLNF) are examined and result are compared and discussed. The last mentioned model (LLNF) is expected to be new in this particular application as we did not find any prior instance in the literature. In addition, the stated models were not employed all together simultaneously in a particular case.

2 Materials and Methods

2.1 Water Balance Model

The Urmia Lake's watershed is a closed basin. As a result, it is expected that using water balance equation (WBE), water level fluctuations of the lake (as terminal of the watershed) can be estimated with rough approximation. It should be noted that due to declining trend of groundwater levels around the lake, especially in recent decades, infiltration of groundwater into lake is small and consequently negligible. According to these descriptions the changes in the lake level is expressed as follow:

$$\begin{aligned} \Delta H(k) &= P(k) - E(k) + 0.001\left(\frac{R_{in}(k)}{A(H)}\right) \\ \widehat{H}(k) &= \widehat{H}(k-1) + \Delta H(k) \end{aligned} \tag{1}$$

Where $P(k)$, $E(k)$, $R_{in}(k)$ and $A(H)$ represent precipitation on the lake surface (mm), evaporation at the lake surface (mm), the volume of incoming flow (m^3) and lake area (km^2), respectively. Here k indicates a particular month. Also, $\widehat{H}(k)$ and $\Delta H(k)$ represent lake lave and level changes in k-th month.

2.2 Multiple Regression Model

Assuming changes in lake water level are independently associated with a linear combination of the original components, those influence on water budget (precipitation on the lake surface, evaporation at the lake surface and volume of incoming flow), the *multiple regression* method (Chatterjee and Hadi 1986) has been used for simulation of monthly changes in water level of the lake. This method estimates coefficients for a multilinear regression of the output data (i.e., lake level changes) on the inputs (i.e., $P(k)$, $E(k)$, $R_{in}(k)$) by using *least squares* technique. The calculated equation by mentioned method can be formed as follow:

$$\begin{aligned} \Delta H(k) &= 0.4425 \times P(k) - 0.5007 \times E(k) + 1.1846 \times Q(k) - 48.907 \\ \widehat{H}(k) &= \widehat{H}(k-1) + \Delta H(k) \end{aligned} \tag{2}$$

Where $Q(k)$ is the height equivalent of the incoming flows volume in month k.

2.3 Linear Predictor Models

Black-box modeling should be used in the cases which fitting data regardless of certain mathematical structure of the model is interested. There are ensembles of *linear* and *nonlinear* black-box models that are able to describe behavior of dynamic systems. The complexity of these models (computational and time complexity) could be different depending on desired flexibility. Practically, application of linear models is conventional for modeling nonlinear systems with low nonlinearity properties, because applying linear models in these cases could lead to satisfying results. According to these reasons, it is worthwhile to start from simple linear model structures. If predefined results are not being obtained, thus progressing toward models that are more complex will be justifiable. It has been mathematically proven that the least squares error (LSE) method is the optimal modeling method in the case of linear systems (Ljung 1987; Nelles 2001). In order to establish a linear model, a finite sequence of the input-output measured data with a constant sampling interval (monthly records) is considered. If there are dynamic linear relations among these variables, they can be described by the following model:

$$\widehat{y}(k) = \sum_{j=1}^{n} a_j y(k-j) + \sum_{l=1}^{p} \sum_{j=1}^{m_i} b_{lj} u_l(k-j) \tag{3}$$

The Eq. 3 describes linear Multi-Input Single-Output (i.e., MISO system, the plant under study) discrete-time system, where a_j and b_{lj} are parameters, m_i denotes the order of the j-th input, and p is the number of inputs, and n is the order of output, usually of the same value (Nelles and Isermann 1996). Among the linear predictor methods, ARX and BJ will be employed to predicting the lake level changes at one month ahead. Two next equations show ARX and BJ models, respectively.

$$y(k) = B(q)/A(q)\,\underline{u}(k) + 1/A(q)v(k) \tag{4}$$

$$y(k) = B(q)/F(q)\,\underline{u}(k) + C(q)/D(q)v(k) \tag{5}$$

Models defined by Eqs. 4 and 5 belong to an identifier family which is named *linear dynamic models*. In addition, both are time-domain models and are subcategory of parametric system identification techniques. They are constructed by two transfer functions include input linear transfer function (ITF) and noise transfer function (NTF). Therefore, the output $y(k)$ can be obtained by passing the input $u(k)$ through the ITF in addition to passing white noise $v(k)$ through NTF. The former usually termed deterministic part, and the later also is called stochastic part. The transfer functions parameters of ARX model can be calculated by linear least squares technique, while typically the parameters of BJ are estimated using nonlinear optimization techniques (Nelles 2001; Ljung 1987). Where in Eqs. 4 and 5, q denotes time shift operator i.e., $q^{-1}H(k) = H(k{-}1)$. It is noteworthy that in the present paper, $y(k)$ refers to lake level (H), $u(k)$ refers to inputs including precipitation (P), evaporation (E), and inflow summation (R_{in}).

2.4 Static Multi-layer Perceptron Network

All previous mentioned methods can be specified as *linear* tools, Fig. 1 shows the *nonlinear* prediction method based on a parallel model. As seen, for prediction purpose, the inputs/output recorded data are available, previous inputs and outputs are injected into the static model. Thus the input vector can be defined as Eq. 6.

$$\underline{X} = \left[U_1(k), U_2(k), \ldots, U_p(k), y(k{-}1), y(k{-}2), \ldots, y(k{-}n)\right]^{\mathrm{T}} \tag{6}$$

Here, $y(k-n)$ is n-th past value of real output and $U_p(k)$ contains past values of the p-th input process according to:

$$U_p(k) = \left[u_p(k), u_p(k{-}1), u_p(k{-}2), u_p(k{-}3), \ldots, u_p(k{-}m_i)\right] \tag{7}$$

Where m_i denotes the order of p-th input, and n is the output order.

Note, in the current study process refers to the Urmia Lake. In practice, any nonlinear static approximator can be used instead of internal nonlinear model. Indeed Fig. 1 is consisting of two parts: a nonlinear static approximator and external dynamics. Thus, the resulted model is able to approximate nonlinear dynamic systems.

Although static MLP networks can be utilized for modeling static systems as well as the systems with low grade of dynamism, it is better to use them in a dynamic form to identify model of process with high grade of dynamism. A static multi-layer perceptron can be used instead of nonlinear static model indicated in Fig. 1, despite other probable choices are also possible (e.g., LLNF model). Generic structure of a static multi-layer perceptron neural

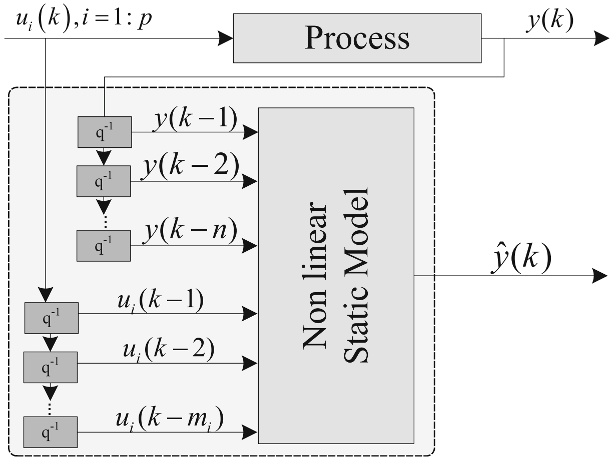

Fig. 1 Nonlinear one-step predictor method based on nonlinear static model

network is shown in Fig. 2. It can be seen none of the past recorded inputs and outputs values are applied into the network (Nozari et al. 2012a, b).

Since it has been proven that MLP network with one hidden layer has a better performance in prediction aims (Nelles 2001; Cybenko 1989; Hornik et al. 1989), in the present work, the MLP model architecture has one-hidden-layer which is depicted in Fig. 2. Moreover, it must be noted that since recorded data are positive, the activation functions of hidden neurons are decided to be logistic function.

