Water Science and Engineering, 2010, 3(3): 269-281
doi:10.3882/j.issn.1674-2370.2010.03.003

http://www.waterjournal.cn
e-mail: wse2008@vip.163.com

Comparison of performance of statistical models in forecasting monthly streamflow of Kizil River, China

Shalamu ABUDU*[1, 2], Chun-liang CUI[1], James Phillip KING[2], Kaiser ABUDUKADEER[3]

1. Xinjiang Water Resources Research Institute, Urumqi 830049, P. R. China
2. Civil Engineering Department, New Mexico State University, NM 88001, USA
3. Xinjiang Water Resources Bureau, Urumqi 830000, P. R. China

Abstract: This paper presents the application of autoregressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA), and Jordan-Elman artificial neural networks (ANN) models in forecasting the monthly streamflow of the Kizil River in Xinjiang, China. Two different types of monthly streamflow data (original and deseasonalized data) were used to develop time series and Jordan-Elman ANN models using previous flow conditions as predictors. The one-month-ahead forecasting performances of all models for the testing period (1998-2005) were compared using the average monthly flow data from the Kalabeili gaging station on the Kizil River. The Jordan-Elman ANN models, using previous flow conditions as inputs, resulted in no significant improvement over time series models in one-month-ahead forecasting. The results suggest that the simple time series models (ARIMA and SARIMA) can be used in one-month-ahead streamflow forecasting at the study site with a simple and explicit model structure and a model performance similar to the Jordan-Elman ANN models.

Key words: *time series model; Jordan-Elman artificial neural networks model; monthly streamflow forecasting*

1 Introduction

Streamflow forecasting is of great importance to water resources management and planning. Medium- to long-term forecasting, at weekly, monthly, seasonal, or even annual time scales, is particularly useful in reservoir operations and irrigation management, as well as institutional and legal aspects of water resources management and planning. Due to their importance, a large number of forecasting models have been developed for streamflow forecasting, including concept-based process-driven models such as the low flow recession model and rainfall-runoff models, and statistics-based data-driven models such as regression models, time series models, artificial neural network models, fuzzy logic models, and the nearest neighbor model (Wang 2006). In recent years, combinations of conceptual and statistical model approaches have been used in medium- to long-term streamflow forecasting for the major rivers in the United States (Karamouz and Zahraie 2004).

*Corresponding author (e-mail: *shalamu@yahoo.cn*)
Received May 20, 2010; accepted Jul. 12, 2010

Of various streamflow forecasting methods, time series analysis has been most widely used in previous decades because of its forecasting capability, inclusion of richer information, and more systematic way of building models in three modeling stages (identification, estimation, and diagnostic check), as standardized by Box and Jenkins (1976). The application of time series models in streamflow forecasting includes univariate models that deal with only one time series and more complex multivariate models (dynamic regression models, also called transfer function-noise (TFN) models), which incorporate exogenous time series variables. The univariate time series models, including the autoregressive integrated moving average (ARIMA) model and its derivatives such as seasonal ARIMA (SARIMA), periodic ARIMA, and deseasonalized ARIMA models, have long been applied in streamflow forecasting, particularly in the modeling of monthly streamflow (McKerchar and Delleur 1974; Noakes et al. 1985; Salas 1992; Bender and Simonovic 1994; Hipel and McLeod 1994; Abrahart and See 2000; Yürekli et al. 2005). The application of TFN models with exogenous variables in streamflow forecasting was also described in these literatures. Thompstone et al. (1985) compared deseasonalized ARIMA, periodic autoregressive (PAR), and TFN models utilizing rainfall and snowmelt inputs, with a conceptual model. They found that the TFN model performs better than other models when forecasting quarter-monthly streamflow. Awadallah and Rousselle (2000) used El Niňo Southern Oscillation (ENSO) sea-surface temperature signals as exogenous input variables to develop a TFN model to forecast summer runoff of the Nile River. Their TFN model suggested that the ENSO input explained 63% of the variability of summer runoff of the Nile River. Mondal and Wasimi (2005) proposed a periodic TFN model and applied it to the monthly forecasting of the Ganges River flow using monthly rainfall data of northern India as the predictor. The results suggested that the methodology has the potential capability of capturing the seasonally varying dynamic relationship between monthly rainfall and streamflow processes.

