

## Comparison of Box-Jenkins time series and ANN in predicting total dissolved solid at the Zāyandé-Rūd River, Iran

G. Asadollahfardi, H. Zangooi, M. Asadi, M. Tayebi Jebeli, M. Meshkat-Dini and N. Roohani

### ABSTRACT

We applied the Box-Jenkins time series model and artificial neural network (ANN) in the framework of a multilayer perceptron (MLP) to predict the total dissolved solids (TDS) in the Zāyandé-Rūd River, Esfahan province, Iran. The MLP inputs were total hardness (TH), bicarbonate ( $\text{HCO}_3^-$ ), sulfate ( $\text{SO}_4^{2-}$ ), chloride ( $\text{Cl}^-$ ), Sodium ( $\text{Na}^+$ ), and Calcium ( $\text{Ca}^{2+}$ ), which were monitored over 9 years by the Esfahan Water Authority. The Autoregressive Integrate Moving Average (ARIMA) (2, 0, 3) (2, 0, 2) time series model with the lowest Akaike factor was selected. The coefficient of determination ( $R^2$ ) and index of agreement ( $IA$ ) between the measured and predicted data of the ARIMA (2, 0, 3) (2, 0, 2) time series model were 0.78 and 0.73, respectively. Using Tansig transfer functions, the Levenberg-Marquardt algorithm for training and an MLP neural network with 10 neurons in a hidden layer were developed.  $R^2$  and  $IA$  between the measured and predicted data were 0.94 and 0.91, respectively. Consequently, the results of the MLP were more reliable than the Box-Jenkins time series to predict TDS in the river.

**Key words** | ANN, Box-Jenkins, water quality, Zāyandé-Rūd River

G. Asadollahfardi (corresponding author)  
H. Zangooi  
M. Tayebi Jebeli  
M. Meshkat-Dini  
N. Roohani  
Department of Civil Engineering, Faculty of  
Technical and Engineering,  
Kharazmi University,  
43 Mofateh Avenue, Tehran,  
Iran  
E-mail: fardi@khu.ac.ir

M. Asadi  
Department of Mechanical Engineering,  
University of Saskatchewan,  
Saskatoon,  
Canada

### INTRODUCTION

Sufficient forecasting of future conditions in water bodies is of high interest to a great variety of stakeholders including government departments, agencies, policymakers, health and medical bodies. Water quality is a basic subject in water resource management. Water salinity is one of the main parameters in water quality, and predicting it is vital for water quality management for both drinking and irrigation uses. The salinity of the surface waters, as an index of the dissolved constituents and ions in the water, is a significant issue in water quality management for irrigation and drinking purposes. Total dissolved ions (TDI), conductivity and total dissolved solids (TDS) can describe salinity (McNeil & Cox 2000).

Water salinity depends on minerals in the water or soil characteristics of the catchment area, of which some parts are the particulate matter and other parts are in solution

(Asadollahfardi *et al.* 2012a, 2012b). Salinity, which characterizes the majority of the dissolved constituents in water, can be determined by a number of methods. One technique to indicate TDI is the total number of ions in solution described as the sum of the main ions in water expressed in mg/l. These contain the cations sodium ion ( $\text{Na}^+$ ), potassium ion ( $\text{K}^+$ ), calcium ion ( $\text{Ca}^{+2}$ ), and magnesium ion ( $\text{Mg}^{+2}$ ) and anions chloride ion ( $\text{Cl}^-$ ) and bicarbonate ion ( $\text{HCO}_3^-$ ) with sulphate ion ( $\text{SO}_4^{2-}$ ) in most surface water and groundwater. Conductivity is another parameter which can provide an easy measurement of the salinity. TDS or 'filterable residue,' the concentration of dissolved substance in water, is another parameter which may be considered. This includes mineral and organic matter (Asadollahfardi *et al.* 2012a, 2012b).

Mathematical and statistical approaches have been employed frequently to predict water quality (Faruk 2010).

Several methods exist for predicting TDS in river water, such as the Box-Jenkins time series, Bayesian time series and artificial neural network (ANN). Some of the applications of the first and third methods are as follows: Jayawardena & Lai (1989) analyzed water quality of the Pearl River using time series. Sun *et al.* (2001) applied Box-Jenkins time series, including Autoregressive Integrate Moving Average (ARIMA) time series, to anticipate the water quality of the Apalachicola Gulf. They recognized water level changes in tide situations, water quality of entering streams, regional rainfall, wind velocity, and wastewater discharge into the water body as the determinative elements in water quality of the Gulf. Asadollahfardi (2002) deployed different Box-Jenkins time series to study surface water quality in Tehran, Iran. Kurunc *et al.* (2005) used ARIMA and Thomas-Fiering performed prediction to monitor water quality and flow rate of the Yeşilirmak River, Turkey. The root mean square error and mean absolute error indicated that the Thomas-Fiering model predictions were more reliable than ARIMA. Asadollahfardi *et al.* (2012a, 2012b) studied the water quality upstream and downstream of the Latian Dam (located in Iran) using the Box-Jenkins time series. Abudu *et al.* (2011) applied ARIMA, transfer function-noise (TFN), and ANN approaches to anticipate the monthly TDS of the Rio Grande in El Paso, Texas. Ranjbar & Khaledian (2014) used the ARIMA method to predict water quality parameters of the Sefid-Rūd River. They reported satisfactory outcomes. Arya & Zhang (2015) applied ARIMA models to predict dissolved oxygen and temperature in four water quality monitoring stations at Stillaguamish River Washington, US. They indicated the aptness of the Box-Jenkins time series in the forecasting. Salmani & Salmani Jajaei (2016) studied the TDS of Karoun River in the southeast of Iran using ARIMA; they also used a transfer function model to formulate TDS as a function of water flow volume.

Maier & Dandy (1996) applied an ANN to estimate the concentration of salinity in the River Murray, South Australia. They reported reasonable results. Zhang *et al.* (1997) used an ANN to forecast river raw water quality in the Northern Saskatchewan. Huang *et al.* (2002) assessed the application of an ANN in the modeling of salinity variation at the Apalachicola River, Florida, USA.