In training process, in order to minimizing error, neural networks try to update their parameters based on learning algorithm and consequently produce proper output associated with the given input. There are many learning algorithms such as Gauss-Newton (GN), Gradient Descent (GD), and Levenberg-Marquardt (LM) (Hagan and Menhaj 1994). We decided to use the LM algorithm to update the MLP network parameters because of the

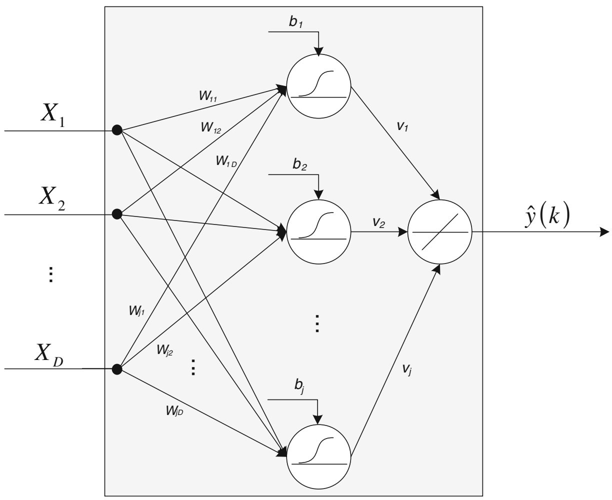

Fig. 2 Topology of a static feed forward MLP network

following advantages: (1) faster convergence compared with the GD method, (2) more robustness compared with the GN method, (3) its interpolation is between the GN and GD methods (i.e., it has the speed of GN and the convergence of GD) (Haykin 1999). The LM algorithm updates the parameters of the MLP neural network according to Eq. 8.

$$\begin{cases} W_{n+1} = W_n + \Delta W \\ \Delta W = -\left(\left[J^T(W) J(W) + \mu I \right]^{-1} \right) \left(J^T(W) e \right) \end{cases} \tag{8}$$

Where e is the error function, J is a Jacobian matrix, and μ is a scalar that makes LM closed to either GD or GN. Here, W contains the weights of the network and defined as follows:

$$W = [w_{110}\, w_{111} \ldots w_{11D}\, w_{1L0}\, w_{1L1} \ldots w_{1LD} \;\; w_{210} \;\; w_{211} \ldots w_{21L}] \tag{9}$$

2.5 Static Locally Linear Neuro-Fuzzy Network

Approximating the best function that characterized the nonlinear complex behavior of the nonlinear process, represents main concern in designing predictor models (Banadaki et al. 2011; Nozari et al. 2012a, b). The fundamental idea behind the LLNF model is dividing the input complex space into smaller and thus simpler (i.e., linear) subspaces. Next, a fuzzy validity function must be assigned to each subspace which determines the validity percentage of each linear model within its region. Therefore, a neuron is constructed by a validity function in addition to its local linear model (LLM). The LLNF model can easily be interpreted as Takagi-Sugeno model where each neuron represents one rule, the validity functions represent the rule premise, and the LLMs represent the rule consequents (Nelles 2001). The general network structure of LLNF model is shown in Fig. 3.

Therefore, in order to constructing a nonlinear dynamic model, indicated topology in Fig. 3 could be a worthy candidate to be replaced with nonlinear static model shown in Fig. 1. With respect to input vector which defined by Eqs. 6 and 7, a multi-input single-output (MISO) dynamic LLNF model for D inputs is presented in Eq. 10.

$$\begin{aligned} \hat{y}(k) = \sum_{j=1}^{M} \sum_{i=1}^{D} & \left[b_{ji1} u_i(k-1) + b_{ji2} u_i(k-2) + \ldots + b_{jim_i} u_i(k-m_i) \ldots \right. \\ & \left. - a_{j1} y(k-1) - a_{j2} y(k-2) - \ldots - a_{jn} y(k-n) + \xi_j \right] \varphi_j\left(\underline{X}\right) \end{aligned} \tag{10}$$

Where b_{jim_i} and a_{jn} represent the numerator and denominator coefficients respectively, ξ_j is the offset of the LLM$_j$. Furthermore, $\varphi_j(\underline{X})$ is the operating point dependent weighting factors.

The parameter vector for j-th LLM is formed as:

$$\underline{w}_j = \left[b_{j11}, b_{j12}, \ldots, b_{j1m_1}, \ldots, b_{jp1}, b_{jp2} \ldots, b_{jpm_p} a_{j1}, \ldots, a_{jn}, \xi_j \right]^{\mathrm{T}} \tag{11}$$

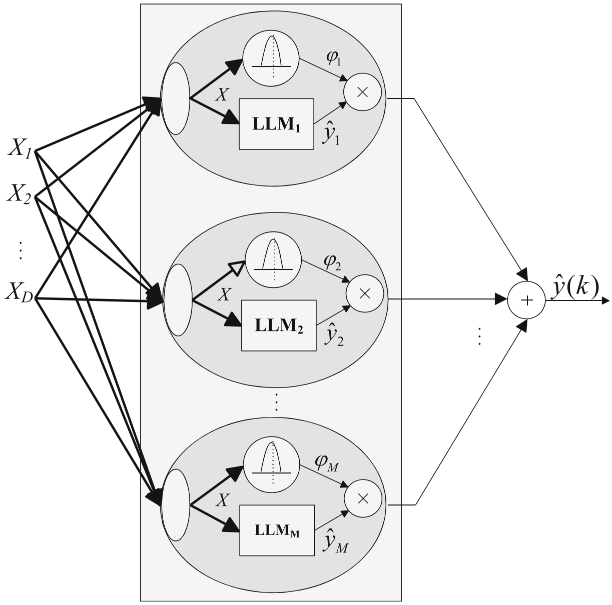

Fig. 3 Topology of static locally linear neuro-fuzzy model

These parameters are locally estimated through a weighted least-squares solution and form the output of j-th LLM. Note that an LLM is only valid in the region where the associated validity function is close to one.

The validity functions depicted in Fig. 3 are normalized Gaussian functions, and can be more formally describe by following equations:

$$\sum_{j=1}^{M} \varphi_j\left(\underline{X}\right) = 1 \tag{12}$$

$$\varphi_j\left(\underline{X}\right) = \frac{\mu_j\left(\underline{X}\right)}{\sum_{j=1}^{M} \mu_j\left(\underline{X}\right)} \tag{13}$$

$$\begin{aligned} \mu_j\left(\underline{X}\right) &= exp\left(-\frac{1}{2}\left(\frac{\left(X_1 - c_{j1}\right)^2}{\sigma_{j1}^2} + \ldots + \frac{\left(X_D - c_{jD}\right)^2}{\sigma_{jD}^2}\right)\right) = \ldots \\ & exp\left(-\frac{1}{2}\left(\frac{\left(X_1 - c_{j1}\right)^2}{\sigma_{j1}^2}\right)\right) \times \ldots \times exp\left(-\frac{1}{2}\left(\frac{\left(X_D - c_{jD}\right)^2}{\sigma_{jD}^2}\right)\right) \end{aligned} \tag{14}$$

Where $D = n + \sum_{j=1}^{p} m_j$, c and σ are the center coordinate and the individual standard deviation, respectively.

Since LLNF (as a local modeling approaches) can be categorized into a big family of algorithms which is called *divide and conquer strategy*, division strategy which applied to the primary complex problem, becomes a crucial factor for success of such an approach. In order to partitioning input space, and in turn, tuning the rule premise parameters (i.e., determining the validation hypercube for each LLM) a progressive tree-configuration algorithm namely locally linear model tree (LOLIMOT) was proposed in (Nelles 2001). The following pseudo code presents the general overview of the LOLIMOT algorithm.

LOLIMOT ALGORITHM

$c_1 = \min(\underline{X}) + (\max(\underline{X}) - \min(\underline{X}))/2$
$\sigma_1 = \max(\underline{X}) - \mathrm{Min}(\underline{X})/3$
$\varphi_1 \leftarrow$ Calculate $(c_1, \sigma_1, \underline{X})$
$\underline{w}_1 \leftarrow$ Least Squares (φ_1, X, y)
$\hat{y}_{LLM_1} \leftarrow \underline{X} \times \underline{w}\,;\ \hat{y} \leftarrow \hat{y}_{LLM_1} \times \varphi_1; e \leftarrow J(y - \hat{y})$
$e_{Local_{LLM_1}} \leftarrow \varphi_1 \times e\,;\ e_{Local} \leftarrow e_{Local_{LLM_1}}$

For i=1 To LLM_{max}
 ($Neuron_{new}$, $Neuron_{del}$) ←Select-Delete Worst Neuron ($Neuron_{old}$, e_{local})
 For l=1 To Input Dimensions
  Splitting region ←Find min-max bound ($Neuron_{del}$)
  (c_{new},σ_{new}) ←Create new neuron (Splitting region)
  Updated model←Insert (c_{new}, σ_{new}, Splitting Dimension)
  For k=1 To i+1
   $\varphi_k \leftarrow$ Calculate$(c_k, \sigma_k, \underline{X})$
   $w_k \leftarrow$ Least Squares$(\varphi_k, \underline{X}, y)$
   $\hat{y}_{LLM_k} \leftarrow \underline{X} \times \underline{w}$
  End
  $\hat{y} \leftarrow \hat{y}_{LLM} \times \varphi, \mathrm{e}_l \leftarrow J(y - \hat{y})$
  e ← Select minimum cost neuron(e_l)
 End
 If e_l < Error Goal
 Break
 For k=1 to i+1
  $\mathrm{e}_{Local_{LLM_k}} \leftarrow \varphi_k \times e$
 End
End