The time series models used in the streamflow forecasting process are mostly linear models. They were built under the assumption that the process follows normal distribution, but most streamflow processes are nonlinear (Wang 2006). Hence, the recently developed machine-learning technique, the artificial neural networks (ANN) model, has gained more and more popularity for hydrological forecasting in recent decades because of its ability to identify complex nonlinear relationships between input and output data sets without the necessity of understanding the nature of the phenomena and without making any underlying assumptions. Previous studies have concluded that ANN models are useful for forecasting streamflows. Markus et al. (1995) predicted monthly flow of the Rio Grande near Del Norte in southern Colorado using ANN models and compared the results with the periodic TFN model. The study showed that the ANN models provided slightly better results than the periodic TFN models using standardized monthly flow data. Hsu et al. (1995) concluded that the ANN model is an effective alternative to the ARMAX (autoregressive moving average with exogenous inputs) model. Huang et al. (2004) compared the ANN and ARIMA models for daily, monthly, quarterly, and annual flow forecasting and concluded that the ANN model provided better forecasting accuracy than the ARIMA model. Many studies have confirmed the superiority or comparableness of the ANN model over the traditional statistical and/or conceptual techniques in modeling hydrological processes (Shamseldin 1997; Coulibaly et al. 2000; Govindaraju and Rao 2000; Salas et al. 2000; Tokar and Markus 2000; Dibike and Solomatine 2001; Abrahart et al. 2004).

The ANN models used in streamflow forecasting are usually multilayer perceptron (MLP). Unlike MLP, the Jordan-Elman neural networks combine the past values of the context unit with the present input to obtain the present net output. The Jordan-Elman neural networks extend the MLP with context units, which provide an ability to extract temporal information from the data. However, its application in streamflow forecasting is limited. This study applied the Jordan-Elman ANN models to monthly streamflow forecasting of the Kizil River in Xinjiang, China. Due to their unique ability to extract temporal data, the performance of the Jordan-Elman ANN models was compared with the time series models using the previous flow data as inputs. The forecasting performance was evaluated using one-month-ahead forecasts at the study site and the results are discussed.

2 Methodologies

2.1 Time series models

Several types of ARIMA modeling methods and their derivatives can be used in the modeling of monthly streamflow, including SARIMA, periodic ARIMA, and deseasonalized ARIMA models. The deseasonalized ARIMA and SARIMA modeling strategies were used in this study.

The general form of the ARIMA model is (Vandaele 1983)

$$\varphi(B)x_t = \theta(B)a_t \tag{1}$$

where $\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - .... - \varphi_p B^p$ is the non-seasonal autoregressive polynomial; B is the backward shift operator; φ_p represents the autoregressive parameters of the model; p is the order of autoregressive polynomial; x_t is the stationary series after differencing, $x_t = (1-B)^d X_t$; d is the number of non-seasonal differencing; X_t is the dependent variable; $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - .... - \theta_q B^q$ is the non-seasonal moving average polynomial; θ_q is the moving average of the model; q is the order of moving average polynomial; and a_t is the white noise process.

The SARIMA model is a type of time series model either multiplicative or nonmultiplicative, and the latter is the simplified form of the former. The general form of the general multiplicative SARIMA model, the SARIMA $(p,d,q)\times(P,D,Q)_s$ Process, is expressed as (Vandaele 1983)

$$\varphi(B)\Phi(B^s)x_t = \theta(B)\Theta(B^s)a_t \tag{2}$$

where $x_t = (1-B)^d (1-B^s)^D X_t$; D is the number of seasonal differencing; $\Phi(B^s)$ is the seasonal autoregressive polynomial, $\Phi(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - .... - \Phi_P B^{Ps}$; s is the order of seasonal differencing; P is the order of seasonal autoregressive polynomial; $\Theta\left(B^s\right)$ is the seasonal moving average polynomial, $\Theta\left(B^s\right) = 1 - \Theta_1 B^s - \Theta_2 B^{2s} - ... - \Theta_Q B^{Qs}$; and Q is the order of seasonal moving average polynomial.

Examination of the autocorrelation function (ACF) and partial autocorrelation function (PACF) provides a thorough basis for analyzing the system behavior under time dependence, and suggests the appropriate parameters to be included in the model. The Box and Jenkins (1976) three-stage standard modeling procedure (identification, estimation, and diagnostic check) was used to develop time series models. The detailed algorithm has also been described by Vandaele (1983). Statistical Analysis Software (SAS) Version 9.1 was used for time series model development in this study.