Misaghi & Mohammadi (2003) used an ANN to analyze the water quality of Zāyandé-Rūd River. They employed a

general regression neural network (GRNN) to model biochemical oxygen demand (BOD) and dissolved oxygen (DO). Kanani *et al.* (2008) reached agreeable results in applying the multilayer perceptron (MLP) neural network and input delay neural network (IDNN) to predict TDS in Achechay River basin, Iran. Asadollahfardi *et al.* (2012a, 2012b) used the MLP and recurrent neural network (RNN) to predict TDS one month in advance in the Talkheh Rud River. Nemati *et al.* (2014) employed an ANN to predict TDS in the Simineh River, Iran. Babu & Reddy (2014) developed a hybrid ARIMA-ANN to forecast time series data. They reported good accuracy of the model. Adhikari (2015) showed that combining time series forecast from several models would result in better outcomes. Gairaa *et al.* (2016) applied Box-Jenkins and ANN models to estimate the daily global solar radiation. The acquired results were satisfactory. Taneja *et al.* (2016) studied time series analysis of aerosol optical depth over New Delhi using the Box-Jenkins ARIMA modeling approach. The data were gathered from 2004 to 2014. The results showed that simple modeling suggested reliable prediction for aerosol optical depth. Asadollahfardi *et al.* (2016) applied a time delay neural network (TDNN) and radial basis function (RBF) to forecast TDS in Zāyandé-Rūd River. Their results of sensitivity illustrated that  $\text{Ca}^{2+}$  and  $\text{SO}_4^{2-}$  parameters had the highest effect on the TDS prediction.

Asadollahfardi *et al.* (2017) applied the MLP and RBF neural network for two stations to predict TDS in Amirkabir Dam in Karaj, Iran. They concluded that RBF performance was better than that of the MLP neural network.

Montaseri *et al.* (2018) applied different types of ANN including two types of adaptive-neuro fuzzy inference system (ANFIS) including ANFIS with grid partition, ANFIS with subtractive clustering, gene expression programming (GEP), wavelet-ANN, wavelet ANFIS and wavelet GEP to forecast TDS of four rivers in different areas of Iran, including Nazlu Chay River, Tajan River, Zāyandé-Rūd River and Helleh River. They concluded that the hybrid wavelet-AI (artificial intelligence) performed better than the AI model in all the rivers.

## Study area

The Zāyandé-Rūd originates in the Zard-Kuh subrange of the Zagros Mountains and reaches to the Gavkhouni

Swamp. The geographic coordinates are N 31 30 30.32 and E 49 30 52.49 (Figure 1, The study area). Average annual rainfall varies between 1,600 mm at the height of Zardkooh Mountain to less than 40 mm in the plain. Average annual temperature is 3.5 °C in the northwestern highlands and 21.5 °C in the eastern areas of the central region of Iran. Sedimentary and metamorphic rocks of the Jurassic and new alluvium Quaternary are the most abundant rock of the riverbed, which are the major sources of fine-grained particles along the river. In general, fine-grained particles are considered themselves a cause for absorption and accumulation of elements of a toxic nature in stream sediments due to the high adsorption (Hossieni Abri 2000; Moienian 2008).

This paper employed Box-Jenkins time series models and MLP neural network RBF to predict TDS in Zāyandé-

Rūd River at Moisan monitoring station. The data were monitored by Isfahan Province Regional Water authority, which collected from 2001 to 2010.

## MATERIALS AND METHODS

### Box-Jenkins methodology

Box & Jenkins (1976) developed a new methodology, which basically breaks down stationary time series data into their components. In the stationary time series, the statistical characteristics of a time series (such as mean and variance) remain constant over time. Development of the Box-Jenkins method is based on the following steps (Sun & Koch 2001):

- Check the data for normality.

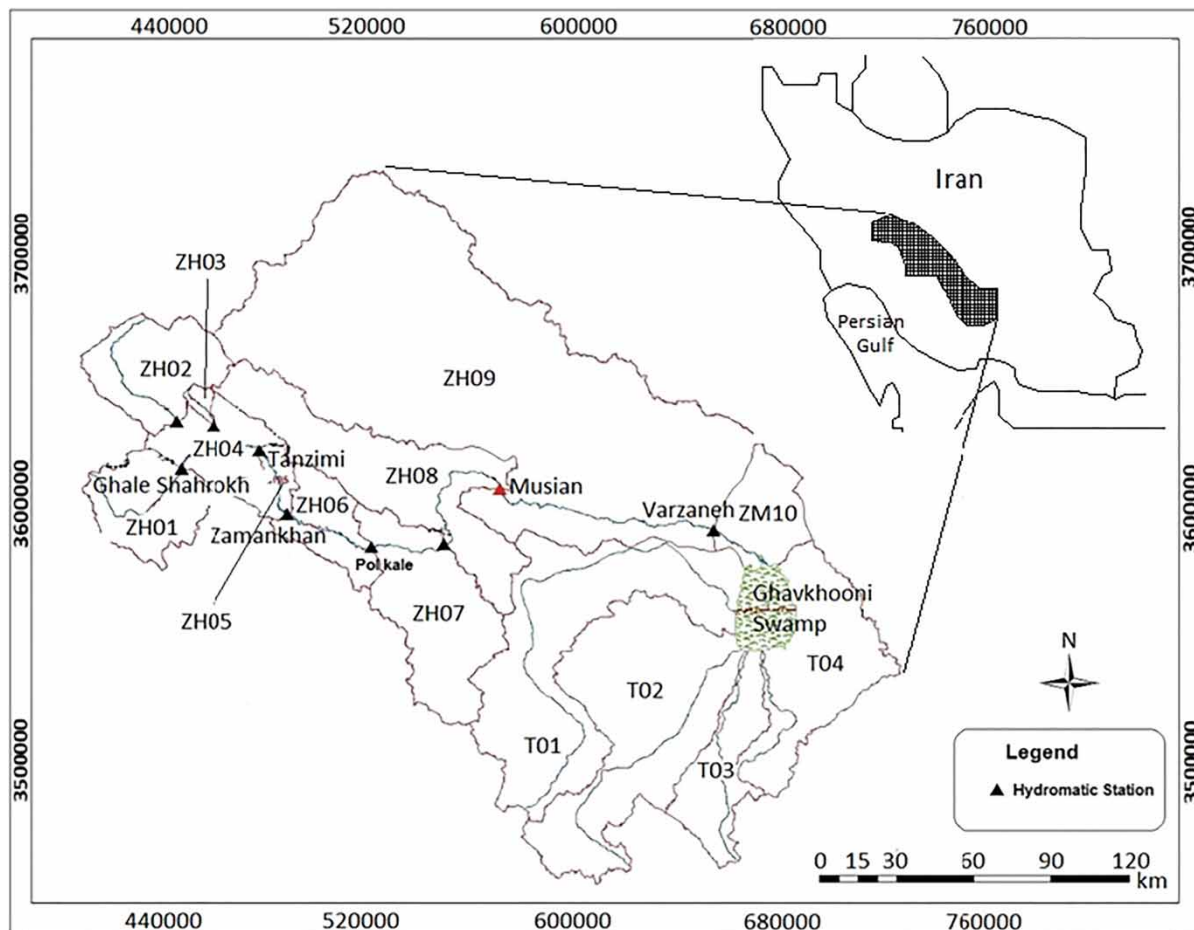


Figure 1 | The study area.