3 Models Evaluation Criteria

In order to evaluate models accuracy, all of them are assessed by three most famous criteria including Root Mean Squares Error (RMSE), correlation coefficient (R), and similarity:

$$RMSE = \left(\frac{1}{Q}\sum_{j=1}^{Q}\left(H(j)-\widehat{H}(j)\right)^2\right)^{1/2} \tag{15}$$

$$Similarity = 100*\left|1-\frac{\left\|\widehat{H}-H\right\|_2}{\left\|H-\overline{m}(H)\right\|_2}\right| \tag{16}$$

$$Correlation\ \ Coefficient = \frac{\sum_{j=1}^{Q}\left(H(j)-\overline{m}(H)\right)\left(\widehat{H}(j)-\overline{m}\left(\widehat{H}\right)\right)}{\left(\sum_{j=1}^{Q}\left(H(j)-\overline{m}(H)\right)\sum_{j=1}^{Q}\left(\widehat{H}(j)-\overline{m}\left(\widehat{H}\right)\right)\right)^{1/2}} \tag{17}$$

Where $\|.\|_2$ denotes Euclidean distance and Q is the number of data samples that are considered for modeling, $\overline{m}$ is the mean value of the measured lake levels, $\widehat{H}$ and H are predicted and measured lake levels, respectively. Note, similarity criterion indicates fitness percentage between measured and model outputs.

4 Descriptions of Study Area and Data

The Urmia Lake located in the south-west of Iran, it has Urmia city in 18 km to the east. It is also focal accumulation of excess surface flows of all rivers in closed Urmia basin. In 1967, Urmia Lake and its islands (except Islami Island) were declared a protected area. In addition, according to Man and the Biosphere (MAB) it has been recognized as a biosphere reserve. Approximate length of the lake is 146 km and its width varies between 15 and 58 km. Moreover its area, height from sea level and water volume are about 5,000 km^2, 1,274 m and 32,000 million cubic meters, respectively. The Urmia Lake's watershed is a small closed basin located in Azarbaijan region which is surrounded by highlands on the Iran border with Turkey, and Azarbaijan mountains at 44° 7′–47° 53′ eastern longitudes and 35° 40′–38° 30′ northern latitudes (Iranian Society of Consulting Engineers 2011). It occupies 3.15 % of Iran area, and also about 7 % of the total surface water resources of the country belong to the lake (Fig. 4).

From the total area of the watershed, mountain areas, about 35,150 km^2, plains and foothills, about 9,000 km^2 and finally the lake and marshy lands around it, about 7,310 km^2, allocate to themselves. Totally, 19 rivers are emptying into the lake that major sub basin of them consist of Baranduzchai and Zolachai to the west, Daryanchai and Tesujchai to the south-east, Azarshahrchai and Sufichai to the east, the Urmia lake on the center, Zarinehrood, Leylanchai, Mardoghchai and Siminehrood to the south and finally Godarchai on the south-west (Iranian Society of Consulting Engineers 2011).

Spring with 59.05 % and fall with 9.64 % of the total watershed runoff are most water filled and driest seasons. Moreover, the runoff of winter and summer are reported as 10.56 % and 10.89 % respectively, and 51 % of whole surface water flows account for base flow (Water Resources Management Company of Iran 2011).

According to research purposes, required statistics and data including input streams flow into the lake, direct precipitation on the lake surface and evaporation in the period from 1967–1968 to 2006–2007 have been collected from regional stations. Statistics defects of data sets have been filled by using correlation method. Based on monthly records measured

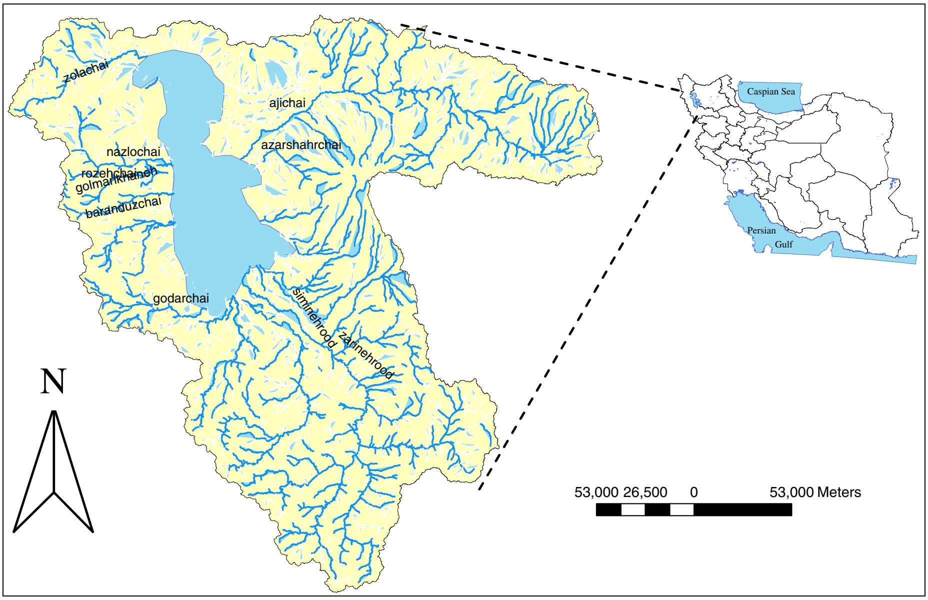

Fig. 4 Location of Urmia Lake and its watershed

at Golmankhaneh station during stated time period, the lake water level fluctuations were obtained and used for modeling and validation stages. An overview of major statistic components of total measured data sets can be seen in Table 1.

Each data is divided into two subsets namely training and validation data sets. The data samples from 1 to 300 (1967 to the end of 1991) considered for training models and data during the period of 1992 to 2006 (301:480) used for models validation.

Table 1 Statistic characteristic of available data

Variable	Maximum	Minimum	Average	Standard deviation
Precipitation−whole data set (mm/month)	131.10	0.00	21.73	22.07
Precipitation−training data set	131.10	0.00	21.96	22.29
Precipitation−validation data set	105.00	0.00	21.35	21.74
Summation of inflows−whole data set (m^3/s)	1332.41	0.82	145.90	206.20
Summation of inflows−training data set	1332.41	1.44	153.70	221.94
Summation of inflows−validation data set	867.21	0.82	113.31	172.65
Evaporation−whole data set (mm/month)	288.88	2.67	96.80	73.92
Evaporation−train data set	288.88	3.13	95.57	73.96
Evaporation−validation data set	254.61	2.67	98.80	73.83
Lake level−whole data set (m)	1278.39	1272.83	1275.70	1.25
Lake level−training data set	1277.59	1273.81	1275.88	0.60
Lake level−validation data set	1278.39	1272.83	1275.42	1.85

5 Results and Discussions

Dynamic order determination problem for nonlinear systems is still not satisfactorily solved and a few attempts have been devoted to this essential issue. It is very common to select the dynamic order of the model by a combination of trial and error together with prior knowledge about the process (if available) (Nelles 2001; Billings and Voon 1987, 1986). In this paper, a suitable order selection strategy for linear process based on autocorrelation and cross correlation methods are used. As a result, the estimation of the effective dynamic orders can be done based on data and with developing a few numbers of models. Although mentioned methods (i.e., autocorrelation and cross correlation) are directly applicable for linear dynamic systems, utilizing them could lead to an educated guess about dynamic order of nonlinear systems, as well.

Figure 5 shows the correlation tests in order to estimating the range of order for the lake's three inputs, based on cross correlation method (between each input and output). This figure also depicts the results of autocorrelation for estimating the range of output order. According to cross correlation values for precipitation and inflow (Fig. 5a and b) 14-th dynamic and around it, also second dynamic and around it (regarding magnitude of them), was examined in term of their effect on output error of the models. Similarly, this procedure was tested for determining the probable effective antecedents of evaporation (Fig. 5c). The results showed first and 12-th dynamics have had tangible influence on the performance of the models. Moreover, regarding Fig. 5d, the autocorrelation plot proposed 1st dynamic as more effective antecedent. Therefore by a combination of obtained results from correlation methods, trial and error, and prior knowledge about the phenomena, appropriate linear (ARX and BJ) and nonlinear (MLP and LLNF) structures have been designed and tested.

