

Forecasting of monthly river flow with autoregressive modeling and data-driven techniques

Özlem Terzi · Gülşah Ergin

Received: 25 June 2012 / Accepted: 29 July 2013 / Published online: 13 August 2013
© Springer-Verlag London 2013

Abstract This study was conducted by using autoregressive (AR) modeling and data-driven techniques which include gene expression programming (GEP), radial basis function network and feed-forward neural networks, and adaptive neural-based fuzzy inference system (ANFIS) techniques to forecast monthly mean flow for Kızılırmak River in Turkey. The lagged monthly river flow measurements from 1955 to 1995 were taken into consideration for development of the models. The correlation coefficient and root-mean-square error performance criteria were used for evaluating the accuracy of the developed models. When the results of developed models were compared with flow measurements using these criteria, it was shown that the AR(2) model gave the best performance among all developed models and the GEP and ANFIS models had good performance in data-driven techniques.

Keywords Autoregressive modeling · Data-driven techniques · Kızılırmak River · River flow

1 Introduction

The determination of flow by using the past measurements is required in design, plan, project, construction, maintenance, especially management of water resources, and determination of natural disasters such as flood and drought. So the studies of hydrological modeling based on the flow data measured in the past are becoming

increasingly important. The completion of missing flow data is important in case if there is a degradation of the measurement device and the land conditions are not established at the measurement station. The forecasting of the missing flow measurements with appropriate models, improving of the model performance, and obtaining of the better forecasting results provide convenience in terms of both economically and usage. Therefore, the hydrological time series models are commonly used for flow forecasting in recent years such as stochastic models and data-driven techniques. The stochastic models firstly proposed by Box and Jenkins [1] have been preferred in especially forecasting of stream flow [6]. Kişi [20] used artificial neural networks (ANN) to predict monthly flow and compared with autoregressive models (AR). He stated the ANN predictions in general are better than those found with AR(4). Yürekli and Öztürk [41] determined alternative autoregressive moving average process (ARMA) models by using the graphs of autocorrelation function (ACF) and partial autocorrelation function (PACF) for streamflow of Kelkit Stream. The plots of the ACF show that ARMA (1,0) with a constant is the best model by considering Schwarz Bayesian criterion (SBC) and error estimates. Wu and Chau [40] investigated ARMA, K-nearest neighbors (KNN), and ANN and phase space reconstruction-based artificial neural networks (ANN-PSR) models to determine the optimal approach of predicting monthly streamflow time series. They compared these models by 1-month-ahead forecast. They determined that the KNN model performs the best among the four models, but only exhibits weak superiority to ARMA.

The data-driven techniques having capability of analyzing long-time series have been preferred by many researchers in hydrology. Of data-driven techniques, artificial neural networks and the adaptive neural-based fuzzy

Ö. Terzi (✉) · G. Ergin
Suleyman Demirel University, 32260 Isparta, Turkey
e-mail: ozlemterzi@sdu.edu.tr

G. Ergin
e-mail: gulsahergin@stud.sdu.edu.tr

inference system which are computer systems developed with the aim of automatically performing capabilities have been investigated for problems in the water researches and meteorology studies such as solar radiation [9, 29], evaporation [19, 35], wind speed [28] and rainfall estimation [5, 26, 31], and river flow [8, 16, 22]. Lin and Chen [25] used the radial basis function network (RBFN) to construct a rainfall–runoff model for the parametric estimation of the network. The result shows that the RBFN can be successfully applied to build the relation of rainfall and runoff. Keskin and Taylan [18] developed flow prediction model, based on the adaptive neural-based fuzzy inference system (ANFIS) and ANN. The results show the ANFIS model is better than ANN model. The ANFIS model and its principles first proposed by Jang [17] have been successfully applied to many problems [30]. Chang and Chang [3] studied the intelligent control of a real-time reservoir operation model and found that given sufficient information to construct the fuzzy rules, the ANFIS helps to ensure more efficient reservoir operation than the classical models based on rule curve. Terzi et al. [37] proposed an alternative model for Penman evaporation estimation from a water surface by using ANFIS and showed that the ANFIS model can be used to estimate daily Penman evaporation for Lake Eğirdir.

Gene expression programming (GEP) was invented by Ferreira [10] and is the natural development of genetic algorithms and genetic programming. The researchers have investigated the applicability of GEP to problems in the field of water resources engineering [4, 13, 36]. Makkasorn et al. [27] applied genetic programming (GP) and artificial neural networks (ANNs) to short-term streamflow forecasting with global climate change implications. Savic et al. [34] developed rainfall–runoff model using GP and ANNs. Güven and Aytok [15] presented GEP to the modeling stage–discharge relationship. Whigham and Crapper [39] described the application of a grammatically based GP system to discover rainfall–runoff relationships for two vastly different catchments. Chang et al. [2] applied fuzzy theory and genetic algorithm (GA) to interpolate precipitation. Zhang et al. [42] investigated the use of GA in a sediment transport model. Reddy and Ghimire [32] used M5 model tree (MT) and GEP to predict suspended sediment loads. They stated that MT gives good performance as compared the model results to sediment rating curve and multiple linear regression.

The objectives of this study are to investigate data-driven techniques for forecasting monthly flow which include GEP, ANN (RBFN and FFNN), and ANFIS techniques and to compare their performances with AR modeling which is one of the traditional time series modeling techniques. This task is intended to be accomplished in Kızılırmak River, Turkey. These techniques are tried to forecast monthly flow

values (F_t) using the previous 1-month (F_{t-1}), 2-month (F_{t-2}), and 3-month (F_{t-3}) flow values. The correlation coefficient (R) and the root-mean-square error (RMSE) performance criteria are employed to validate all developed models.

2 The modeling techniques

A brief overview of the GEP, ANN, ANFIS, and AR modeling techniques used in forecasting monthly flow was presented here.