- Identification, including plot of the transformed series: autocorrelation function (ACF) and partial autocorrelation function (PACF).
- Estimation, which consists of maximum likelihood estimates (MLE) for the model parameters (Ansley algorithm)
- Diagnostic checks which consist of overfitting and check of residuals (modified Portmanteau test).
- Model structure chosen criteria which include Akaike information criterion (AIC) and Bayesian information criterion (BIC).

Equation (1) illustrates the general non-seasonal autoregressive moving average (ARMA) model of order  $(p, q)$ .

$$Z_t = \delta + \varphi_1 Z_{t-1} + \dots + \varphi_p Z_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (1)$$

where,  $\varphi_p(B)$  and  $\theta_q(B)$  are the autoregressive and moving average operators, respectively (Equations (2) and (3));

$$\varphi_p(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p \quad (2)$$

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q \quad (3)$$

where  $B$  is the backward shift operator and means  $B_k Z_t = Z_{t-k}$ .

If a series of  $Z_t, Z_{t-1}, \dots, Z_n$  is non-stationary (the mean and variance of the series is varying with time), Equation (4) indicates the relation between  $\varphi_p(B)$  and  $\theta_q(B)$ . In other words, Equation (4) presents the autoregressive integrated moving average  $ARIMA(p, d, q)$  model with integers  $p, d, q$ , indicating the order of the model.

$$\varphi_p(B) \nabla_d Z_t = \theta_q(B) a_t \quad (4)$$

The perfect description of the Box-Jenkins method can be found in Box & Jenkins (1976).

To determine types of time series, the ACF could be beneficial. The ACF of a random process is the correlation between amounts of the process at different times, as a function of the two times or the time lag. Equations (5) and (6) represent the ACF with  $k$  lag and the PACF.

$$\frac{\sum_{i=1}^{n-k} [z(i) - \bar{z}][z(i+k) - \bar{z}]}{\sum_{i=1}^n [z(i) - \bar{z}]^2} \quad (5)$$

$$\varphi(k) = \frac{\rho(k) - \sum_{i=1}^{k-1} \varphi(i)[k-1-i]\rho(k-1-i)}{1 - \sum_{i=1}^{k-1} \varphi(i)[k-1-i]\rho(i)} \quad (6)$$

where,  $k$  is the ACF value with  $k$  time lag;  $z(i)$  and  $z(i+k)$  are variable amounts of time series data in the  $i$  step and  $i+k$  step with  $k$  time lag;  $\bar{z}$  is the average value of the variables; and  $\varphi(k)$  is the PACF value with  $k$  time lag.

The ANN is a data processing system, which behaves similarly to the human neurological system. The ANN contains three unique components, including weight ( $w$ ), bias ( $b$ ) and transfer function ( $F$ ). Based on the scalar inputs ( $p$ ), ANN results in the scalar outputs ( $a$ ) (Figure 2 and Equation (7)).

$$a = f(n) = f(wp + b) \quad (7)$$

The MLP is a feed-forward network based on perceptron-type neurons, which are arranged in layers. The MLP has an input layer, one or more hidden layers and an output layer (Figure 3). In the MLP, each neuron in each layer of the network is connected to every other neuron in the adjacent forwarding layer. In the other words, the MLP is a fully connected network. The  $k$ -th output of a single hidden layer MLP can be written as follows (Menhaj 1998):

$$y_k = g_{out} \left( \sum_{j=0}^{N_2} w_{kj}^{(2)} g_{in} \left( \sum_{i=0}^{N_1} w_{ji}^{(1)} X_i \right) \right) \quad (8)$$

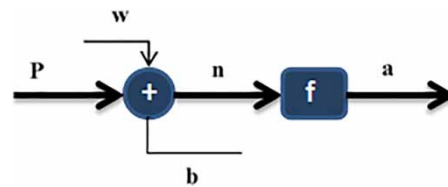


Figure 2 | Schematic of an artificial neural network.

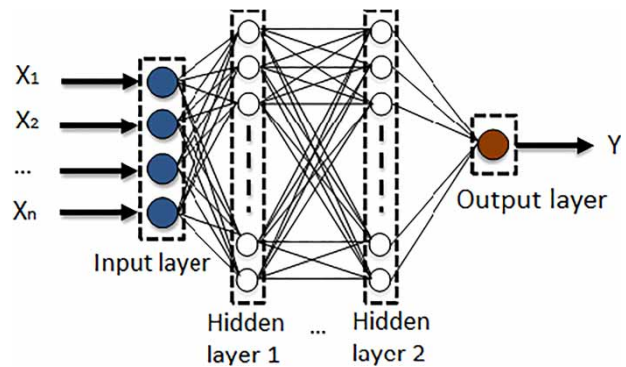


Figure 3 | An MLP with a hidden layer.

where,  $y_k$  represents the  $k$ -th output,  $w$  is the weights and biases,  $g_{out}$  is the output layer transfer function,  $g_{in}$  is the input layer transfer function, and  $i$  is the  $i$ -th layer.

### Determination of the network architecture

An appropriate number of hidden layers in an ANN is essential for best presentation of the network. In insufficient hidden layers, the mapping might not correctly be assessed. Having excessive hidden layers raises the network intricacy and does not inevitably lead to a rise in the precision of the network. Studies represented that a feed forward neural network with a hidden layer of sigmoid tangent and a linear output layer would be able to estimate each obscured function (Figure 4; Cybenko 1989; Hornik 1991, 1993; Leshno et al. 1993). Consequently, the network would be simpler (Hornik et al. 1989). The transfer function of the hidden layer in the MLP was a sigmoid tangent. The function of the output layer was a linear function. The number of neurons in the hidden layer was obtained by applying a trial and error method.

The slope of transfer function tangent sigmoid varies from  $-1$  to  $1$  (Figure 4). Therefore, changes in the scale of input data and measured data might be inevitable (Equation (9)). This equation is a generalized feature scaling method (Espejo 2004).

$$A_s(\text{scaled}) = \left[ \frac{A_i - A}{B - A} \right] \times 2 - 1 \quad (9)$$

where  $A_s$  and  $A_i$  are the scaled and measured (observed) values of input at time  $t$ , respectively.  $A$  and  $B$  are the lowest and highest values of the series of inputs. MATLAB software (R2012b) was applied to operate the model. In an MLP neural network, we normalized the measured input data and

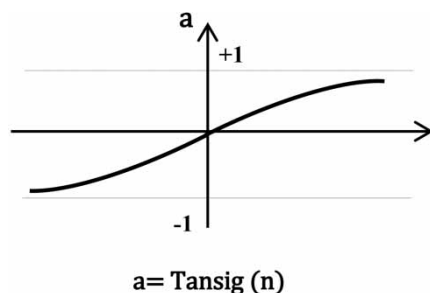


Figure 4 | Transfer function tangent sigmoid.

changed them to bipolar ( $-1$  to  $1$ ) with  $k$  segment criteria. Therefore, the results were bipolar.  $k$  was employed to change the output to the original form (Razavi 2006).