2.2 Jordan-Elman ANN models

ANN are flexible mathematical structures capable of identifying complex nonlinear relationships between input and output data sets. The most commonly used type of ANN is a feed forward network termed MLP. In this type of network, the artificial neurons, or processing units, are arranged in a layered configuration containing an input layer, usually one hidden layer, and an output layer. Units in the hidden and output layers are connected to all of the units in the preceding layer. Each connection carries a weighting factor. The weighted sum of all inputs to a processing unit is calculated and compared to a threshold value. The activation signal is then passed through a mathematical transfer function to create an output signal that is sent to processing units in the next layer. Kim and Valdes (2003) described three-layered feed forward neural networks (FFNN) and provided a general framework for representing nonlinear functional mapping between a set of input and output variables. Three-layered FFNN are based on a linear combination of the input variables, which are transformed by a nonlinear activation function. The explicit expression for an output value of FFNN for one output neuron is

$$\hat{y}_{pk} = f_0 \left[\sum_{j=1}^{M} w_{kj} f_{\mathrm{h}} \left(\sum_{i=1}^{N} w_{ji} x_{pi} + w_{j0} \right) + w_{k0} \right] \tag{3}$$

where $\hat{y}_{pk}$ is the model response for pattern (observation) p and the kth output neuron; f_0 is the activation function for the output neuron; w_{kj} is a weight in the output layer connecting the jth neuron in the hidden layer and the kth neuron in the output layer; f_{h} is the activation function of the hidden neuron; w_{ji} is a weight in the hidden layer connecting the ith neuron in the input layer and the jth neuron in the hidden layer; x_{pi} is a value of the ith input for pattern p; w_{j0} is the bias for the jth hidden neuron; w_{k0} is the bias for the kth output neuron; and N and M are the total numbers of neurons in the input hidden layers, respectively. The weights are different in the hidden and output layers, and their values can be changed during the process of network training. The relationship of the available input and output variables is generated by the training process.

The Jordan-Elman networks are essentially the extension of MLP networks with context units that can remember past activities. With this unique feature, this type of network is better fitted to model time series because it can extract information based on the autocorrelation of the time series data. Unlike MLP, the Jordan-Elman networks combine the past values of the context unit with the present input to obtain the present net output. Fig. 1 describes the structure of a Jordan-Elman ANN model. Context units are required when learning patterns over time (i.e., when the past value of the network influences the present processing). In the Elman network, the outputs of the hidden processing elements from the previous time step are copied to the context units. In the Jordan network, the output of the network is copied to the context units. In addition, the context units are locally recurrent (i.e., they feed back to themselves). The local recurrence decreases the values by a multiplicative time constant as they are fed back. This constant determines the memory depth (i.e., how long a given value fed to the context unit will be remembered). One can treat the context units as input units, just as if they were obtained from an external source such as a file. Since the recurrent connections within the context units are fixed, static backpropagation is used to train these networks. The objective of the backpropagation training process is to adjust the weights of the network to minimize the sum of square errors of the network, which approximates the model outputs to the target values with a selected error goal $E(m)$:

$$E(m) = \frac{1}{2}\sum_{p=1}^{m}\left[y_p(m) - \hat{y}_p(m)\right]^2 \tag{4}$$

where $y_p(m)$ is the desired target response, and $\hat{y}_p(m)$ is the actual response of the network at the *m*th iteration for pattern *p*.

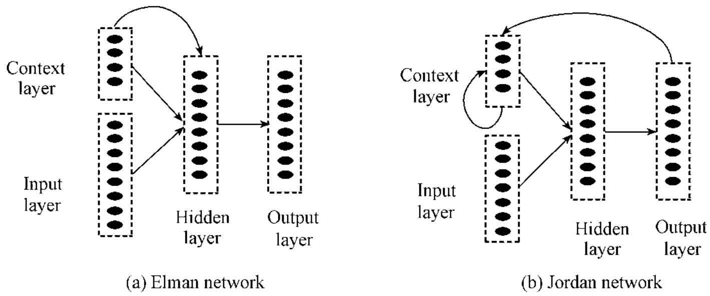

Fig. 1 Block diagrams of Jordan-Elman ANN model

NeuroSolutions Version 5.1 software, a neural network development environment, was used in this study for neural network modeling.

3 Study area

The Kizil River is located in Kashgar and Kizilsu prefectures in Xinjiang, China. The geographical location is approximately between longitudes 73°30′E and 77°30′E, and between latitudes 38°00′N and 40°30′N. The Kizil River Basin above the Kalabeili gaging station was selected as a study site for this research. The Kalabeili gaging station was used as a forecasting point (Fig. 2). It is located at latitude 39°33′N and longitude 75°12′E. The area of the drainage basin upstream from the gaging station is about 13 700 km^2, and the elevation is 1 700 m above sea level. The gaging station has observed daily streamflow data beginning from 1958. The monthly average streamflow at this station from 1959 to 2005 was used in this study.