2.1 Gene expression programming (GEP)

Gene expression programming (GEP) is, like genetic algorithms (GAs) and genetic programming (GP), a genetic algorithm as it uses populations of individuals, selects them according to fitness, and introduces genetic variation using one or more genetic operators. The fundamental difference between the three algorithms resides in the nature of the individuals: In GAs, the individuals are linear strings of fixed length (chromosomes); in GP, the individuals are nonlinear entities of different sizes and shapes (parse trees); and in GEP, the individuals are encoded as linear strings of fixed length (the genome or chromosomes) which are afterward expressed as nonlinear entities of different sizes and shapes (i.e., simple diagram representations or expression trees) [10].

There are five major preparatory steps of genetic programming paradigm to solve a problem.

1. To identify the set of terminals to be used in the individual computer programs in the population, the terminals can be viewed as the input to the computer program being sought by GP. In turn, the output of the computer program consists of the value(s) returned by the program.
2. To determine a set of functions. The function set is arithmetic operators ($*$, $/$, $-$, $+$), mathematical functions (\sin , \cos , \log), logical expressions (IF–THEN–ELSEs), and Boolean operators (AND, OR, NOT) or any other user-defined function. The terminals and the functions are the ingredients from which the individual computer programs in the population are composed.
3. To identify a way of evaluating how good a given computer program is at solving the problem at hand.
4. To select the values of certain parameters to control the runs. This step involves control parameters which are the values of the numerical parameters and qualitative variables for controlling the run.
5. To specify the criterion for designating a result and the criterion for terminating a run [21].

The fundamental steps of the gene expression programming are schematically represented in Fig. 1. The process begins with the random generation of the chromosomes of a certain number of individuals. Then, these chromosomes are expressed, and the fitness of each individual is evaluated against a set of fitness cases. The individuals are then selected according to their fitness to reproduce with modification, leaving progeny with new traits. These new individuals are, in their turn, subjected to the same developmental process: expression of the genomes, confrontation of the selection environment, selection, and reproduction with modification. The process is repeated for a certain number of generations or until a good solution has been found [12].

The individuals of gene expression programming are encoded in linear chromosomes which are expressed or translated into expression trees (branched entities). Thus, in GEP, the genotype (the linear chromosomes) and the phenotype (the expression trees) are different entities (both structurally and functionally) that, nevertheless, work together forming an indivisible whole. In contrast to its analogous cellular gene expression, GEP is rather simple. The main players in GEP are only two: the chromosomes and the expression trees (ETs), being the latter the

expression of the genetic information encoded in the chromosomes [14].

In nature, the phenotype has multiple levels of complexity, the most complex being the organism itself. But tRNAs, proteins, ribosomes, cells, and so forth are also products of expression, and all of them are ultimately encoded in the genome. In all cases, however, the expression of the genetic information starts with transcription (the synthesis of RNA) and, for protein genes, proceeds with translation (the synthesis of proteins). In GEP, from the simplest individual to the most complex, the expression of genetic information starts with translation, the transfer of information from a gene into an ET. In contrast to nature, the expression of the genetic information in GEP is very simple. Worth emphasizing is the fact that in GEP, there is no need for transcription: The message in the gene is directly translated into an ET [10]. As the translation which is the process of information decoding (from the chromosomes to the expression trees), it includes code and rules. The genetic code of GEP is very simple: a one-to-one relationship between the symbols of the chromosome and the functions or terminals they represent in the trees. The rules are also very simple: They determine the spatial organization of the functions and terminals in the expression trees and the type of interaction between sub-expression trees. Therefore, there are two languages in GEP: the language of the genes and the language of expression trees. However, thanks to the simple rules that determine the structure of expression trees and their interactions, it is possible to infer immediately the expression tree given the sequence of a gene and vice versa. This bilingual and unequivocal system is called Karva language. For example, the mathematical expression:

$$\frac{a * b}{c} + \sqrt{d - e} \tag{1}$$

It can also be represented as an expression tree in Fig. 2 where Q represents the square root function.

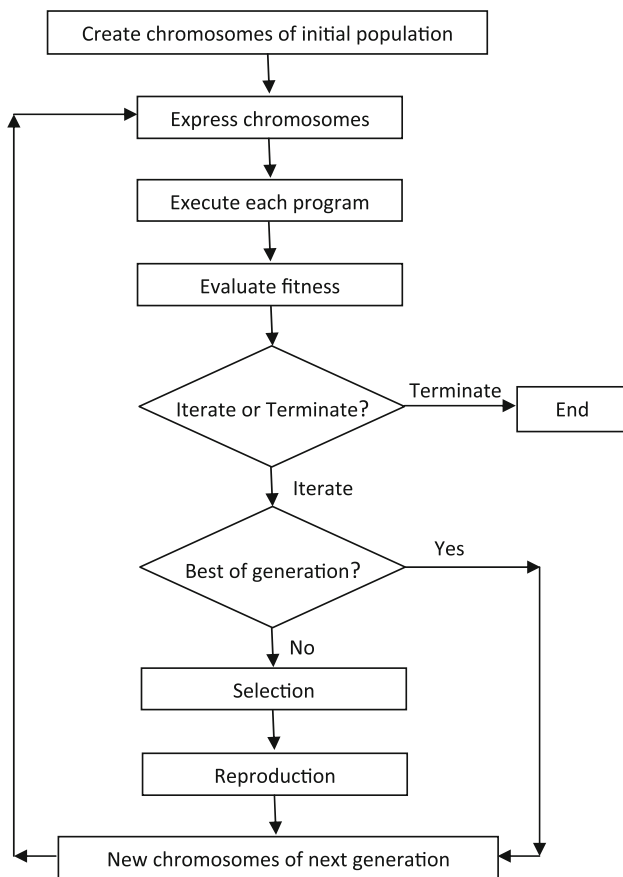


Fig. 1 The flowchart of gene expression programming [12]

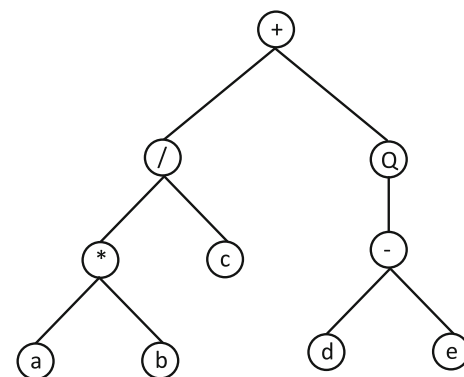


Fig. 2 An example of expression trees

This kind of diagram representation is what is called the phenotype of GEP chromosomes, and the genotype can be easily inferred from the phenotype as follows:

$$0123456789 \\ + / Q * c - abde \quad (2)$$

which is the straightforward reading of the expression tree from left to right and from top to bottom. The expression 2 is an open reading frames, starting at '+' (position 0) and terminating at 'e' (position 9). This expression is from Karva notation [11].