Selection the number of neurons and also rules in ANN and fuzzy-based models, always has been raised as a main concern. In addition, improper selection of them may lead to wasteful complexity and over parameterization of models. In this section of article we

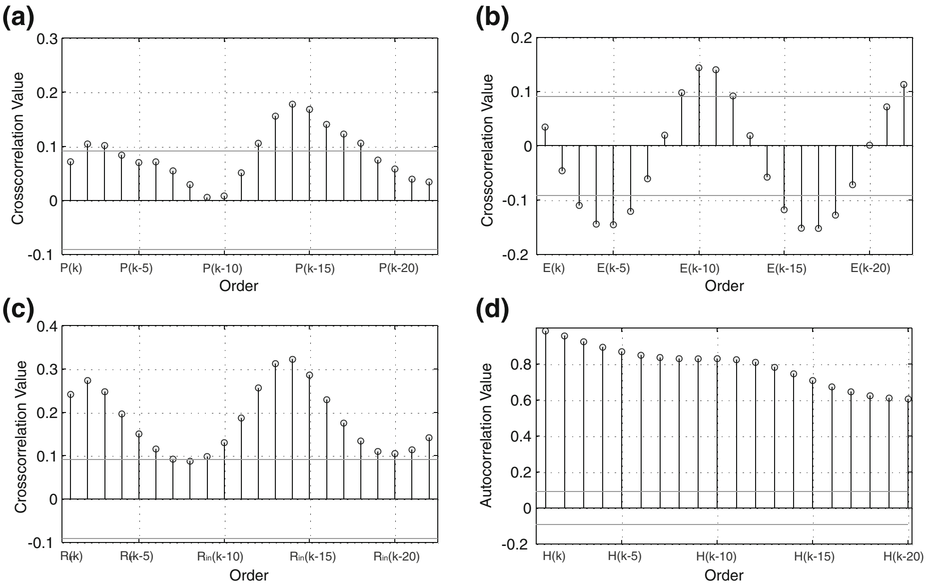

Fig. 5 Cross correlation on inputs (P, E and R_{in}) and autocorrelation on output (H)

Table 2 The structure of all nonlinear models

Model	Neuron*/rule** number	Input structure	Input number
SNM-1*	5	$H(k)=f(P(k), E(k), R_{in}(k))$	3
SNM-2*	5	$H(k)=f(P(k), E(k), R_{in}(k))$	3
SFM-1**	2	$H(k)=f(P(k), E(k), R_{in}(k))$	3
SFM-2**	2	$H(k)=f(P(k), E(k), R_{in}(k))$	3
DNM-1*	16	$H(k)=f(P(k-1), P(k-2), E(k-1), E(k-2), R_{in}(k-1), R_{in}(k-2), H(k-1))$	7
DNM-2*	23	$H(k)=f(P(k-1),\ldots, P(k-4), E(k-1),\ldots, E(k-4), R_{in}(k-1),\ldots, R_{in}(k-4)), H(k-1))$	13
DNM-3*	29	$H(k)=f(P(k-1),\ldots, P(k-6), E(k-1),\ldots, E(k-6), R_{in}(k-1),\ldots, R_{in}(k-6), H(k-1))$	19
DFM-1**	5	$H(k)=f(P(k-1), P(k-2), E(k-1), E(k-2), R_{in}(k-1), R_{in}(k-2), H(k-1))$	7
DFM-2**	8	$H(k)=f(P(k-1),\ldots, P(k-6), E(k-1),\ldots,E(k-6),R_{in}(k-1),\ldots, R_{in}(k-6),H(k-1))$	19
DFM-3**	11	$H(k)=f(P(k-1),\ldots, P(k-11), E(k-1),\ldots, E(k-11), R_{in}(k-1),\ldots, R_{in}(k-11), H(k-1))$	34

employ the RMSE curve to determine number of neuron/rule for all MLP and LLNF models in both static and dynamic cases. A typical RMSE curve for dynamic neuro-fuzzy model one (DFM-1, see Table 2) model is shown in Fig. 6 which shows variations of both training and validation sets with respect to the increment of rule numbers. As can be seen, while the rule number increased to 4, the RMSE values of training and validation sets declined steadily. Whereas, when the number of rules became greater than 5 the validation RMSE graph separated from training values. In other words, the critical point that determines the number of neuron (or rule) is where two graphs start separation from each other. The same procedure has been done for obtaining suitable number of neuron/rule for other ANN and LLNF models.

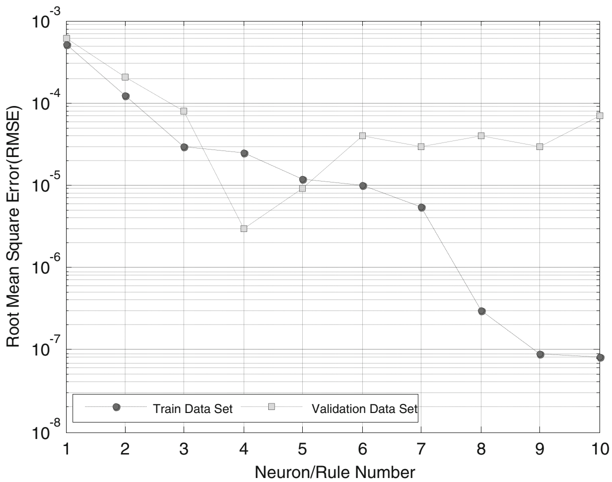

Fig. 6 A typical RMSE curve for estimating the number of neurons/rules

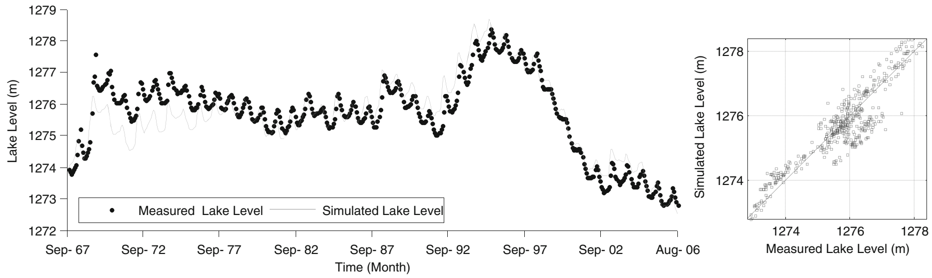

Fig. 7 Measured and simulated water level by WB model (*left*) and its scatter plot (*right*)

The structures of all nonlinear models which are investigated in this paper are shown in Table 2.

The findings simulated by water balance model and measured data are compared in Fig. 7 where the scatter diagram of it can be seen across from the lake level's graph.

Simulated lake level in Fig. 7 demonstrates remarkable results where 0.486 and 0.924 calculated for RMSE and correlation coefficient, respectively. Since all effective hydrological components appear in water balance equation, these results were respected. The existing offset between measured and simulated graphs can be justified by impact of groundwater. For instant, in the period of 1967 to 1977 the simulated graph lies under measured graph that are illustrated by thin gray line and black dot respectively. On the other hand, from year 1987 up to the end the simulated graph almost always lies above the measured one. As a result, undoubtedly there were some terms that *were not embedded* in the Eq. 1 in which cause these effects. However, the critical point is that there is no straightforward and accrued way to measure these un-modeled parameters such as groundwater. Figure 8 illustrates the measured and simulated values of the lake level obtained by the multiple regression method.

The obtained results in Fig. 7 could be seen again Fig. 8 with slight changes since both model (WBE and MRM) are similar in nature. As Fig. 8 indicates, the offset between measured/simulated graphs in the stated periods of time, took place here more clearly compared with water balance model that makes un-modeled components thesis more probable. In other works, probable contributors to the lake level changes (e.g., natural water exchange between the lake and its watershed, precipitation on the lake watershed, temperature, wind speed, groundwater changes, and etc.) which are ignored in Eq. 2 could cause these gaps between measured and simulated graphs. However, acquired correlation coefficient by MRM (i.e., 0.905) indicates

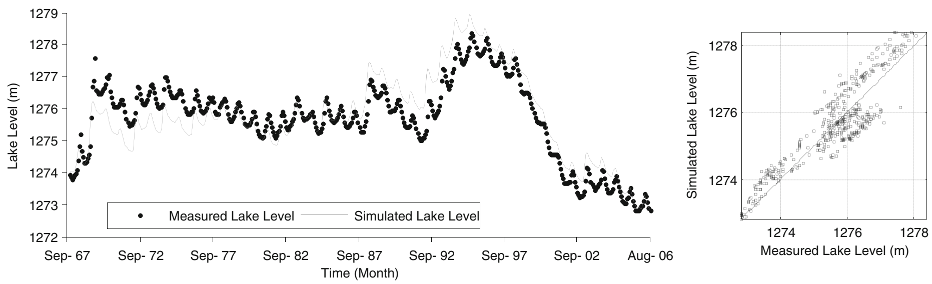

Fig. 8 Measured and simulated water level by MRM model (*left*) and its scatter plot (*right*)

Table 3 Accuracy results achieved basis on the proposed modeling ways in statistical point of view

Model	Training set			Validation set		
	RMSE(m)	Similarity(%)	R	RMSE(m)	Similarity(%)	R
Water balance	–	–	–	0.486	60.935	0.924
Multiple regression	–	–	–	0.565	54.563	0.905
ARX (1,1,0)	**0.436**	**57.766**	**0.709**	**0.312**	**83.054**	**0.989**
ARX (1,12,0)	0.263	56.386	0.902	0.532	71.092	0.986
Box-Jenkins (1,1,1,1,0)	0.628	59.420	0.803	0.352	79.871	0.982
Box-Jenkins (12,1,1,12,0)	0.581	55.841	0.681	0.394	78.590	0.979

variables (measured/simulated data sets) can be considered highly correlated. Overall, according to Figs. 7 and 8 and Table 3 WBE outperform MRM.