To obtain robust results from the ANN, low correlated inputs are preferred (Kuncheva 2004). According to protocol for developing ANN models stated by Wu et al. (2014), we can use only significant input parameters or can also consider input independence. Input parameters which were significant were chosen. Also, in our previous study, Asadollahfardi et al. (2016) performed a factor analysis to find suitable water quality input parameters for the RBF neural network and the results of factor analysis were four water quality parameters including total hardness (TH),  $\text{HCO}_3^-$ ,  $\text{Ca}^{2+}$  and  $\text{Cl}^-$  which were selected in this study and two other parameters including  $\text{Na}^+$  and  $\text{SO}_4^{2-}$  were added to them because they had a significant role in prediction of TDS in the river (input significance).

### Learning rate

Prior to training the MLP neural network, it is necessary to prepare the weights and biases and their values are chosen randomly between 0 and 1. After that, the neural network prepares for training. The training process needs a set of training data. Throughout training, weights and biases of the neural network are collaboratively adjusted to minimize the mean squared error (MSE) between the predicted and measured data. The performance of the algorithm is very sensitive to the appropriate setting of the learning rate. On the condition that the learning rate is very high, the algorithm may fluctuate and become unstable. On the condition that the learning rate is very small, the algorithm takes too long to achieve coverage. We selected the learning rate with an iterative procedure. The performance of the steepest descent algorithm would be increased if the learning rate is permitted to change during the training process (Amini 2008).

A back-propagation learning algorithm and fast convergences of the network parameters can be applied to determine the  $\alpha$  coefficient (learning rate). In each time step, small values of  $\alpha$  reduce variations of the parameters and slow the converging process. Meanwhile, high values of  $\alpha$  increase the convergence amount, resulting in instability in the network parameters. Obtaining a suitable  $\alpha$  is one of the most sensitive processes in using the algorithm for back

propagation. An adaptive  $\alpha$  aims to make the learning step as big as possible while keeping the learning stable (Asadollahfardi et al. 2016).  $\alpha$  is based on the steepest descent technique and aims to quickly minimize the sum square error of the outputs learning rate (Menhaj 1998). The learning rate is a hyper-parameter which controls the amount of adjustment of the weights of the MLP neural network with respect to the loss gradient performed by the user.

The steepest descent is likewise named the error back propagation (EBP) algorithm. For the EBP algorithm, the larger the training with constant  $\alpha$ , the faster and less stable the process will be. The Levenberg-Marquardt algorithm is much faster than the EBP algorithm and more stable than the Gauss-Newton algorithm (Yu & Wilamowski 2010). The Levenberg-Marquardt algorithm solves the flaw existing in both the gradient technique and the Gauss-Newton algorithm for artificial neural networks by a combination of two mentioned algorithms. The Levenberg-Marquardt algorithm also had a few disadvantages, but most of them have been solved by researchers such as Wilamowski & Yu (2010) and Transtrum & Senthna (2012). Wilamowski and Yu computed the Quasi-Hessian matrix and gradient vector directly without Jacobin matrix multiplications and storage. Therefore, the memory limitation of the Levenberg-Marquardt algorithm was solved. Transtrum & Senthna (2012) introduced some improvements to the Levenberg-Marquardt algorithm to improve both its convergence speed and strength to initial guess. They obtained the geodesic acceleration correction to the Levenberg-Marquardt algorithm step by using second order correction in the Taylor approximation of the residuals and ensuring that the extrinsic curvature of the model graph is small. They stated that the correction can be calculated at each step with minimal addition computation. Therefore, we used the Levenberg-Marquardt algorithm.

This study used three standard errors in the training stage. They were volume error (VE), root mean square error (RMSE) and mean bias error (MBE) (Equations (10) to (12); Kennedy & Neville 1964).

$$VE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \times 100 \quad (10)$$

$$MBE = \frac{1}{n} \sum_{t=1}^n \left| \frac{F_t - A_t}{A_t} \right| \quad (11)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (A_t - F_t)^2} \quad (12)$$

where,  $A_t$  is the measured data at time  $t$  ( $1 < t < n$ );  $n$  is the number of data;  $F_t$  is the predicted data at time  $t$ ;  $\bar{F}$  is the mean predicted data; and  $\bar{A}$  is the mean measured data.

We used the TH, bicarbonate ( $\text{HCO}_3^-$ ), calcium ions ( $\text{Ca}^{2+}$ ), sodium ion ( $\text{Na}^+$ ), chlorine ion ( $\text{Cl}^-$ ) and sulfate ion ( $\text{SO}_4^{2-}$ ) as the input data and predicted TDS at the Mosian monitoring station. One hundred and twenty monthly data were available from 2001–2010. Ninety-six sets of data were assigned for training. Twelve sets were used to validate the model and 12 sets were considered for model testing (Table 1). These data were available for authors. We can update our model whenever new data become available.

First, the input data were normalized. Then, the required numbers of layers and neurons were determined using a trial and error approach.

## Model efficiency

To evaluate the reliability of the results, the coefficient of determination ( $R^2$ ), index of agreement ( $IA$ ) and Nash-

**Table 1** | The statistical summary of data at Mosian station

Parameters	$\text{Ca}^{2+}$ , (mg/l)	$\text{Na}^+$ , (mg/l)	$\text{Cl}^-$ (mg/l)	$\text{SO}_4^{2-}$ (mg/l)	$\text{HCO}_3$ (mg/l)	TH (mg/l)	TDS (mg/l)
Number (months)	120	120	120	120	120	120	120
Average	3.40	2.20	1.85	2.14	2.98	2.35	432.98
Minimum	1.70	0.10	0.20	0.10	1.10	1.40	232.05
Maximum	7.80	8.30	9.60	9.51	5.30	5.20	1220.70
Standard deviation	0.96	1.78	1.48	1.53	0.67	0.65	187.51
Mode	2.9	1	0.9	1.4	2.9	2.1	320.45