2.2 Artificial neural networks (ANN)

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between the elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in Fig. 3. Here, the network is adjusted, based on a comparison of the output and the target, until the sum of square differences between the target and output values becomes the minimum. Typically, many such input/target output pairs are used to train a network. Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision, and control systems. Today, neural networks can be trained to solve problems that are difficult for conventional computers or human beings [7].

2.2.1 Feed-forward neural networks (FFNN)

Feed-forward ANNs comprise a system of neurons, which are arranged in layers. Between the input and output layers,

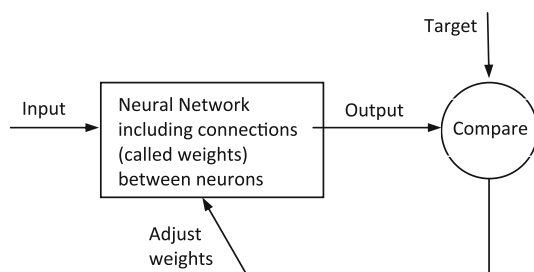


Fig. 3 Basic principle of artificial neural networks

there may be one or more hidden layers. The neurons in each layer are connected to the neurons in a subsequent layer by a weight w , which may be adjusted during training. A data pattern comprising the values x_i presented at the input layer i is propagated forward through the network toward the first hidden layer j . Each hidden neuron receives the weighted outputs $w_{ij}x_{ij}$ from the neurons in the previous layer. These are summed to produce a net value, which is then transformed to an output value upon the application of an activation function [16].

2.2.2 Radial basis function networks (RBFN)

A radial basis function network is a two-layer network whose output neurons form a linear combination of the basis functions computed by the hidden neurons. The basis functions in the hidden layer produce a localized response to the input. That is, each hidden neuron has a localized receptive field. The basis function can be viewed as the activation function in the hidden layer. The network employs a radial basis function such as the Gaussian function, which is the most popular hidden layer function. The others are thin-plate-spline, multiquadric, inverse quadratic, inverse multiquadric function, and polyharmonic spline. The basis function used in this study is a Gaussian function [23].

2.3 Adaptive neural-based fuzzy inference system (ANFIS)

Various fuzzy inference system (FIS) types are studied in the literature, and each one is characterized by consequent parameters. In this section, a brief description of ANFIS model principles is presented. The reader is referred to Chang and Chang [3] for more detail.

Fundamentally, ANFIS is a graphical network representation of Sugeno-type fuzzy systems, endowed by neural learning capabilities. The network is comprised of nodes and with specific functions, or duties, collected in layers with specific functions [38].

In order to illustrate ANFIS's representational strength, the neural fuzzy control systems are considered based on the Tagaki–Sugeno–Kang (TSK) fuzzy rules, whose consequent parts are linear combinations of their preconditions. The TSK fuzzy rules are in the following forms:

$$R^j : \text{IF } x_1 \text{ is } A_1^j \text{ AND } x_2 \text{ is } A_2^j \text{ AND } \dots \text{ AND } x_n \text{ is } A_n^j, \\ \text{THEN } y = f_j = a_0^j + a_1^j x_1 + a_2^j x_2 + \dots + a_n^j x_n \quad (3)$$

where x_i 's ($i = 1, 2, \dots, n$) are input variables, y is the output variable (solar radiation measurements), A_i^j are linguistic terms of the precondition part with membership functions $\mu_{A_i^j}(x_i)$ ($j = 1, 2, \dots, n$), and $a_i^j \in \mathbb{R}$ are

coefficients of linear equations $f_i(x_1, x_2, \dots, x_n)$. To simplify the discussion, it is necessary to focus on a specific neuro-fuzzy Controller (NFC) of the type called an adaptive neural-based fuzzy inference system (ANFIS).

Let us assume that the fuzzy control system under consideration consists of two inputs x_1 and x_2 and one output y and that the rule base contains two TSK fuzzy rules as follows:

$$R^1 : \text{IF } x_1 \text{ is } A_1^1 \text{ AND } x_2 \text{ is } A_2^1, \quad (4)$$

$$\text{THEN } y = f_1 = a_0^1 + a_1^1 x_1 + a_2^1 x_2$$

$$R^2 : \text{IF } x_1 \text{ is } A_1^2 \text{ AND } x_2 \text{ is } A_2^2, \quad (5)$$

$$\text{THEN } y = f_2 = a_0^2 + a_1^2 x_1 + a_2^2 x_2$$

In TSK fuzzy system, for given input values x_1 and x_2 , the inferred output y^* is calculated by the following formula [24]:

$$y^* = (\mu_1 f_1 + \mu_2 f_2) / (\mu_1 + \mu_2) \quad (6)$$

where μ_j are firing strengths of R^j , $j = 1, 2$, given by the equation below,

$$\mu_j = \mu_{A_1^j}(x_1) + \mu_{A_2^j}(x_2), \quad j = 1, 2 \quad (7)$$

2.4 Autoregressive modeling (AR)

Time series models have been extensively used in hydrology and water resources since the early 1960's, for modeling annual and periodic hydrologic time series. The

application of these models has been attractive in hydrology mainly because the autoregressive form has an intuitive type of time dependence (the value of variable at the present time depends on the values at previous times), and they are the simplest models to use [33].