As other linear models more famous ARX and Box-Jenkins have been exploited for monthly prediction. According to Fig. 5, we decided to adjust output dynamic depth to one. Besides, assuming a linear process, also with respect to the optimality of the least-squares solution for modeling linear process, it is expected that the best fitness will be obtained by ARX method. Figure 9 represents the measured lake level and their predicted responses of ARX model. The scatter diagram for both training and validation sets are depicted above of associated district which are separated by a vertical line. The numerical results obtained by Box-Jenkins method are also depicted in last two rows of the Table 3 as well as the row distinguished in bold, indicate the best linear model.

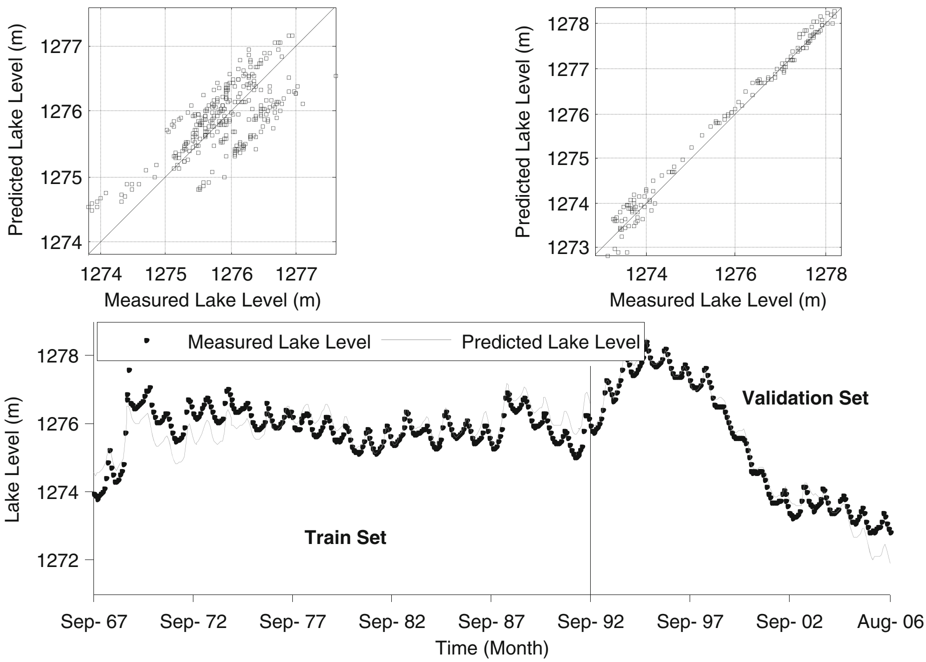

Fig. 9 Measured and predicted water level by ARX model (*bottom*) and its associated scatter plot for training and validation sets (*top*)

With a glance, one can verify that ARX has a tracking behavior, but as Fig. 9 shows it did not have the optimal fitness with measured data which is expected to be accrued in case of linear processes. Calculated equation by using ARX model can be seen in Eq. 18.

$$\widehat{H}(k) = 0.0008108 \times P(k-1) - 0.00109 \times E(k-1) + 1.857 \times \mathrm{e}^{-10} \times R_{in}(k-1) \quad (18)$$

Table 3 illustrates the values of RMSE, R and similarity for three linear studied methods. Note that integers written parenthesis for ARX denote n_a, n_b, and n_k where are number of A and B model parameters and input-output delay respectively (because of 1 month time sampling interval, delay consider zero i.e., n_k=0). Similarly for BJ model, integers in parenthesis specify n_b, n_c, n_d, and n_f which are orders of the B, C, D, and F polynomials respectively, and the last integer indicate input delay.

As a result, all obtained findings using recent mentioned three linear models, were somewhat able to satisfy three criteria. But all these three models assume that a given time series is generated form an underlying linear process. Thus, they may not always produce perfect results when are employed for modeling *hydrological* time series which are often *nonlinear*. According to these reasons, we decided to examine the capacity of nonlinear methods namely MLP and LLNF to estimate the lake level changes.

In order to infer how effective output dynamic is on the result, a static MLP neural network has been designed and tested. As indicated in Fig. 10, in training set the static neural network model-1 (SNM-1) was partially able to track the measured data. In contrast, in validation set the model output did not follow the reference time series at all.

It may be easily referred to inability of static neural networks in extrapolating data beyond the range those were used in training phase (i.e., period between Sep. 67 and Sep. 92). From the

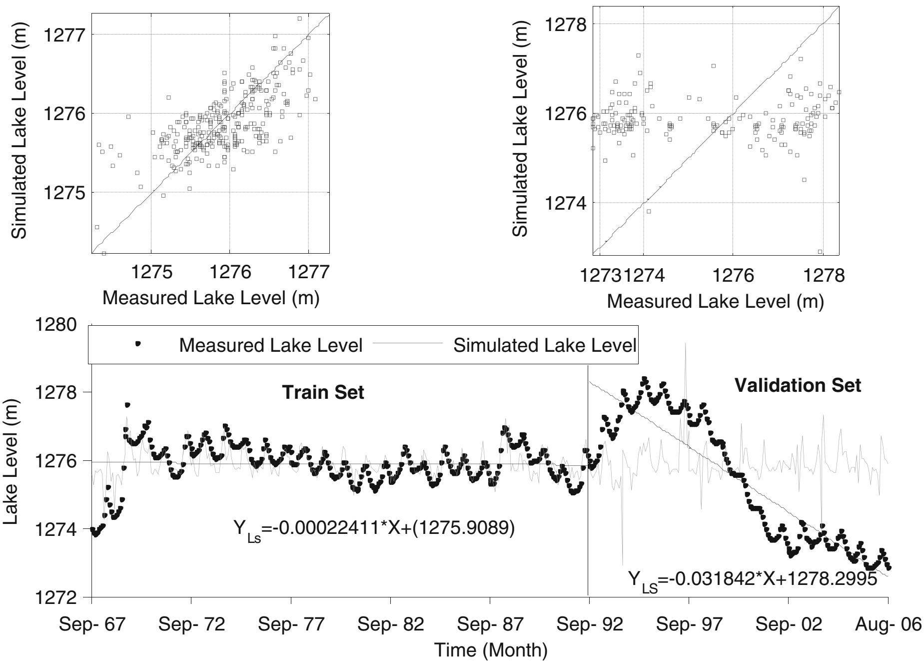

Fig. 10 Measured and simulated water level by SNM-1 model (*bottom*) and its associated scatter plot for training and validation sets (*top*)

measured lake level graphs, this can be literally seen because global climate changes the lake level regime has changed completely. In other words, it is difficult for an interpolator model to estimate values that are outside the range of data which were used for training.

In order to further explore, the last mentioned neural network was again trained and evaluated with different time periods. In this scenario the data from September 1967 to 1982 are considered to be training set; and data from September 1982 until September 1992 constitute the validation set (Fig. 11). In this case, despite the fact that the validation data have not been seen by the neural network, it could estimate measured data set several times better than SNM-1 because the whole two data sets are pottering around a particular mean value. In addition, the drawn regression line in Fig. 11 did not show a considerable variation since both slope and y-intercept remained almost stable. In contrast, the regression line in Fig. 10 indicates that after a slight fall, there was a dramatic decline. The same procedure was performed for two static LLNF models and the findings are depicted in the first four row of Table 4.