Sutcliffe efficiency ( $E$ ) were applied (Equations (13) to (15); Heckman 1979; Krause *et al.* 2005; Willmott *et al.* 2012). We selected three indicators for performance of the models because testing model efficiency with three indicators is more reliable than using only one indicator (Noriasi *et al.* 2007).

$$R^2 = \left( \frac{\sum_{i=1}^n (A_t - \bar{A})(F_t - \bar{F})}{(\sum_{i=1}^n (A_t - \bar{A})^2) \sum_{i=1}^n ((F_t - \bar{F})^2)^{0.5}} \right)^2 \quad (13)$$

$$IA = d = 1.0 - \frac{\sum_{t=1}^N (A_t - F_t)^2}{\sum_{t=1}^N (|F_t - \bar{A}| + |A_t - \bar{A}|)^2} \quad (14)$$

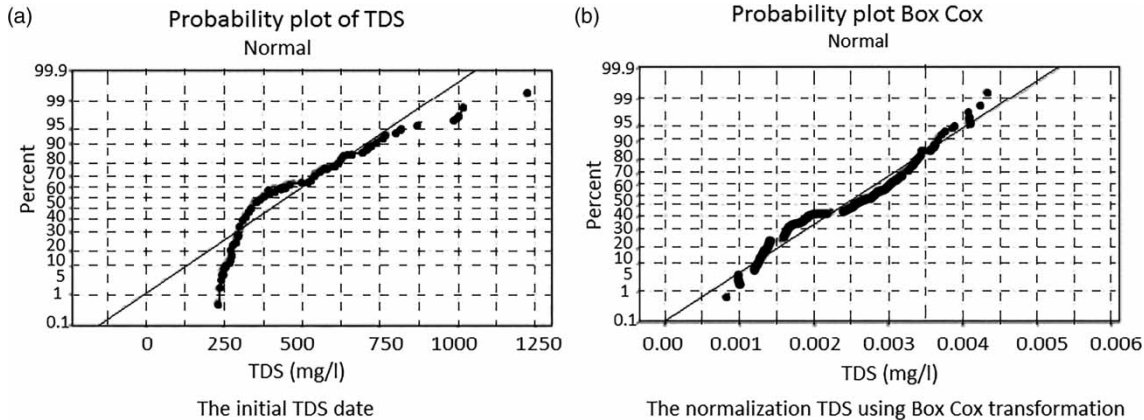
$$E = 1.0 - \frac{\sum_{t=1}^N (A_t - F_t)^2}{\sum_{t=1}^N (A_t - \bar{A})^2} \quad (15)$$

## RESULTS AND DISCUSSION

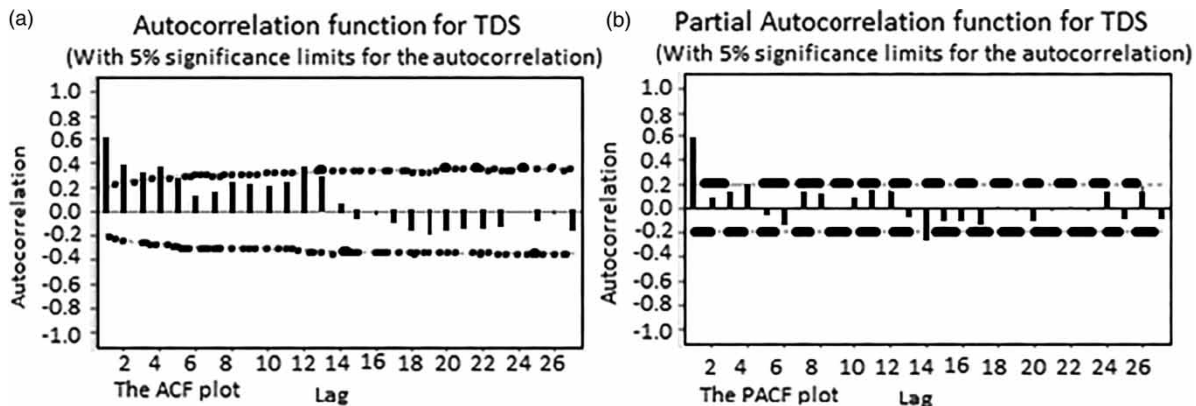
### Box-Jenkins results

Due to the necessity of the normalized data for model development, a normality test was conducted with the TDS data before and after Box Cox transformation (Figure 5).

To estimate the order of AR (autoregressive) and MA (moving average), ACF and PACF of TDS data were computed (Figure 6). Three models ARIMA (2,0,3) (1,0,2), ARIMA (2,0,3) (2,0,2) and A1RIMA (1,0,3) (1,0,1) were examined. Afterward, the AIC for each model was determined, which were 1226.9, 1223.5, and 1229.5, respectively. A model with the smallest AIC would be preferable, which was ARIMA (2, 0, 3) (2, 0, 2).



**Figure 5** | The normalized TDS data before and after Box Cox transformation. (a) The initial TDS data. (b) The normalization TDS using Box Cox transformation.



**Figure 6** | Plotted ACF and PACF for TDS data. (a) The ACF plot. (b) The PACF plot.

To investigate the independence of residuals assumption, ACF and PACF of the residuals were considered. Figure 7 indicates residual plots for TDS data using ARIMA (2, 0, 3) (2, 0, 2). Therefore, the normality of the residuals was obtained by fitting the ARIMA (2,0,3) (2,0,2) model and graph of normality (histogram), which indicated that the graph was not funnel-shaped.

Figure 8 illustrates the ACF and PACF of TDS residuals of TDS. The functions were varied between -0.2 and 0.2, which depicted the independence of residuals and the suitability of the selected model.

To investigate the robustness of the model, IA and R<sup>2</sup> of the measured and predicted TDS were calculated (Table 2). They could be considered acceptable and were 0.73 and 0.78, respectively.