The autoregressive model (AR) may be generally written as

$$y_t = \mu + \varphi_1(y_{t-1} - \mu) + \dots + \varphi_p(y_{t-p} - \mu) + \varepsilon_t \quad (8)$$

where y_t is the time-dependent series (variable) and ε_t is the time-independent (uncorrelated) series which is independent of y_t , and it is also normally distributed with mean zero and σ_ε^2 . The coefficients $\varphi_1, \dots, \varphi_p$ are called the autoregression coefficients. The parameter set of the model of Eq. (8) is $\{\mu, \sigma^2, \varphi_1, \dots, \varphi_p, \sigma_\varepsilon^2\}$, and it must be specified or estimated from data.

Autoregressive models with periodic parameters are those models in which part or all of their parameters vary within the year or they are periodic. These models are often referred to as periodic AR models. The time series used in hydrological studies are generally annual or monthly [33].

It is assumed that AR models are stationary and follow the normal distribution. However, it is possible to eliminate such assumptions with the data-driven models. Also, the use of data-driven methods can be easier in terms of the processing time according to the AR modeling.

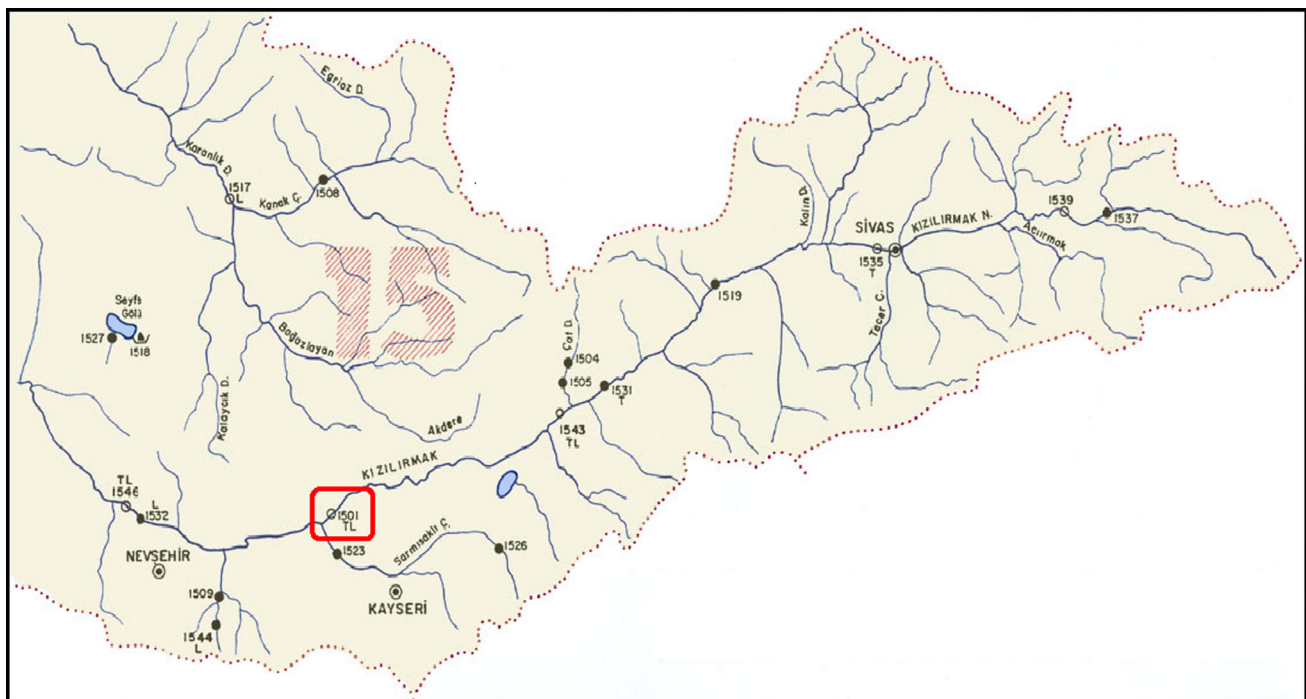


Fig. 4 The location of the Yamula (1501) station

3 Study region and data

The length of the Kızılırmak River which is the longest river in Turkey is 1,355 km. The area of the watershed is 78,646 km². The yearly average flow and rainfall are about 184 m³/s and 446.1 mm, respectively. The data used to develop model include the monthly mean flow observations between 1955 and 1995, i.e., a total of 480 months in this study. The monthly mean flow data were obtained for Yamula (1501) station of the Kızılırmak River from the General Directorate of Electrical Power Resources Survey and Development Administration. The location of the station is shown in Fig. 4. In the modeling, the training data set consisted of the years 1955–1987. The trained models were used to run a set of test data for year 1988–1995.

4 Application and results

In this study, the previous 1-month ($F_{(t-1)}$), 2-month ($F_{(t-2)}$), and 3-month ($F_{(t-3)}$) mean flow values of Kızılırmak River were used in developing models. The models were developed by using AR, GEP, FFNN, RBFN, and ANFIS techniques.

In AR modeling, monthly mean flow data set is periodic series due to shorter time interval than annual data set. Internal dependence increases because statistical characteristics in periodic series are different for another day of same process. First, the series must be fitted to normal distribution and then standardized for removing periodicity of monthly mean flow data set. It was controlled according to skewness whether or not flow values are fit to normal distribution. It was seen that flow values are not fit to normal distribution. Then, the logarithmic transformation function was applied to flow values. The transformed flow values were given in Fig. 5. As shown in Fig. 5, there is a periodicity for logarithmic flow values. The moment values

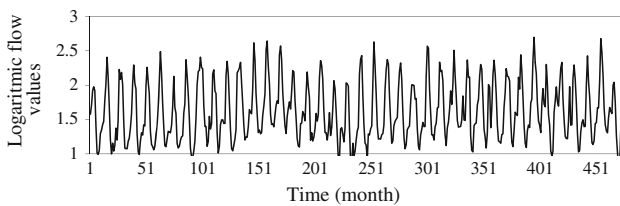


Fig. 5 Transformed flow values

Table 1 The moment values of the flow data

Basin	Flow	Periodic mean (μ_τ)	Periodic SD (σ_τ)	Transformation function ($Y_{v,\tau}$)	Average skewness (Υ)
Kızılırmak	Kızılırmak River	68.73	78.86	Log ($Z_{v,\tau}$)	0.41

(periodic mean (μ_τ), periodic standard deviation (σ_τ) and skewness) of flow data and transform function were determined and given in Table 1.