Therefore, as static intelligent methods showed, the lake level value at time (t) is *crucial* to achieve acceptable performance by a typical intelligent model at time (t+1) and further. Noteworthy, the correlation trail (Fig. 5) and both mathematical models namely WBE and MRM confirm this claim. Thus, in the rest of the paper output dynamic order considered to be one in all dynamic models. Similarly, input antecedents could affect the performance of the models. Besides, regarding Fig. 5 and some trial and errors the best dynamic depth for predicting the lake level could be (t−11). In spite of injecting dynamics into models would lead to the better results, at the same time it leads to an increase in both time and computational complexity of the models. In other words, according to system's dynamic, the model structure can vary in term of complexity. Therefore, it will be worthwhile to examine ability of low order models to achieve predefined accuracy. Figure 12 demonstrates performance of dynamic neural network model-1

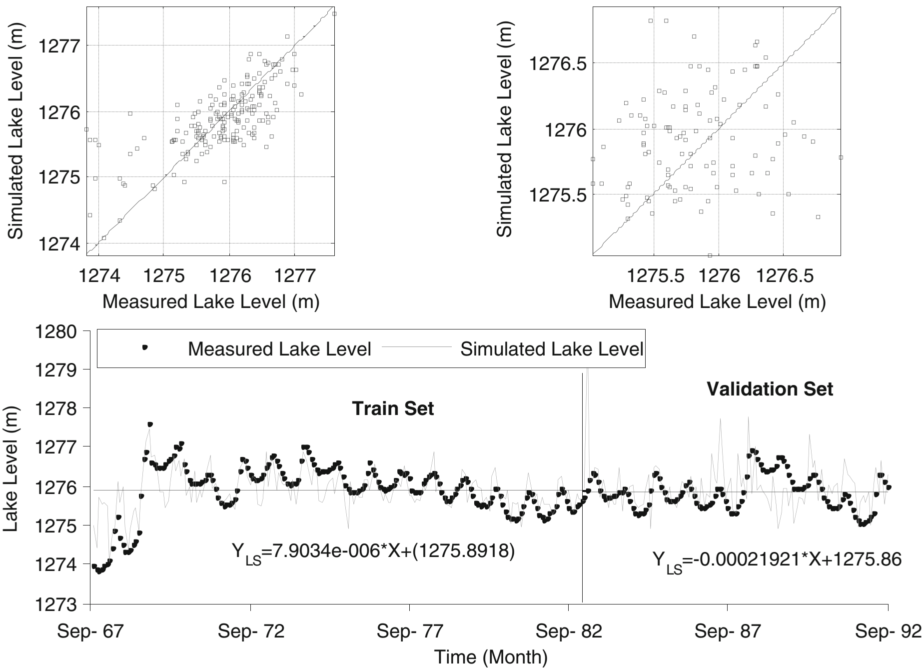

Fig. 11 Measured and simulated water level by SNM-2 model (*bottom*) and its associated scatter plot for training and validation sets (*top*)

Table 4 Performances calculated by nonlinear models for training and testing periods

Model	Training set			Validation set		
	RMSE(m)	Similarity(%)	R	RMSE(m)	Similarity(%)	R
SNM-1	0.463	23.300	0.642	1.952	6.005	0.642
SNM-2	0.472	31.494	0.729	0.701	61.998	0.196
SFM-1	1.396	10.147	0.188	2.055	13.619	0.229
SFM-2	0.571	17.104	0.565	0.442	12.054	0.414
DNM-1	0.080	86.749	0.991	0.340	81.555	0.995
DNM-2	0.056	90.700	0.996	0.159	91.354	0.996
DNM-3	0.053	91.276	0.996	0.145	92.151	0.998
DFM-1	0.042	92.994	0.998	0.136	92.602	0.997
DFM-2	0.024	96.069	0.999	0.117	93.622	0.998
DFM-3	**0.017**	**97.215**	**1.000**	**0.109**	**94.081**	**0.998**

(DNM-1) which benefits from 16 neurons in its hidden layer and the input order adjusted to be 2 as can be seen in Table 2.

According to Fig. 12 it can be clearly seen in both measured/predicted graph and scatter plot, predicted data deviate from reference graph (i.e., measured values) specially in upper (1276.5 m to 1278.3 m) and lower (1272.8 m to 1274.5 m) bands. Thus, at the next effort, input order has been increased up to 4 (DNM-2) which consequently leads to an increase in the number of inputs

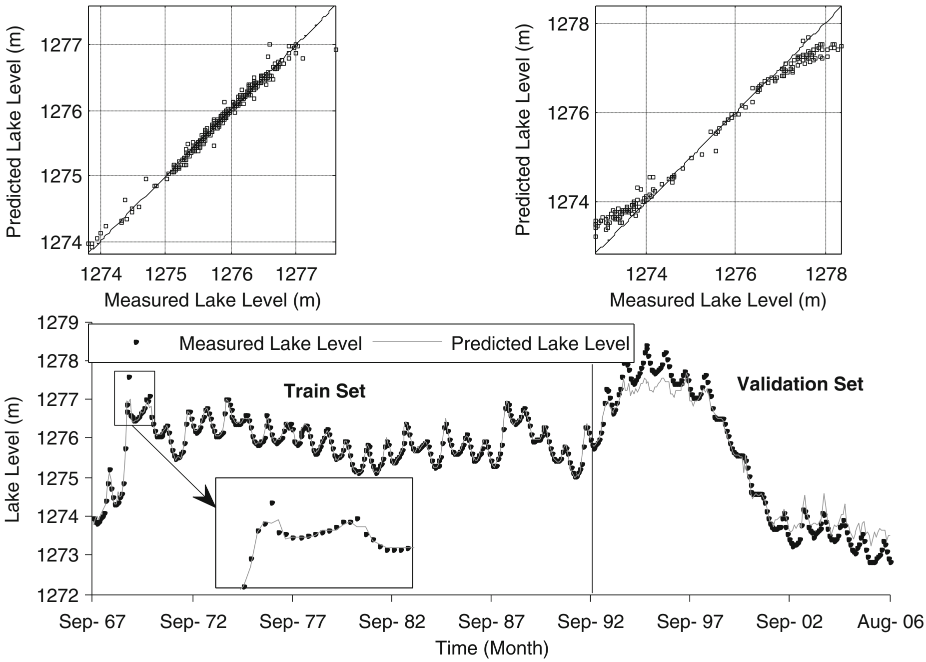

Fig. 12 Measured and predicted water level by DNM-1 model (*bottom*) and its associated scatter plot for training and validation sets (*top*)

and in turn, the number of neurons that is unavoidable to handle these newly added inputs. In order to avoiding duplication, function of DNM-2 is just reported in numerical form (Table 4).

As Fig. 13 demonstrates, the deviation problem has been largely resolved by DNM-3 which utilized 29 neurons within its hidden layer. At this point, employed neural network can tightly follow the reference time series through whole validation strict as well as in controversial discussed periods (upper and lower bands). This procedure verifies that by adding the input dynamics the findings gradually evolve. Also, last obtained results (DNM-3) are worthy enough to convince us to interrupt increasing input dynamics as continuing this procedure leads to high-cost networks. Moreover, the number of data samples in this research is limited while abundant number of neurons in neural networks usually leads to over parameterization problem. In other words in these situations, the number of data samples would not be adequate for training such big-size networks.

One of the main advantages of LLNFs compare to NNs could appear in this situation (abundant number of inputs and thus high-cost neural networks), since LLNFs usually need less number of rules to conducting training process with the same number of inputs (see Table 2). This characteristic makes neuro-fuzzy methods powerful means to modeling systems with higher degree of dynamic. The predicted result by the dynamic fuzzy model-1 (DFM-1) and the measured the lake water-levels are compared in Fig. 14. Indeed, it was found that a low-cost neuro-fuzzy mode (DFM-1) performed almost the same as our most complicated neural network (DNM-3).

Since LLNF has admissible computational cost, it was tested for two other deeper dynamics. As it was expected, by increasing the order of models, they perform more accurately. The statistics results achieved by DFM-2 can be followed in Table 4. Apart from numerical results reported in Table 4, the findings produced by DFM-3 can also be seen in Fig. 15.

Table 4 contains statistic results calculated by all nonlinear models which were examined in this paper. Two first rows indicate the ability of two static MLP neural networks that were

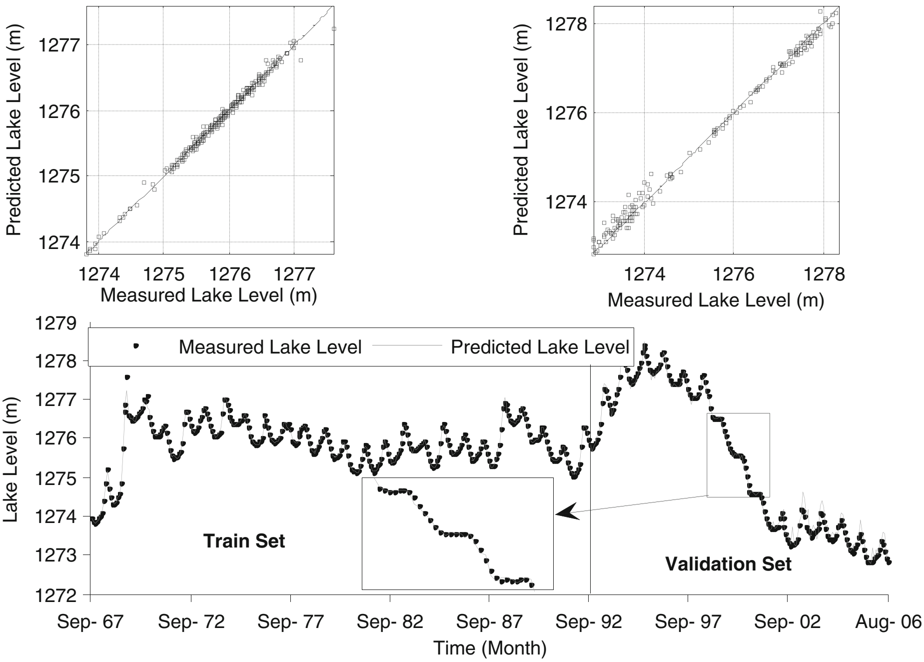

Fig. 13 Measured and predicted water level by DNM-3 model (*bottom*) and its associated scatter plot for training and validation sets (*top*)