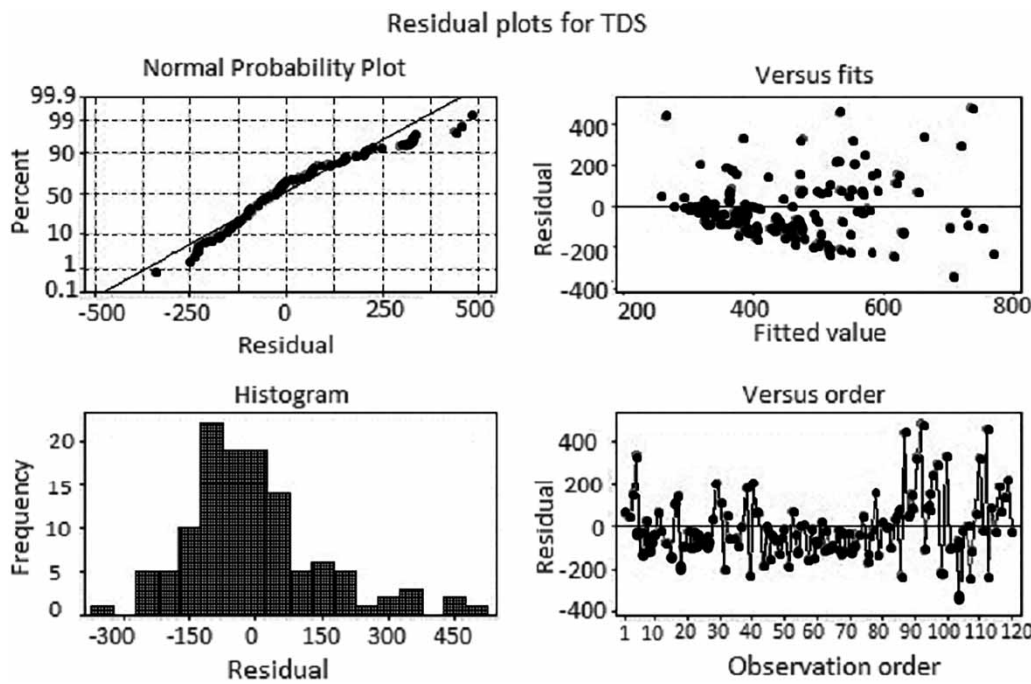


Figure 7 | Graphs of the residuals ARIMA (2, 0, 3) (2, 0, 2).

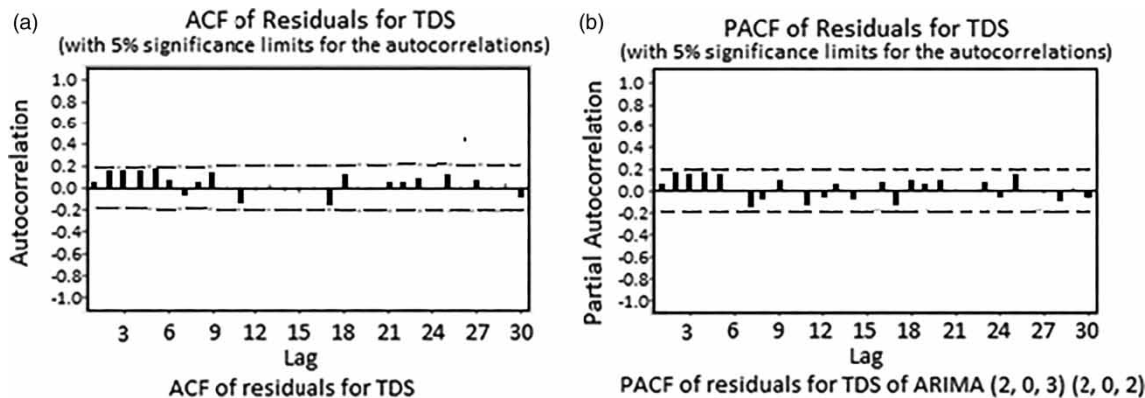


Figure 8 | The ACF and PACF of residuals of TDS.



**Table 2** | The predicted and the measured TDS based on ARIMA (2, 0, 3) (2, 0, 2) model

Months	Predicted data (mg/l)	Measured data (mg/l)	Lower limit	Upper limit
1	450.367	575.25	142.593	758.14
2	622.195	872.30	266.849	977.54
3	466.367	560.95	101.649	831.09
4	714.697	991.25	348.489	1080.90
5	424.513	531.70	57.951	91.077
6	533.371	449.15	164.431	902.31
7	407.223	344.50	36.187	778.26
8	398.286	539.50	26.763	769.81
9	542.163	661.05	170.620	913.71
10	493.017	555.10	121.057	864.98
11	588.222	747.50	215.801	960.64
12	441.634	534.95	69.068	814.20
Error criteria		$IA = 0.73$		$R^2 = 0.78$

## MLP neural network results

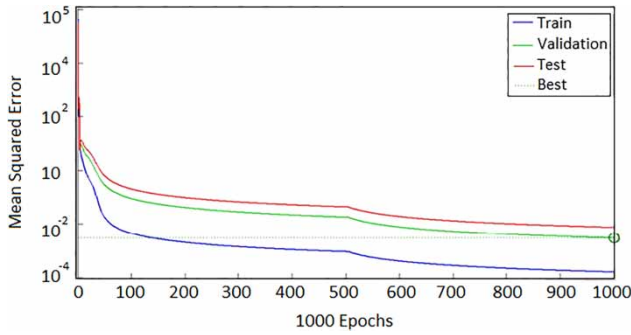
We selected the MLP with one hidden layer with different numbers of neurons, from 2 to 11 neurons, using Tansig, Purelin and Logsig transfer functions with the Levenberg-Marquardt (trainlm) training method to find the best model. To obtain the optimum performance of the model, we evaluated  $R^2$ ,  $IA$ ,  $RMSE$ ,  $E$  and  $MBE$  with one hidden layer. Therefore, the model was run with different and possible transfer and amounts of hidden layers and neurons (Table 3). The results indicated that an MLP with one hidden layer and 10 neurons in the hidden layer as well as the Tansig transfer function would give a good performance.

In higher numbers of epochs, the errors of training, validation and testing of MLP would decrease. So that the error would be insignificant in an epoch number of 1000 (Figure 9).

**Table 3** | The performance of the MLP with different transfer functions and neurons with a hidden layer

No	Number of neurons	Training method	Transfer function	The model performance				
				$R^2$	$RMSE$	$IA$	$E$	$MBE$
1	2	Levenberg-Marquardt (trainlm)	Tansig <sup>a</sup>	0.75	3.58	0.78	0.73	0.127
2			Purelin <sup>b</sup>	0.56	7.42	0.57	0.54	0.319
3			Logsig <sup>c</sup>	0.73	4.19	0.74	0.71	0.154
4	4		Tansig	0.81	2.14	0.84	0.80	0.094
5			Purelin	0.59	5.87	0.61	0.58	0.257
6			Logsig	0.80	1.98	0.82	0.80	0.085
7	6		Tansig	0.90	0.749	0.91	0.89	0.051
8			Purelin	0.63	5.23	0.66	0.61	0.228
9			Logsig	0.84	0.926	0.86	0.83	0.062
10	8		Tansig	0.89	0.841	0.92	0.89	0.045
11			Purelin	0.64	4.84	0.65	0.64	0.211
12			Logsig	0.86	0.957	0.88	0.84	0.052
13	<b>10</b>		<b>Tansig</b>	<b>0.91</b>	<b>0.547</b>	<b>0.94</b>	<b>0.90</b>	<b>0.023</b>
14			Purelin	0.68	3.21	0.70	0.67	0.157
15			Logsig	0.89	0.613	0.91	0.88	0.037
16	12		Tansig	0.91	0.677	0.92	0.89	0.031
17			Purelin	0.66	3.79	0.69	0.64	0.182
18			Logsig	0.87	0.655	0.89	0.86	0.041

<sup>a</sup>Tansig: tan-sigmoid transfer function.<sup>b</sup>Logsig: log-sigmoid transfer function.<sup>c</sup>Purelin: linear transfer function.