The standard normal series was obtained by applying standardization process to historical time series. The autocorrelation function (ACF) and partial autocorrelation function (PACF) of standard normal series were obtained. The upper and lower limits were determined for 95 % confidence interval (Fig. 6). It was shown that y_t series is a dependent series according to autocorrelation values.

Then, autocorrelation coefficient (δk) was calculated, and residual series was determined according to this value (Table 2). AR(1), AR(2), and AR(3) models were tested, and it was concluded that AR(2) is most appropriate model for autocorrelation values. The autocorrelation values of models were given in Fig. 7. As shown in Fig. 7, there is an agreement between autocorrelation values of AR(2) model and historical series. It was controlled that the AR(2) model provided stationarity condition in using Eq. (9).

$$u^p - \varphi_1 u^{p-1} - \varphi_2 u^{p-2} - \dots - \varphi_p = 0 \tag{9}$$

Variance of residual series σ_ε^2 was obtained according to Eq. (10),

$$\sigma_\varepsilon^2 = \frac{N\sigma^2}{(N-p)} \left(1 - \sum_{j=1}^p \varphi_j r_j\right) \tag{10}$$

where N is number of data; p is model parameter; and φ is autoregression coefficient. The Akaike’s information criterion (AIC) was used to investigate fitness of the selected model degree. AICs of AR(1), AR(2), and AR(3) models were calculated as 0.624, 0.623, and 0.705, respectively. It was confirmed that the AR(2) model having the smallest AIC is appropriate.

The synthetic series was generated for AR(2) model. Initially, the 25 random series were produced. The synthetic series were obtained by using AR(2) model and residual of the historical data. Hence, the mean, standard deviation, and ACF of the synthetic series with in 95 % confidence interval were calculated and compared with historical series (Figs. 8, 9, 10). As shown these figures, there is agreement between the historical and synthetic series. It was observed that the statistics of the synthetic series is within 95 % confidence interval for the selected AR(2) model. This situation indicates suitability of AR(2) model for predicting flow.

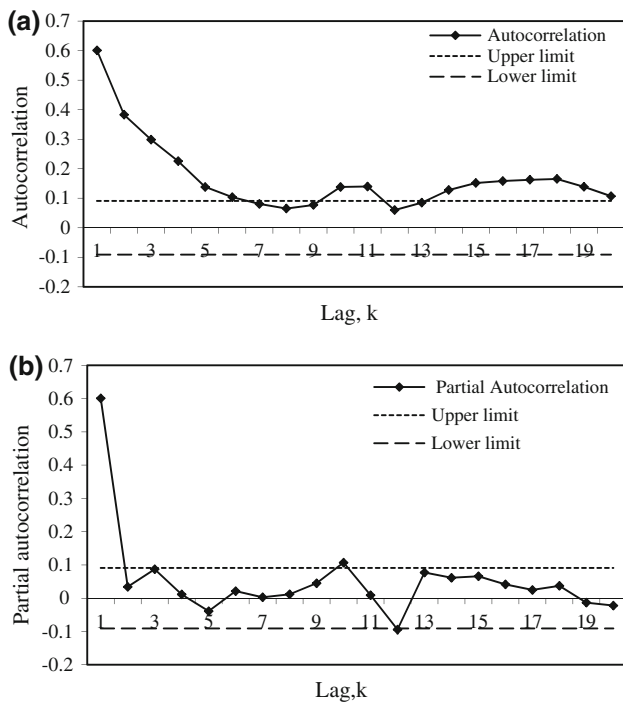


Fig. 6 The 95 % confidence interval **a** autocorrelation and **b** partial autocorrelation values

Table 2 Autocorrelation coefficients of AR models

AR(1)	AR(2)		AR(3)		
ϕ_1	ϕ_1	ϕ_2	ϕ_1	ϕ_2	ϕ_3
0.601	0.580	0.031	0.252	-0.024	0.094

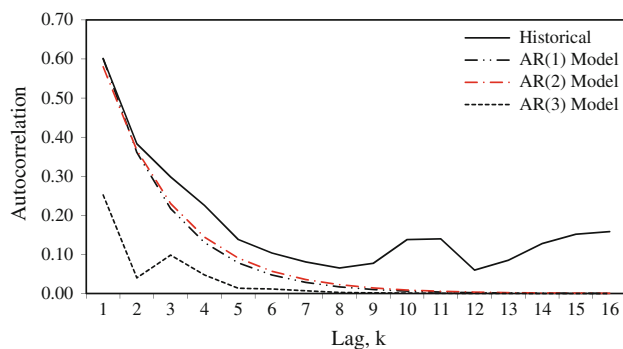


Fig. 7 The relationship ACFs of historical and AR(1), AR(2), and AR(3) models

In GEP, there are five major steps, and the first is to choose the fitness function. For this study, the *R*-square-based fitness function was selected. This kind of fitness function is very useful as it is usually interested in finding a model with a high value of *R*-square. The second step is to choose the set of terminals *T* and the set of functions *F*. The terminal set consisted, obviously, of the independent variable, giving $T = \{F_{(t-1)}, F_{(t-2)}, F_{(t-3)},$ where *F* is the

monthly mean flow data of Kızılırmak River}. The various arithmetic operators were used, $F = \{+, -, *, /, \text{power}, \sqrt{}, e^x, \ln(x), \log(x), 10^x\}$ in this study. The third step is to choose the structural organization of chromosomes, namely the length of the head and the number of genes: The length of the head equals to 8, and the number of genes per chromosome equals to 3 in this study. The fourth step is to choose the kind of linking function. In this problem, the sub-expression trees were linked by addition. And finally, the fifth major step in preparing to use gene expression programming is to choose the set of genetic operators and their rates. The genetic operators developed to introduce genetic diversity in GP populations always produce valid