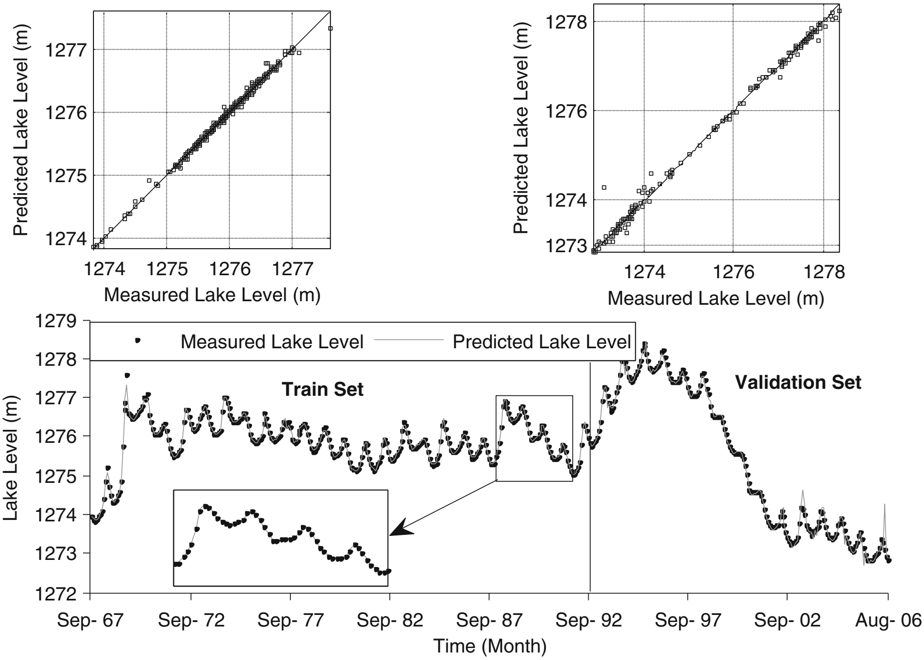

Fig. 14 Measured and predicted water level by DFM-1 model (*bottom*) and its associated scatter plot for training and validation sets (*top*)

different in term of their training and testing data time periods. The next two rows indicate the capacity of two static LLNF models in the same training and testing data time periods. The first four rows revealed the inability of static intelligent methods in predicting the lake level changes behavior since none of the assessment criteria indicates promising results. The efficiency of dynamic NN can be seen in the middle of the Table 4 in which a sudden improvement took place in all triple measurement tools. This dramatic growth reflecting the necessity of existing input/output antecedents in order to achieve a trustable model.

Finally, the results achieved by dynamic LLNF models, printed in last three rows of the Table 4. According to Tables 3 and 4, the LLNF methods surpass five other methods. In addition, among three studied LLNFs models the last one (the row in bold i.e., DFM-3) showed the best performance.

As mentioned in the section 1, Karimi et al. (2012) exploited three approaches, namely, Auto Regressive Moving Average (ARMA), ANFIS, and GEP to encounter the problem of Urmia Lake water-level fluctuations.

There were some specific assumptions in their work that must be taken into consideration before making a comparison between the results. Firstly, their data were collected over three decades (whereas our data were collected over four decades). Secondly, their used data were recorded daily, while ours were recorded monthly. Consequently, the numbers of used data for training and validation phases are different, and hence the chosen time periods for training and validation phases are different as well. Also their adopted methodologies are different from those investigated in our study.

However, perhaps the major distinction between our works is in the way of utilizing models. In other words, they used a time series model (ARMA) for predicting the lake levels and thus they could just apply previous recorded lake levels to the model without considering the three effective physical inputs (E, Rin, and P) for modeling. Besides, they applied the same inputs (only the lake level's past records) to ANFIS and GEP which

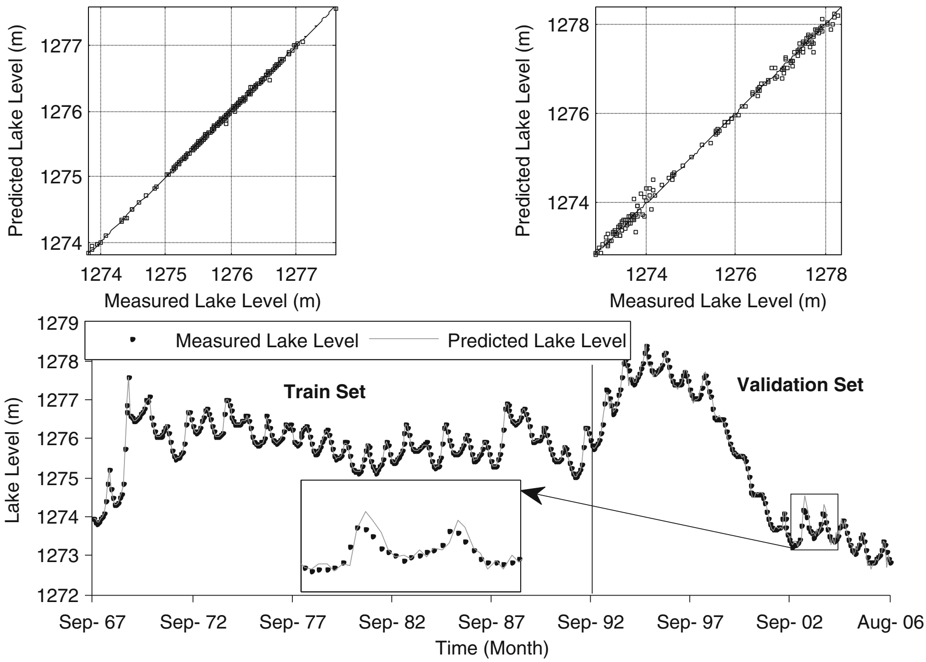

Fig. 15 Measured and predicted water level by DFM-3 model (*bottom*) and its associated scatter plot for training and validation sets (*top*)

are mostly known for their ability in modeling nonlinear dynamic systems, and not for time-series models.

Also, for validating their model, they divided the data into two periods of testing and validation. As the results of Test period showed, at 30-day ahead horizon, the ANFIS was claimed to be the best model with R^2=0.884 and *RMSE*=0.141 m. However, the outcomes of the Validation period were widely different from the Test period since ANFIS was outperformed by the other two methodologies with a wide gap: since the GEP model was reported as the best with *RMSE*=0.15 and the ARMA model obtained *RMSE*=0.17, while the ANFIS model yielded *RMSE*=0.81 respectively.

The abovementioned numerical results can be compared with the findings of our paper with respect to the explained distinctions. As their results showed, we presented more accurate models since they benefit from effective physical inputs, which are inflows, precipitation, and evaporation. In other words, using dynamic models (instead of time series) is highly recommended in cases where real-world effective inputs are well understood. Furthermore, utilizing time series is justifiable when the variables influencing the system are unknown, are not measureable, or are large in number such that it not feasible to incorporate them in modeling (Nelles 2001). Obviously, because no external input applied into their model, achieving relatively undistinguished models was expected (compare the results mentioned in the previous paragraph to those reported in the last row of Table 4). Moreover, time series allows only short-term predictions (usually one-step i.e., 1 month ahead) with sufficient accuracy (Nelles 2001).

Additionally, time series models are able to predict future (usually one step ahead). However, as mentioned in section 1, one can consider two other goals for modeling systems, including control and optimization purposes. In control application, only those models can be embedded into the control loop that are able to receive control commands. For instance, there are 29 reservoir dams with 1,712 million cubic meters (MCM) regulated volume (Iranian Ministry of Energy 2007) constructed on upstream rivers in the Urmia Lake basin, and thus releases from these reservoirs could be considered as control command and dynamic model of the lake water-level may be considered as a control object (plant) with slight modifications.