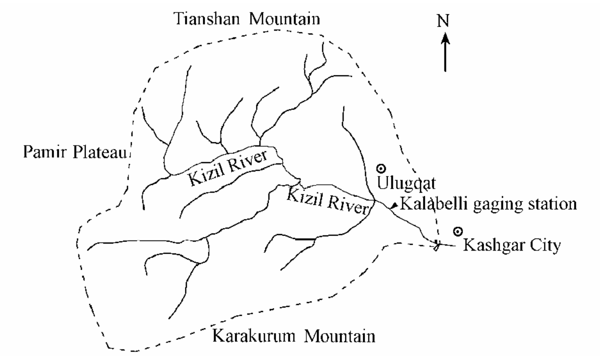

Fig. 2 Location of study site, Kizil River Basin above Kalabeili gaging station

4 Results and discussion

4.1 Time series models

Monthly streamflow time series exhibit periodical variation to the order of 12. To obtain a better approximation of normal distribution for the time series model, deseasonalization was applied to the monthly data. Except for the SARIMA model developed for original data, the deseasonalization was performed using monthly averages and the standard deviation in order to remove the seasonal variation of the data. The deseasonalized series is the series after removing the seasonal variation by standardization using the following expression:

$$y_t = \frac{Y_t - \overline{Y}_t}{\hat{\sigma}_t} \tag{5}$$

where y_t is the deseasonalized monthly flow for month *t*, Y_t is the original monthly flow for month *t*, $\overline{Y}_t$ is the sample average of the original monthly flow for month *t*, and $\hat{\sigma}_t$ is the sample standard deviation of the original monthly flow for month *t*.

The deseasonalized monthly flow was used to fit the ARIMA model. The entire data set was divided into a calibration set and a testing set, which covered the data periods from 1958 to 1997 and from 1998 to 2005, respectively. Based on ACF and PACF, the following SARIMA and ARIMA models, which passed all the diagnostic checks, were developed for the series:

Original data SARIMA model:

$$\left(1-0.329B\right)y_t = \left(1-0.813B^{12}\right)a_t \tag{6}$$

where $y_t = Y_t - Y_{t-12}$.

Original data SARIMA forecasting model:

$$\hat{y}_t = 0.329 y_{t-1} + 0.813 y_{t-12} - 0.813 \hat{y}_{t-12} \tag{7}$$

where $\hat{y}_t$ is the forecasted value for month *t*, and $\hat{y}_{t-12}$ is the forecasted value for month $t-12$, that is, 12 months earlier.

Deseasonalized data ARIMA model:

$$\left(1 - 0.366B - 0.099B^2 + 0.171B^3\right) y_t = a_t \tag{8}$$

where y_t is calculated as in Eq. (5).

Deseasonalized data ARIMA forecasting model:

$$\hat{y}_t = 0.366 y_{t-1} + 0.099 y_{t-2} - 0.171 y_{t-3} \tag{9}$$

One-month-ahead rolling forward forecasts were made for eight years (1998 to 2005) to evaluate the performance of the models.

4.2 Jordan-Elman ANN models

To compare the performance of ARIMA and SARIMA models with the Jordan-Elman ANN models, four Jordan-Elman ANN models were developed based on the sample autocorrelation functions of original monthly data and deseasonalized monthly data and final structures of the ARIMA and SARIMA models. The ANN-1 and ANN-2 models corresponded to the SARIMA model constructed for original data, and the ANN-3 and ANN-4 models were based on the structure of the ARIMA model developed for deseasonalized data. These four models are as follows:

ANN-1: inputs were monthly flow data at lags 1, 12, and 13:

$$y_t = f\left(y_{t-1}, y_{t-12}, y_{t-13}\right) \tag{10}$$

ANN-2: inputs were monthly flow data at lags 1, 6, 11, 12, 13 and the monthly historical average flow:

$$y_t = f\left(y_{t-1}, y_{t-6}, y_{t-11}, y_{t-12}, y_{t-13}, \overline{y}_t\right) \tag{11}$$

ANN-3: inputs were the deseasonalized data at lags 1, 2, and 3:

$$y_t = f\left(y_{t-1}, y_{t-2}, y_{t-3}\right) \tag{12}$$

ANN-4: inputs were the deseasonalized data at lags 1, 2, 3, 4, 5, 6, and 7:

$$y_t = f\left(y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4}, y_{t-5}, y_{t-6}, y_{t-7}\right) \tag{13}$$