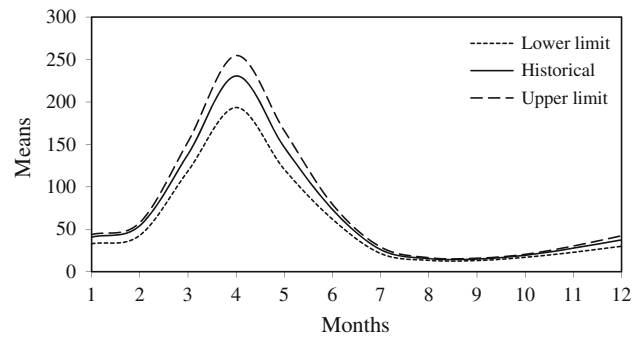


Fig. 8 The mean values for synthetic series generated by AR(2)

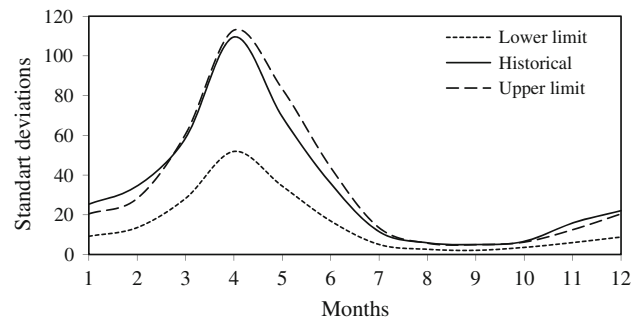


Fig. 9 Standard deviations for synthetic series generated by AR(2)

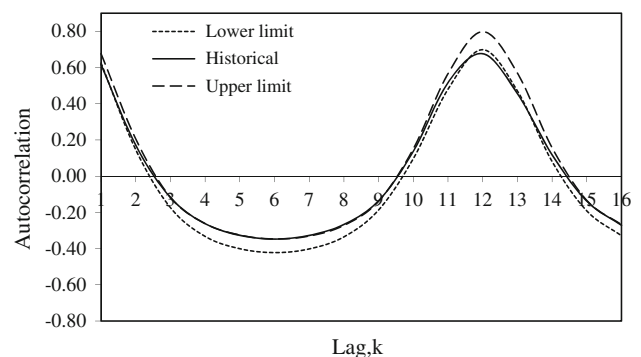


Fig. 10 The correlogram of historical and synthetic series generated by AR(2)

Table 3 Parameters of the GEP models

Number of chromosomes	50
Number of genes	3
Linking function	+
Head size	8
Mutation rate	0.044
One-point recombination rate	0.3
Two-point recombination rate	0.3
Gene recombination rate	0.1
Gene transposition rate	0.1

expression trees. Variation in the population is introduced by applying one or more genetic operators, i.e., crossover, mutation, and rotation to selected chromosomes. The genetic operators used in this study were given in Table 3.

In ANN modeling, feed-forward neural networks (FFNN) and radial basis function network (RBFN) were used for forecasting monthly flow. Prior to execution of the model, standardization was done according to the following expression such that all data values fall between 0 and 1.

$$F = (F_i - F_{\min}) / (F_{\max} - F_{\min}) \quad (11)$$

where F is the standardized value of the F_i , F_{\max} , and F_{\min} are the maximum and minimum values in all observation sequence [35]. In RBFN, the models were developed to forecast monthly flow using the same input combinations. The various spread constants were tested for each RBFN model. In FFNN, FFNN(i, j, k) indicates a network architecture with i, j , and k neurons in input, hidden, and output layers, respectively. Herein, i runs 1, 2, and 3; j assume values of 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12, whereas $k = 1$ is adopted in order to decide about the best FFNN model alternative. The numbers of neuron in hidden layer were determined using a trial-and error-method by considering the performance criteria for testing data set. The hyperbolic tangent sigmoid, logarithmic sigmoid, and linear activation functions were tried for hidden and output layers in modeling. The appropriate activation function was determined as the tangent sigmoid function after trial-and-error. The stopping criterion was employed 1,000 epochs for training, because the variation of error was too small after this epoch. The learning rate and momentum are the parameters that affect the speed of the convergence of the back-propagation algorithm. A learning rate of 0.001 and momentum 0.1 were fixed for selected network after training, and the model selection was completed for training set. The trained networks were used to run testing set.

In ANFIS modeling, each variable may have several values (in terms of rules), and each rule includes several parameters of membership functions. For instance, if each

Table 4 The R and RMSE values of the developed models using $F_{(t-1)}$ parameter

Model	Training set		Testing set	
	RMSE (m ³ /sn)	R	RMSE (m ³ /sn)	R
AR(1)	41.595	0.838	39.330	0.813
GEP	56.742	0.668	49.952	0.671
FFNN	57.098	0.663	50.937	0.665
RBFN	56.268	0.675	51.085	0.652
ANFIS	56.740	0.668	49.968	0.666

Table 5 The R and RMSE values of the developed models using $F_{(t-1)}$ and $F_{(t-2)}$ parameters

Model	Training set		Testing set	
	RMSE (m ³ /sn)	R	RMSE (m ³ /sn)	R
AR(2)	41.644	0.837	34.601	0.842
GEP	47.887	0.777	62.164	0.721
FFNN	48.402	0.772	64.341	0.695
RBFN	51.733	0.733	67.296	0.661
ANFIS	46.534	0.791	62.331	0.717

Table 6 The R and RMSE values of the developed models using $F_{(t-1)}$, $F_{(t-2)}$, and $F_{(t-3)}$ parameters

Model	Training set		Testing set	
	RMSE (m ³ /sn)	R	RMSE (m ³ /sn)	R
AR(3)	42.627	0.829	55.243	0.793
GEP	47.337	0.785	62.344	0.722
FFNN	48.439	0.773	64.475	0.695
RBFN	50.038	0.756	66.216	0.676
ANFIS	45.528	0.803	62.821	0.713

variable has three rules and each rule includes three parameters, then there are 45 [5 (variables) \times 3 (rules) \times 3 (parameters)] parameters needed to be determined in layer 2. The sole reason for having three memberships for each variable is due to the reduction in the number of rule base alternatives. Also, the ANFIS trains these membership functions according to data in layer 3, these rules will generate 3⁵ nodes, and there are 1,458 (3⁵ \times 6) parameters undetermined within the defuzzification process in layer 5. In this study, to establish the rule base relationship between the input and output variables, subtractive fuzzy clustering was used. In this study, hybrid and back-propagation optimization methods were tested, and the highest R and the lowest RMSE values were obtained by using back-propagation method.