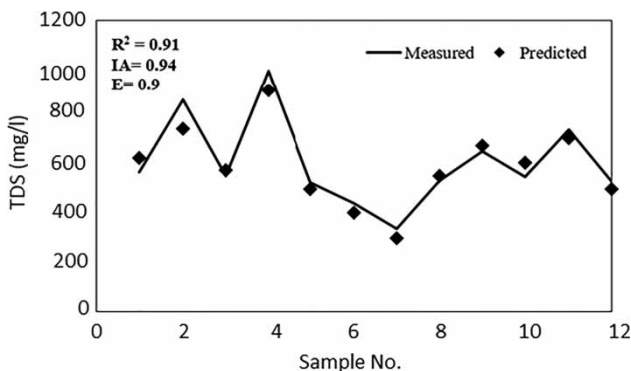


**Figure 9** | Decreasing errors in training, validation and testing of MLP neural network by increasing epochs.

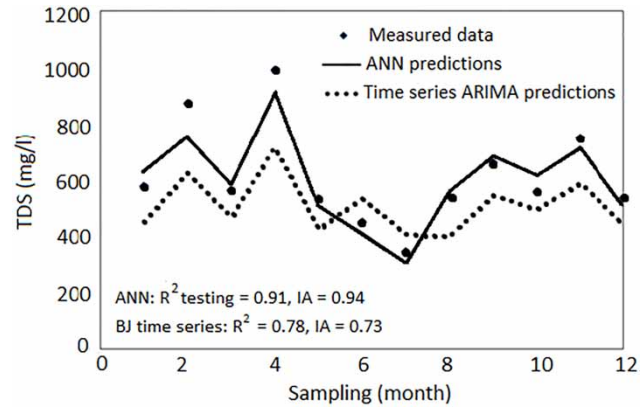
The acquired results of the MLP indicated a good agreement between the measured and the predicted amounts of TDS. So that,  $R^2$ ,  $IA$  and  $E$  were 0.91, 0.94 and 0.9 respectively (Figure 10).

Figure 11 indicates the comparison between measured data, time series Box-Jenkins prediction and MLP neural network prediction during testing. As indicated in Figure 11, the MLP method was more reliable than the Box-Jenkins time series in the prediction of TDS concentrations.

The difference between the present study and previous work with Asadollahfardi *et al.* (2016) is that we applied univariate Box-Jenkins time series and the MLP neural network, while in the previous study we applied only a RBF neural network. We conducted this study to find the reliability of Box-Jenkins time series and the MLP neural network compared to our previous study using the RBF neural network in the river, since predicting accurate and reliable TDS is very significant in water quality management.



**Figure 10** | The comparison of measured and MLP prediction of TDS concentration.



**Figure 11** | The comparison between measured data, time series Box-Jenkins prediction and MLP neural network prediction during testing.

The difference between our study and the Montaseri *et al.* (2018) investigation is that they applied different types of ANN to predict TDS of four different rivers, including Zaandam rud River, and compared different ANNs to forecast TDS, whereas we compared the MLP neural network and Box-Jenkins time series to predict TDS of Zāyandé-Rūd River.

## CONCLUSIONS

Considering the results and discussions of the application of MLP and Box-Jenkins time series in the prediction of TDS concentration at the Zāyandé-Rūd River, the following conclusions can be drawn:

- We developed ARIMA (2, 0, 3) (2, 0, 2) Box-Jenkins time series in the prediction of TDS in the river. The coefficient of determination ( $R^2$ ) and index of agreement ( $IA$ ) between the measured and predicted TDS were equal to 0.78 and 0.73, respectively.
- Applying the MLP with 10 neurons in a hidden layer using the Tansig transfer function, a suitable TDS prediction was achieved. Values of  $R^2$  and  $IA$  between the measured and predicted TDS concentrations were 0.91 and 0.94, respectively.
- The model efficiency of the MLP neural network to predict TDS in Zāyandé-Rūd River was better than that of the Box-Jenkins time series.