As in the data partitioning in SARIMA and ARIMA modeling, the total data period (1959 to 2005) was divided into a calibration period (1959 to 1997) and a testing period (1998 to 2005). Then, the calibration period was divided further into a training period (1959 to 1990) and a cross validation period (1991 to 1997). The training process utilized the Jordan-Elman networks with the context unit controlling the forgetting factor through the time constant of 0.8 (usually between 0 and 1). The transfer function of the context unit employed a linear integrator axon. The hyperbolic tangent function was used as the activation function in the hidden layer with the momentum learning rule with a step size of 0.1 and a momentum of 0.7. The output layer utilized the linear function. The training termination criteria employed cross validation techniques that stop the training when the cross validation error begins to increase. The number of maximum training epochs was set as 3 000 and the training was terminated when there was no further improvement in the cross validation after 300 epochs. The networks were trained six to ten times and the best networks were selected based on the minimum cross validation error. The best weights of the network were automatically saved when the cross validation error reached its lowest point. One hidden layer was chosen and the nodes in the hidden layer were decided by trial and error. The best network structures were identified as 3-9-1 for ANN-1 and ANN-3, 6-7-1 for ANN-2, and 7-5-1 for ANN-4.

4.3 Model comparison

Using the SARIMA, ARIMA and Jordan-Elman ANN models described in the previous sections, one-month-ahead forecasts were performed for the period from January 1998 to December 2005 (a total of 96 months), in order to evaluate the performance of the models. The SARIMA and ARIMA models utilized rolling forward forecasts in which the model parameters were modified for every month forecasted when the new observation was available. To compare the statistical performance of the models for one-month-ahead forecasting, the following statistical indices were used:

Coefficient of determination (R^2):

$$R^2 = \left[\frac{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)\left(F_i - \bar{F}\right)}{\sqrt{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2 \sum_{i=1}^{n}(\left(F_i - \bar{F}\right)^2}} \right]^2 \tag{14}$$

Root mean squared error (*RMSE*):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - F_i\right)^2} \tag{15}$$

Model efficiency (*E*):

$$E = 1 - \frac{\sum (Y_i - F_i)^2}{\sum (Y_i - \bar{Y})^2} \tag{16}$$

where Y_i is the observed flow, F_i is the forecasted flow, n is the number of data points, $\bar{Y}$ is the average of observed flow, and $\bar{F}$ is the average of forecasted flow.

The model efficiency is a model evaluation criterion proposed by Nash and Sutcliffe (1970). A model efficiency of 90% or above indicates very satisfactory performance. A value in the range of 80% to 90% indicates fairly good performance. A value below 80% indicates a questionable fit.

The one-month-ahead forecasting performances of all models for the calibration and testing periods are shown in Table 1. Based on the performance comparison of the models, the ARIMA model for deseasonalized data performed slightly better than SARIMA models developed for original data (the model efficiencies are 0.85 and 0.88 for SARIMA and 0.87

and 0.89 for ARIMA in calibration and testing periods, respectively). Of the four ANN models, ANN-4 performed slightly better than the others, although the improvement was not significant. As can be seen, the ANN-4 developed using deseasonalized monthly data resulted in the lowest forecasting errors (18.89 m^3/s and 21.12 m^3/s), the highest coefficients of determination (0.88 and 0.90), and the highest model efficiencies (0.88 and 0.89), as compared to other models, in calibration and testing periods, respectively. The performance of the ARIMA model was comparable to that of the ANN-4 model with a similar performance (the root mean squared errors are 19.06 m^3/s and 21.23 m^3/s, the coefficients of determination are 0.88 and 0.89, and the model efficiencies are 0.87 and 0.89 for the calibration and testing periods, respectively). In general, no significant improvement could be observed for ANN models over SARIMA and ARIMA models in one-month-ahead forecasts. This may be due to the fact that only the previous monthly flow was considered as the forecasting inputs in both time series and ANN models. Although nonlinear in nature, the ANN models are considered deterministic models that cannot capture the stochasticity of the streamflow process. In contrast, the time series models can account for the stochasticity of the streamflow process although they are essentially linear models. In general, their performances are equivalent. However, the performance of the ANN models may be improved if other proper exogenous variables are identified and used as inputs.