6 Conclusion

In this study various models were developed for estimating the Urmia Lake water-surface fluctuations. From two distinct points of views these models were specialized into linear-nonlinear and simulator-predictor models. Two hydraulically conventional methods namely WBE and MRM models, as was initially expected, had noticeable results in term of accuracy. Later it appeared that because of the rigid nature of such models they were not able to handle un-modeled parameters such as groundwater, or uncertainties such as those imposed into the model during constructing evaporation data from raw ones. Consequently, ARX model which used LS method for training procedure showed more flexibility in involving stated uncertainties. Although all linear models generated promising findings, unsatisfactory accuracy cause investigating application of nonlinear intelligent models including MLP neural network and LLNF models. All results obtained by neural networks and neuro-fuzzy networks demonstrate that the MLP and LLNF can be applied successfully to plant the predictor models which provide accurate and trustful monthly lake water-level prediction. In other words, both MLP and LLNF had great ability of learning input–output time series which is generated form an underlying hydro meteorological process moreover intelligent methods generally outperform WBE, MRM and ARX. On the other hand, in case of limited data access, ARX model become more preferred method since nonlinear models need more data for tuning their parameters.

Apart from accuracy point of view, LLNF (with LOLIMOT algorithm for training) has some special advantages over other conventional intelligent models such as MLP and ANFIS. By considering equal number of input, LLNF models showed better results in water level predictions than the MLP models. In other words, when a more flexible models are needed to handle systems with more dynamic order (for instance, the case under study), both time and memory required for a typical NN increased substantially, while the LLNF copes with situation more properly. Therefore, the authors would like to examine ability of proposed mode in this paper with previous offered models in same situation. In addition, since almost all hydrological phenomena are stochastic or influence stochastic components investigating seasonal performance of the models in term of both accuracy and reliability (Uncertainty analysis) could be subject of future work.

Acknowledgments First author would like to thank Ms. Mina Ghanavati for her help and feedback. The authors express their thanks to Dr. Mahdi Aliyari Shooredeli for his scientific recommendations. We also sincerely appreciate Dr. Ellips Masehian for his valuable guidance.

References

Afshar A, Mariño MA, Saadatpour M, Afshar A (2011) Fuzzy TOPSIS multi-criteria decision analysis applied to Karun reservoirs system. Water Resour Manag 25(2):545–563

Altunkaynak A (2007) Forecasting surface water level fluctuations of Lake Van by artificial neural networks. Water Resour Manag 21(2):399–408

Astushi U, Masaki T, Yoshio I (2004) Lake-level changes during the past 100,000 years at Lake Baikal, southern Siberia. Quat Res 62(2):214–222

Aytek A, Alp M (2008) An application of artificial intelligence for rainfall runoff modeling. J Earth Syst Sci 117(2):145–155
Babovic V, Keijzer M (2002) Rainfall runoff modelling based on genetic programming. Nord Hydrol 33(5):331–346
Banadaki DH, Nozari HA, Kakahaji H (2011) Nonlinear simulator model identification of a walking beam furnace using recurrent local linear neuro-fuzzy network. Int J Control Autom 4(4):123–134
Bengtsson L, Malm J (1997) Using rainfall–runoff modeling to interpret lake level data. J Paleolimnol 18(3):235–248
Billings SA, Voon WSF (1986) Correlation based model validity tests for non-linear models. Int J Control 44(1):235–244
Billings SA, Voon WSF (1987) Piecewise linear identification of non-linear systems. Int J control 46(1):215–235
Chatterjee S, Hadi AS (1986) Influential observations, high leverage points, and outliers in linear regression. Stat Sci 1(3):379–416
Cybenko G (1989) Approximation by superposition of a sigmoidal function. Math Control Signals Syst 2(4):303–314
Ghorbani MA, Khatibi R, Aytek A, Makarynskyy O, Shiri J (2010) Sea water level forecasting using genetic programming and comparing the performance with Artificial Neural Networks. Comput Geosci 36(5):620–627
Güldal V, Tongal H (2010) Comparison of recurrent neural network, adaptive neuro-fuzzy inference system and stochastic models in Eğirdir lake level forecasting. Water Resour Manag 24(1):105–128
Hagan MT, Menhaj MB (1994) Training feedforward networks with the Marquardt algorithm. IEEE Trans Neural Netw 5(6):989–993
Hassanzadeh E, Zarghami M, Hassanzadeh Y (2012) Determining the main factors in declining the Urmia lake level by using system dynamics modeling. Water Resour Manag 26(1):129–145
Haykin S (1999) Neural networks: a comprehensive foundation, 2nd Ed. Prentice Hall PTR Upper Saddle River, NJ, USA
Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
Iranian Ministry of Energy (2007) Water allocation for the development projects in Urmia Lake basin, technical report of Water and Wastewater Planning Office
Iranian Society of Consulting Engineers (2011) Environmental crisis of Lake Urmia. J Consult Eng 53:30–52
Jones RN, McMahon TA, Bowler JM (2001) Modelling historical lake levels and recent climate change at three closed lakes, Western Victoria, Australia (c. 1840–1990). J Hydrol 246(1–4):159–180
Kadıoğlu M, Sen Z, Batur F (1999) Cumulative departures model for lake-water fluctuations. J Hydrol Eng 4(3):245–250
Karimi S, Shiri J, Kisi O, Makarynskyy O (2012) Forecasting water level fluctuations of Urmieh lake using gene expression programming and adaptive neuro-fuzzy inference system. Int J Ocean Clim Syst 3(2):109–126
Karimi S, Kisi O, Shiri J, Makarynskyy O (2013) Neuro-fuzzy and neural network techniques for forecasting sea level in Darwin Harbor, Australia. Comput Geosci 52:50–59
Kavehkar SH, Ghorbani MA, Khokhlov V, Ashrafzadeh A, Darbandi S (2011) Exploiting two intelligent models to predict water level: a field study of Urmia lake, Iran. Int J Civ Environ Eng 3(3):162–166
Khu ST, Liong SY, Babovic V, Madsen H, Muttil N (2001) Genetic programming and its application in real-time runoff forecasting. J Am Water Resour Assoc 37(2):439–451
Kisi O, Shiri J, Nikoofar B (2012) Forecasting daily lake levels using artificial intelligence approaches. Comput Geosci 41:169–180
Liong SY, Gautam TR, Khu ST, Babovic V, Keijzer M, Muttil N (2002) Genetic programming: a new paradigm in rainfall runoff modeling. J Am Water Resour Assoc 38(3):705–718
Ljung L (1987) System identification theory for the user. Prentice Hall, Upper Saddle River
Marce R, Comerma M, Garcia JC, Armengol J (2004) A neuro-fuzzy modeling tool to estimate fluvial nutrient loads in watersheds under time-varying human impact. Limnol Oceanogr Methods 2:342–355
Muttil N, Liong SY (2001) Improving runoff forecasting by input variable selection in GP. In Proceedings of the World Water and Environmental Resources Congress, pp. 1−7 Orlando, Florida, USA
Nelles O (2001) Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Springer, Berlin
Nelles O, Isermann R (1996) Basis function networks for interpolation of local linear models. In Proceedings of the 35th IEEE Conference on Decision and Control, pp. 470–475 Kobe, Japan
Nozari HA, Banadaki HD, Mokhtare M, Vahed SH (2012a) Intelligent non-linear modelling of an industrial winding process using recurrent local linear neuro-fuzzy networks. J Zhejiang Univ Sci C 13(6):403–412
Nozari HA, Simani S, Shoorehdeli MA, Banadaki DH (2012b) Model-based robust fault detection and isolation of an industrial gas turbine prototype using soft computing techniques. Neurocomputing 91(15):29–47

Ostadrahimi L, Mariño MA, Afshar A (2012) Multi-reservoir operation rules: multi-swarm PSO-based optimization approach. Water Resour Manag 26(2):407–427

Roshan G, Ghanghermeh AA, Nasrabadi T, Meimandi JB (2013) Effect of global warming on intensity and frequency curves of precipitation, case study of Northwestern Iran. Water Resour Manag 27(5):1563–1579

Sang YF (2013) Improved wavelet modeling framework for hydrologic time series forecasting. Water Resour Manag 27(8):2807–2821

Shiri J, Makarynskyy O, Kisi O, Dierickx W, Fard A (2011) Prediction of short-term operational water levels using an adaptive neuro-fuzzy inference system. J Waterw Port Coast Ocean Eng 137(6):344–354

Tabari H, Nikbakht J, Talaee PH (2013) Hydrological drought assessment in Northwestern Iran based on Streamflow Drought Index (SDI). Water Resour Manag 27(1):137–151

Talebizadeh M, Moridnejad A (2011) Uncertainty analysis for the forecast of lake level fluctuations using ensembles of ANN and ANFIS models. Expert Syst Appl 38(4):4126–4135

Vaziri M (1997) Predicting Caspian Sea surface water level by ANN and ARIMA models. Waterw Port Coast Ocean Eng 123(4):158–162

Water Resources Management Company of Iran (2011) Status of water supply installation of Urmia Lake's watershed, technical report (in Persian)