## REFERENCES

- Abudu, S., King, J. P. & Sheng, Z. 2011 Comparison of the performance of statistical models in forecasting monthly total dissolved solids in the Rio Grande. *Journal of the American Water Resources Association (JAWRA)* **48** (1), 10–23.
- Adhikari, R. 2015 A neural network based linear ensemble framework for time series forecasting. *Neurocomputing* **157**, 231–242.
- Amini, J. 2008 Optimum learning rate in back-propagation neural network for classification of satellite images (IRS-ID). *Scientia Iranica* **15** (6), 558–567.
- Arya, F. & Zhang, L. 2015 Time series analysis of water quality parameters at Stillaguamish River using order series method. *Stochastic Environmental Research and Risk Assessment* **29** (1), 227–239.
- Asadollahfardi, G. 2002 Analysis of surface water quality in Tehran. *Water Quality Research Journal of Canada* **37** (2), 489–511.
- Asadollahfardi, G., Rahbar, M. & Fatemiaghda, M. 2012a Application of time series models to predict water quality of upstream and downstream of the Latian Dam in Iran. *Universal Journal of Environmental Research and Technology* **2** (1), 26–36.
- Asadollahfardi, G., Taklifi, A. & Ghanbari, A. 2012b Application of artificial neural network to predict TDS in Talkheh Rud River. *Journal of Irrigation and Draining Engineering* **138**, 363–370.
- Asadollahfardi, G., Meshkatadini, A., Homatoun Aria, S. & Roohani, N. 2016 Application of artificial neural networks to predict total dissolved solids in the river Zayanderud, Iran. *Environmental Engineering Research* **21** (4), 333–340.
- Asadollahfardi, G., Zangooei, H., Homayoun Aria, S. & Danesh, E. 2017 Application of artificial neural networks to predict total dissolved solid at the Karaj Dam. *Environmental Quality Management* **26** (3), 55–72.
- Babu, C. N. & Reddy, B. E. 2014 A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data. *Applied Soft Computing* **23**, 27–38.
- Box, G. E. P. & Jenkins, G. M. 1976 *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco.
- Cybenko, G. 1989 Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals and Systems* **2** (4), 303–314.
- Espejo, M. R. 2004 The Oxford dictionary of statistical terms. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **167** (2), 377–377.
- Faruk, D. Ö. 2010 A hybrid neural network and ARIMA model for water quality time series prediction. *Engineering Applications of Artificial Intelligence* **23** (4), 586–594.
- Gairaa, K., Khellaf, A., Messlem, Y. & Chellali, F. 2016 Estimation of the daily global solar radiation based on Box-Jenkins and NN model, A combined approach. *Renewable and Sustainable Energy Reviews* **57**, 238–249.
- Heckman, J. J. 1979 Sample selection bias as a specification error. *Econometrica: Journal of the Econometric Society* **47** (1), 153–161.
- Hornik, K. 1991 Approximation capabilities of multilayer feed forward networks. *Neural Networks* **4** (2), 251–257.
- Hornik, K. 1993 Some new results on neural network approximation. *Neural Networks* **6** (8), 1069–1072.
- Hornik, K., Stinchcombe, M. & White, H. 1989 Multilayer feed forward networks are universal approximators. *Neural Networks* **2** (5), 359–366.
- Hossieni Abri, H. 2000 *Zayanderud River From the Origin to the Lagoon*. Golha Publisher, Esfahan, Iran (in Farsi).
- Huang, W. & Foo, S. 2002 Neural network modeling of salinity variation in Apalachicola River. *Water Research* **36** (1), 356–362.
- Jayawardena, A. & Lai, F. 1989 Time series analysis of water quality data in Pearl River, China. *Journal of Environmental Engineering* **115** (3), 590–607.
- Kanani, S. H., Asadollahfardi, G. & Ghanbari, A. 2008 Application of artificial neural network to predict total dissolved solid in the Achechay River basin. *World Applied Sciences Journal* **4** (5), 646–654.
- Kennedy, J. B. & Neville, A. D. 1964 *Basic Statistical Methods for Engineers and Scientists*, 2nd edn. Holden-Day, San Francisco, California; Harper and Row, New York.
- Krause, P., Boyle, D. P. & Bäse, F. 2005 Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences* **5**, 89–97.
- Kuncheva, L. 2004 *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, New York, USA.
- Kurunc, A., Yurekli, K. & Cevik, O. 2005 Performance of two stochastic approaches for forecasting water quality and stream flow data from Yeşilırmak River, Turkey. *Environmental Modeling & Software* **20** (9), 1195–1200.
- Leshno, M., Lin, V. Y., Pinkus, A. & Schocken, S. 1993 Multilayer feed forward networks with a nonpolynomial activation function can approximate any function. *Neural Networks* **6** (6), 861–867.
- Maier, H. R. & Dandy, G. C. 1996 The use of artificial neural networks for the prediction of water quality parameters. *Water Resources Research* **32** (4), 1013–1022.
- McNeil, V. H. & Cox, M. E. 2000 Relationship between conductivity and analysed composition in a large set of natural surface-water samples, Queensland, Australia. *Environmental Geology* **39** (12), 1325–1333.
- Menhaj, M. 1998 *Computational Intelligence, Fundamentals of Artificial Neural Networks*, Vol. 1. Amirkabir University, (in Farsi).
- Misaghi, F. & Mohammadi, K. 2003 Estimating Water Quality Changes in the Zayandeh Rud River Using Artificial Neural Network Model. In: *Annual Meeting of Canadian Society of Agricultural Engineering*, 6–9 July, Montreal, Canada.
- Moienian, M. 2008 *Natural View of Zayandrud in Isfahan*. Jahdaneshgahi Publisher, Esfahan, Iran (in Farsi).
- Montaseri, M., Ghavidel, S. Z. Z. & Sanikhani, H. 2018 Water quality variation in different climates of Iran :toward

- modelling total dissolved solid using soft computing techniques. *Stochastic Environmental Research and Risk Assessment* **32**, 2253–2273.
- Nemati, S., Naghipour, L. & Hasan Fazeli Far, M. 2014 Artificial neural network modeling of total dissolved solid in the Simineh River, Iran. *Journal of Civil Engineering and Urbanism* **4** (1), 8–14.
- Noriasi, D. N., Arnold, J. G., VanLiew, M. W., Binger, R. L., Hamel, R. D. & Veith, T. L. 2007 Model evaluation guideline for systematic quantification of accuracy in watershed simulation. *Transaction of ASABE* **50** (3), 685–690.
- Ranjbar, M. & Khaledian, M. 2014 Using ARIMA time series model in forecasting the trend of changes in qualitative parameters of Sefidrud River. *Journal of Applied and Basic Sciences* **8** (3), 346–351.
- Razavi, F. 2006 *Rain Prediction Applying Artificial Neural Network*. M.S thesis, Amir kabir University of Technology, Tehran, Iran.
- Salmani, M. H. & Salmani Jajaei, E. 2016 Forecasting models for flow and total dissolved solid in Karoun river-Iran. *Journal of Hydrology* **539**, 148–159.
- Sun, H. & Koch, M. 2001 Case study: analysis and forecasting of salinity in Apalachicola Bay, Florida, using Box-Jenkins ARIMA models. *Journal of Hydraulic Engineering* **127** (9), 718–728.
- Taneja, K., Ahmad, S., Ahmad, K. & Attri, S. D. 2016 Time series analysis of aerosol optical depth over New Delhi using Box-Jenkins ARIMA modeling approach. *Atmospheric Pollution Research* **7** (4), 585–596.
- Transtrum, M. K. & Senthna, J. P. 2012 Improvement to the Levenberg-Marquardt algorithm for nonlinear least-square minimization. Available at <https://arxiv.org/abs/1201.5885> (accessed 2 August 2018).
- Wilamowski, B. M. & Yu, H. 2010 Improved computation for Levenberg-Marquardt algorithm training. *IEEE Transactions on Neural Network* **21** (6), 930–937.
- Willmott, C. J., Robeson, S. M. & Matsuura, K. 2012 A refined index of model performance. *International Journal of Climatology* **32** (13), 2088–2094.
- Wu, W., Dandy, G. C. & Maier, H. R. 2014 Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process of drinking water quality modelling. *Environmental Modelling & Software Journal* **54**, 108–127.
- Yu, H. & Wilamowski, B. M. 2010 Levenberg-Marquardt training, K10149\_C012 indd, Chapter 12, pp. 1–12.
- Zhang, Q. & Stanley, S. J. 1997 Forecasting raw-water quality parameters for the North Saskatchewan River by neural network modeling. *Water Research* **1354** (97), 72–79.

First received 3 August 2018; accepted in revised form 29 August 2018. Available online 21 September 2018