Table 1 Comparison of model performance in calibration and testing periods

Model	Calibration period (n = 468)			Testing period (n = 96)		
	R^2	$RMSE$ (m^3/s)	E	R^2	$RMSE$ (m^3/s)	E
SARIMA	0.85	20.80	0.85	0.88	21.50	0.88
ARIMA	0.88	19.06	0.87	0.89	21.23	0.89
ANN-1	0.83	22.41	0.83	0.88	22.25	0.87
ANN-2	0.86	19.88	0.86	0.88	22.33	0.87
ANN-3	0.87	19.17	0.87	0.90	22.09	0.88
ANN-4	0.88	18.89	0.88	0.90	21.12	0.89

The time series models (SARIMA and ARIMA) and the Jordan-Elman ANN models yielded similar performances in one-month-ahead forecasts for testing periods from 1998 to 2005, a total of 96 data points. The scatter plots of observed monthly flow and one-month-ahead forecasts of all models are given in Fig. 3. As can be seen, all the models tend to underestimate high flows. This may be due to the fact that only previous flow conditions were used as predictors in the forecasting models. Other important factors that have significant contributions to the streamflow processes, such as precipitation, snowmelt, and temperature, were not included in the modeling process due to data limitation. In general, the deseasonalized data ANN models seemed to show slightly better performance compared to time series models, with a R^2 of 0.90, but the slope of the fitted line was less than that of time series models (the slope of the fitted line was 0.84 for ANN-4, while the ARIMA model generated a slope of 0.89). No significant difference was observed between all models in forecasting monthly flow for the testing period. This again shows that although the time series and Jordan-Elman ANN models are different in model structure and algorithm, they are essentially using the same information (previous flow conditions) as inputs to the forecasting models. It would make a difference if there were more information, such as rainfall, snowpack, etc., included in the Jordan-Elman ANN models. It is more difficult to develop time series models with more information than ANN models. As in this study, the ANN model may not improve model performance compared to the simple ARIMA model for forecasting monthly flow at the study site, although it has the capability of mapping complex nonlinear relationships between input and output data sets. This suggests that more investigation needs to be done at the study site to find the monthly relationships between other predictors in order to develop more accurate streamflow forecasting models.

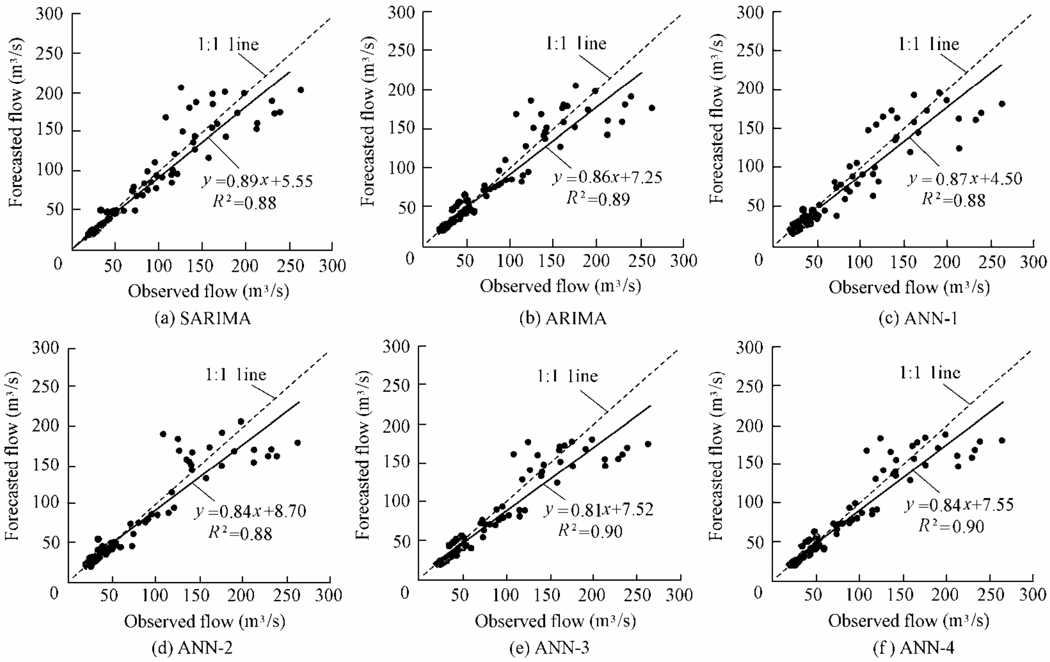

Fig. 3 Observed and forecasted monthly streamflow using different models for testing period

Fig. 4 displays the time series of one-month-ahead forecasting of ARIMA and ANN-4 models and observed monthly flow during testing period (1998 to 2005). The comparison of forecasted and observed monthly flow time series in 1998 through 2005 indicates that ANN-4 did not show a distinct forecast improvement over the simple ARIMA model. Both model forecasts showed a good match with observed monthly flow. However, the ARIMA model has a much simpler form and can be expressed in a more explicit forecasting equation form than the Jordan-Elman ANN model. The weakness of ANN models is that they are essentially black box models that cannot be expressed explicitly by a mathematical equation. Hence, in terms of parsimony and practical use, the ARIMA or SARIMA models would be preferred for forecasting monthly flow at the study site when using previous flow conditions as predictors.