Fig. 11 The scatter diagrams between the AR(2) model versus monthly flow measurements for **a** training set and **b** testing set

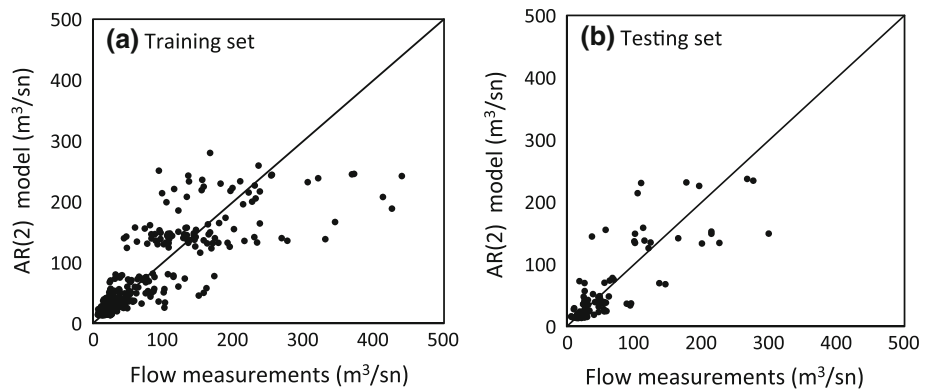
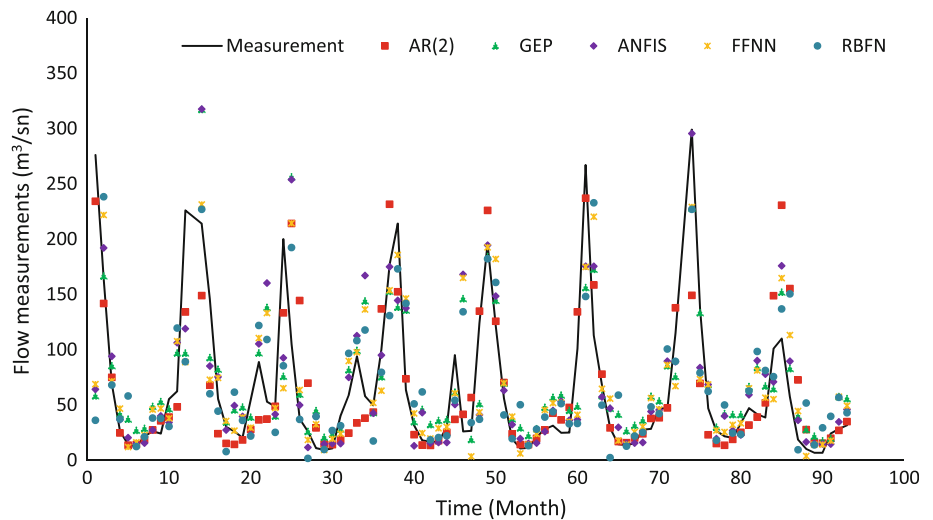


Fig. 12 Time series of forecasted and measured monthly flow values for testing set



In this paper, the same training and testing sets are used for comparing of all the above-developed models. The correlation coefficient (R) and the root-mean-square error (RMSE) performance criteria are employed to evaluate the performances of the developed models. These criteria are given for three input combinations (1) $F_{(t-1)}$, (2) $F_{(t-1)}$, $F_{(t-2)}$, and (3) $F_{(t-1)}$, $F_{(t-2)}$, $F_{(t-3)}$ in Tables 4, 5, and 6, respectively.

It can be observed from Tables 4, 5, and 6 that AR modeling performed the best R values within AR, GEP, FFNN, RBFN, and ANFIS techniques for training and testing sets. The GEP and ANFIS models have higher performance than other data-driven models developed in this study. In processing of AR modeling given above, there is not good agreement between the AR(3) model and historical flow series, and it was shown that RMSE criterion of AR(3) model is 55.243 for testing set. The AR(2) model has the best agreement with historical series compared to AR(1) and AR(3) models, and the highest R (0.842) and lowest RMSE (34.601) criteria for testing set. Also, this situation was generally supported by GEP, FFNN, RBFN, and ANFIS models developed by using

$F_{(t-1)}$ and $F_{(t-2)}$ parameters. The results of the AR(2) model were plotted against monthly flow measurements for training and testing sets in Fig. 11. The results of the all models developed by using $F_{(t-1)}$, $F_{(t-2)}$ parameters together with the monthly flow measurements for testing set were presented in Fig. 12. As seen from Fig. 12, the AR(2) model provided much closer values to the observed flow values than the other models.

5 Conclusions

This study investigated the ability of AR modeling and GEP, FFNN, RBFN, and ANFIS data-driven techniques to forecast monthly mean flow values. The proposed methods were applied to Kızılırmak River, in Turkey which is used in irrigation, in power generation, and as drinking water. The various models were developed by using lagged monthly mean flow values obtained from Yamula (1501) station of Kızılırmak River. The comparison results indicated that AR models gave better performance than the data-driven models. This is because the AR model is a

univariate model. It was shown that the AR(2) model had the best correlation coefficient in the AR models. The GEP and ANFIS models obtained better results than the other data-driven models except AR(2). The performance of the developed models suggested that the flow could be forecasted using AR approaches. Finally, AR(2) model can be used for forecasting flow in which measurement system has failed or to forecast missing monthly flow data in hydrological modeling studies. Also, it is important to underline the high computing speed gained by data-driven approaches in comparison with stochastic models. The performance of the data-driven models can be improved by using flow data of another station and rainfall data as input parameters in future studies.

References

- Box GEP, Jenkins GM (1970) Times series analysis forecasting and control. Holden-Day, San Francisco
- Chang CL, Lo SL, Yu SL (2005) Applying fuzzy theory and genetic algorithm to interpolate precipitation. *J Hydrol* 314:92–104
- Chang L-C, Chang F-J (2001) Intelligent control for modelling of real-time reservoir operation. *Hydrol Process* 15:1621–1634
- Cheng CT, Ou CP, Chau KW (2002) Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration. *J Hydrol* 268:72–86
- Chiang YM, Chang FJ, Jou BJD, Lin PF (2007) Dynamic ANN for precipitation estimation and forecasting from radar observations. *J Hydrol* 334:250–261
- Çobaner M, Çetin M, Yurtal R (2005) Investigation of the deterministic and stochastic properties of the river flows. *Çukurova Univ J Eng Archit Fac* 20(1):129–138 (In Turkish)
- Demuth H, Beale M (2001) Neural network toolbox user's guide—version 4. The Mathworks Inc, Natick, USA, pp 840
- Dibike YB, Solomatine DP (2001) River flow forecasting using artificial neural networks. *Phys Chem Earth B* 26:1–7
- Dorvlo ASS, Jervase JA, Al-Lawati A (2002) Solar radiation estimation using artificial neural networks. *Appl Energy* 71:307–319
- Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. *Complex Syst* 13(2):87–129
- Ferreira C (2002) Gene expression programming in problem solving. In: Roy R, Ovaska S, Furuhashi T, Hoffman F (eds) *Soft computing and industry-recent applications*. Springer, Berlin, pp 635–654
- Ferreira C (2006) Gene-expression programming: Mathematical modeling by an artificial intelligence. Springer, Berlin
- Ghorbani MA, Khatibi R, Aytek A, Makarynsky O, Shiri J (2010) Sea water level forecasting using genetic programming and comparing the performance with Artificial Neural Networks. *Comput Geosci* 36:620–627
- Grosan C, Abraham A (2006) Evolving computer programs for knowledge discovery. *Int J Syst Manag* 4(2):7–24
- Güven A, Aytek A (2009) New approach for stage–discharge relationship: gene-expression programming. *J Hydrol Eng* 14(8):812–820
- Imrie CE, Durucan S, Korre A (2000) River flow prediction using artificial neural networks: generalization beyond the calibration range. *J Hydrol* 233:138–153
- Jang JSR (1992) Self-learning fuzzy controllers based on temporal back propagation. *IEEE Trans Neural Networks* 3(5):714–723
- Keskin ME, Taylan ED (2009) Artificial models for inter-basin flow prediction in southern Turkey. *J Hydrol Eng* 14(7):752–758
- Keskin ME, Terzi Ö (2006) Artificial neural network models of daily pan evaporation. *J Hydrol Eng* 11(1):65–70
- Kişi Ö (2004) River flow modeling using artificial neural networks. *J Hydrol Eng* 9(1):60–63
- Koza JR (1992) Genetic programming on the programming of computers by means of natural selection. A Bradford book. The MIT Press, Cambridge
- Kumar DN, Raju KS, Satish T (2004) River flow forecasting using recurrent neural networks. *Water Resour Manag* 18:143–161
- Fu L (1994) Neural networks in computer intelligence. McGraw-Hill International Editions, New York
- Lin CT, Lee CSG (1995) Neural fuzzy systems. Prentice Hall PTR 797, NJ
- Lin GF, Chen LH (2004) A non-linear rainfall-runoff model using radial basis function network. *J Hydrol* 289:1–8
- Lin GF, Wu MC (2009) A hybrid neural network model for typhoon-rainfall forecasting. *J Hydrol* 375:450–458
- Makkeasorn A, Chang NB, Zhou X (2008) Short-term streamflow forecasting with global climate change implications—a comparative study between genetic programming and neural network models. *J Hydrol* 352:336–354
- Mohandes M, Rehman S, Halawani TO (1998) A neural networks approach for wind speed prediction. *Renew Energy* 13:345–354
- Mohandes M, Balghonaim A, Kassas M, Rehman S, Halawani TO (2000) Use of radial basis functions for estimating monthly mean daily solar radiation. *Sol Energy* 68:161–168
- Nayak PC, Sudheer KP, Rangan DM, Ramasastri KS (2004) A neuro-fuzzy computing technique for modeling hydrological time series. *J Hydrol* 291(1–2):52–66
- Ramirez MCV, Velhob HFC, Ferreira NJ (2005) Artificial neural network technique for rainfall forecasting applied to the São Paulo region. *J Hydrol* 301:146–162
- Reddy MJ, Ghimire BNS (2009) Use of model tree and gene expression programming to predict the suspended sediment load in rivers. *J Intell Syst* 18(3):211–228
- Salas JD, Delleur JW, Yevjevich V, Lane WL (1980) Applied modelling of hydrological time series. Water Resources Publications, Littleton
- Savic DA, Walters GA, Davidson JW (1999) A genetic programming approach to rainfall-runoff modelling. *Water Resour Manag* 13:219–231
- Sudheer KP, Gosain AK, Mohana Rangan D, Saheb SM (2002) Modelling evaporation using an artificial neural network algorithm. *Hydrol Process* 16:3189–3202
- Teegavarapu RSV, Tufail M, Ormsbee L (2009) Optimal functional forms for estimation of missing precipitation data. *J Hydrol* 374:106–115
- Terzi Ö, Keskin ME, Taylan D (2006) Estimating evaporation using ANFIS. *J Irrig Drain Eng* 132(5):503–507
- Tsoukalas LH, Uhrig RE (1997) Fuzzy and neural approaches in engineering. A Wiley-Interscience Publications. John Wiley & Sons Inc., New York, p 587
- Whigham PA, Crapper PF (2001) Modeling rainfall-runoff using genetic programming. *Math Comput Model* 33:707–721
- Wu CL, Chau KW (2010) Data-driven models for monthly streamflow time series prediction. *Eng Appl Artif Intell* 23:1350–1367
- Yürekli K, Öztürk F (2003) Stochastic modeling of annual maximum and minimum streamflow of Kelkit stream. *Water Int* 28(4):433–441
- Zhang FX, Wai WHO, Jiang YW (2010) Jiang Prediction of sediment transportation in deep bay (Hong Kong) using genetic algorithm. *J Hydrodyn* 22(5):599–604