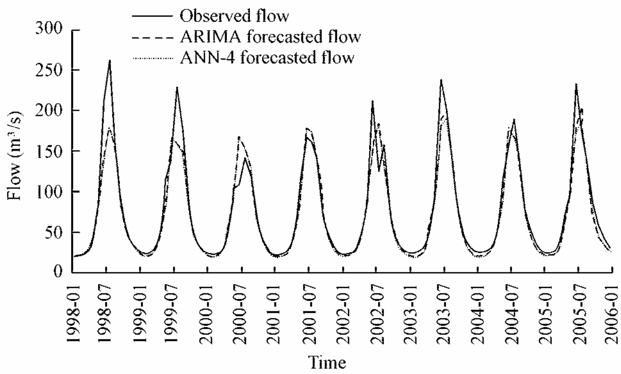

Fig. 4 Comparison of observed and forecasted flow using ARIMA and ANN-4 models for testing period

However, due to their nonlinear nature, the ANN models can easily include different predictor variables and would show better performance than the ARIMA linear model if there were more information, such as precipitation, snowpack, temperature, etc., available for the study site. In contrast, there is always a difficulty in incorporating more information in the time series models. The direct inclusion of snowpack information in the monthly time series models is particularly challenging. The snowpack has a close relationship with the streamflow processes only during some months of the year. Since the snow only exists in the winter and spring in most basins, it is difficult to obtain a continuous snowpack time series that could be included in the time series modeling. Hence, when there are more predictors available, the ANN models may be the best modeling option for monthly streamflow forecasting.

5 Conclusions

Monthly streamflow forecasting is of vital importance to decision-making in water management. This study applied time series models and Jordan-Elman ANN models in forecasting monthly flow at the Kalabeili gaging station on the Kizil River in Xinjiang, China. The performances of one-month-ahead model forecasting for the testing period from 1998 to 2005 were compared using the ARIMA, SARIMA, and Jordan-Elman ANN models with previous flow conditions as inputs. No significant difference in model performance was observed for one-month-ahead forecasting using the time series and Jordan-Elman ANN models. The ARIMA model performed similarly to Jordan-Elman ANN models when using previous flow conditions as predictors. Additionally, the ARIMA modeling process is straightforward and reflects the stochasticity of streamflow processes, expressing them simply and explicitly in mathematical equations, which is not possible in neural networks models. Hence, ARIMA models are the reasonable choice for one-month-ahead flow forecasting at the study site for better water and environmental management.

The main reason for the similar performance of time series models and ANN models at the study site may be due to the fact that only previous flows were used as model predictors. The inclusion of other predicting variables such as snowpack, precipitation, and temperature information in monthly time series modeling is challenging. In contrast, the ANN models can easily incorporate different predicting variables and would have shown better performance than ARIMA linear models if more information were available for the study site. The forecasting models presented in this study are limited to applications within the study basin and with observed specific conditions. These models are yet to be tested for other basins with different characteristics. Further investigation should be conducted to identify potential predictors and build forecasting models using ANN models and/or other more advanced modeling methods, in order to improve forecasting accuracy at the study site.

References

Abrahart, R. J., and See, L. 2000. Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes*, 14(11-12), 2157-2172. [doi:10.1002/1099-1085(20000815/30)14:11/12<2157::AID-HYP57> 3.0.CO;2-S]

Abrahart, R. J., Kneale, P. E., and See, L. M. 2004. *Neural Networks for Hydrological Modeling*. Leiden: A. A. Balkema publishers.

Awadallah, A. G., and Rousselle, J. 2000. Improving forecasts of Nile flood using SST inputs in TFN model. *Journal of Hydrologic Engineering*, 5(4), 371-379. [doi:10.1061/(ASCE)1084-0699(2000)5:4(371)]

Bender, M., and Simonovic, S. 1994. Time-series modeling for long-range stream-flow forecasting. *Journal of Water Resources Planning and Management*, 120(6), 857-870. [doi:10.1061/(ASCE)0733-9496(1994) 120:6(857)]

Box, G. E. P., and Jenkins, G. M. 1976. *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.

Coulibaly, P., Anctil, F., and Bobée, B. 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology*, 230(3-4), 244-257. [doi:10.1016/ S0022-1694(00) 00214-6]

Dibike, Y. B., and Solomatine, D. P. 2001. River flow forecasting using artificial neural networks. *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere*, 26(1), 1-7. [doi:10.1016/S1464-1909(01)85005-X]

Govindaraju, R. S., and Rao, A. R. 2000. *Artificial Neural Networks in Hydrology*. Dordrecht: Kluwer Academic Publishers.

Hipel, K. W., and McLeod, A. I. 1994. *Time Series Modeling of Water Rresources and Environmental Systems*. Amsterdam: Elsevier Science.

Hsu, K., Gupta, H. V., and Sorooshian, S. 1995. Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research*, 31(10), 2517-2530. [doi:10.1029/95WR01955]

Huang, W., Xu, B., and Chan-Hilton, A. 2004. Forecasting flows in Apalachicola River using neural networks. *Hydrological Processes*, 18(13), 2545-2564. [doi:10.1002/hyp.1492]

Karamouz, M., and Zahraie, B. 2004. Seasonal streamflow forecasting using snow budget and El Niño-southern oscillation climate signals: Application to the Salt River Basin in Arizona. *Journal of Hydrologic Engineering*, 9(6), 523-533. [doi:10.1061/(ASCE)1084-0699(2004)9:6(523)]

Kim, T., and Valdes, J. B. 2003. Nonlinear model for drought forecasting based on a conjunction of wavelet transforms and neural networks. *Journal of Hydrologic Engineering*, 8(6), 319-328. [doi:10.1061/ (ASCE)1084-0699(2003)8:6(319)]

Markus, M., Salas, J. D., and Shin, H. 1995. Predicting streamflows based on neural networks. *Proceedings of the First International Conference on Water Resources Engineering*, 1641-1646. New York: American

Society of Civil Engineers.
McKerchar, A. I., and Delleur, J. W. 1974. Application of seasonal parametric linear stochastic models to monthly flow data. *Water Resources Research*, 10(2), 246-255. [doi:10.1029/WR010i002p00246]
Mondal, M. S., and Wasimi, S. A. 2005. Periodic transfer function-noise model for forecasting. *Journal of Hydrologic Engineering*, 10(5), 353-362. [doi:10.1061/(ASCE)1084-0699(2005)10:5(353)]
Nash, J. E., and Sutcliffe, J. V. 1970. River flow forecasting through conceptual models: 1. A discussion of principles. *Journal of Hydrology*, 10(3), 282-290. [doi:10.1016/0022-1694(70)90255-6]
Noakes, D. J., McLeod, A. I., and Hipel, K. W. 1985. Forecasting monthly riverflow time series. *International Journal of Forecasting*, 1(2), 179-190. [doi:10.1016/0169-2070(85)90022-6]
Salas, J. D. 1992. Analysis and modeling of hydrologic time series. Maidment, D. R., ed., *Handbook of Hydrology*, 19.1-19.72. New York: McGraw-Hill.
Salas, J. D., Markus, M., and Tokar, A. S. 2000. Streamflow forecasting based on artificial neural networks. Govindaraju, R. S., and Rao, A. R., eds., *Artificial neural networks in hydrology*, 23-51. Dordrecht: Kluwer Academic Publishers.
Shamseldin, A. Y. 1997. Application of a neural network technique to rainfall-runoff modeling. *Journal of Hydrology*, 199(3-4), 272-294. [doi:10.1016/S0022-1694(96)03330-6]
Thompstone, R. M., Hipel, K. W., and Mcleod, A. I. 1985. Forecasting quarter-monthly river flow. *Journal of the American Water Resources Association*, 21(5), 731-741. [doi:10.1111/j.1752-1688.1985.tb00168.x]
Tokar, A. S. and Markus, M. 2000. Precipitation-runoff modeling using artificial neural networks and conceptual models. *Journal of Hydrologic Engineering*, 5(2), 156-161. [doi:10.1061/(ASCE)1084-0699(2000)5:2(156)]
Vandaele, W. 1983. *Applied Time Series and Box-Jenkins Models*. New York: Academic Press.
Wang, W. 2006. *Stochasticity, Nonlinearity and Forecasting of Streamflow Processes*. Amsterdam: IOS Press.
Yürekli, K., Kurunç, A., and Öztürk, F. 2005. Testing the residuals of an ARIMA model on the Cekerek Stream Watershed in Turkey. *Turkish Journal of Engineering and Environmental Sciences*, 29(2), 61-74.