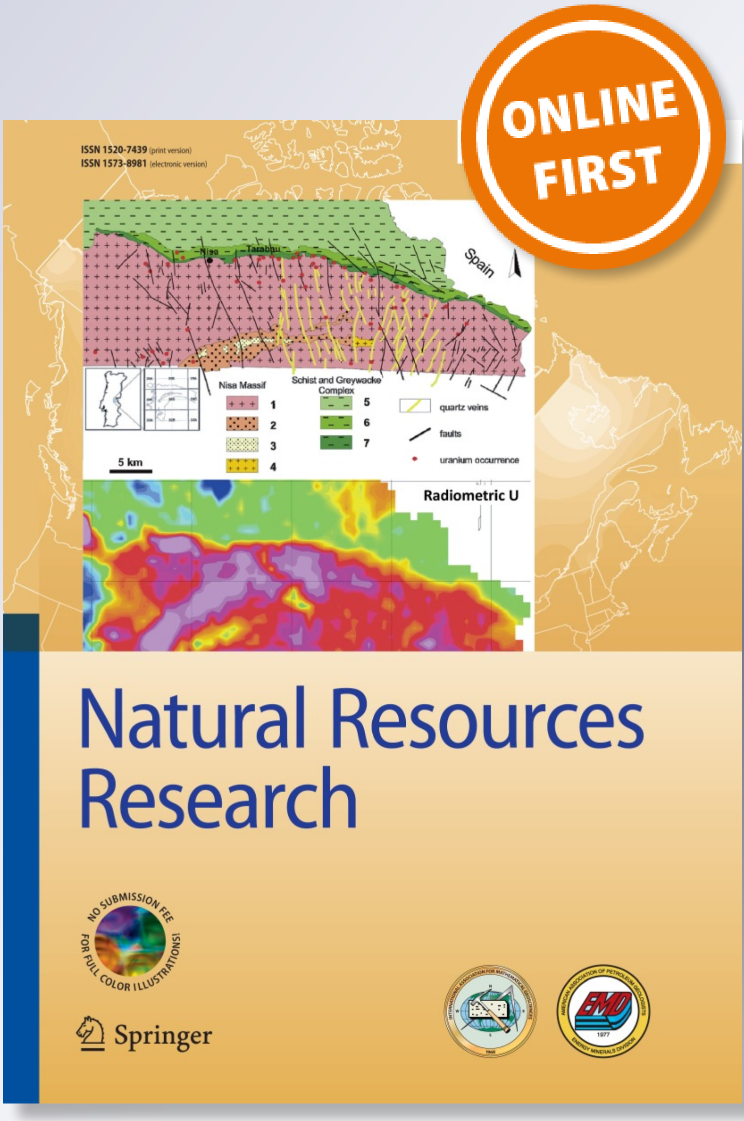
Comparison of Volterra Model and Artificial Neural Networks for Rainfall–Runoff Simulation

Mahsa Hasanpour Kashani, Mohammad Ali Ghorbani, Yagob Dinpashoh & Sedaghat Shahmorad

Natural Resources Research
formerly `Nonrenewable Resources'

ISSN 1520-7439

Nat Resour Res
DOI 10.1007/s11053-014-9235-y



Springer

Natural Resources Research (© 2014)
DOI: 10.1007/s11053-014-9235-y

Comparison of Volterra Model and Artificial Neural Networks for Rainfall–Runoff Simulation

Mahsa Hasanpour Kashani,[1,3] Mohammad Ali Ghorbani,[1] Yagob Dinpashoh,[1] and Sedaghat Shahmorad[2]

Received 13 December 2013; accepted 3 April 2014

This study evaluates the performances of two distinct linear and non-linear models for simulating non-linear rainfall–runoff processes and their applications to flood forecasting in the Navrood River basin, Iran. Due to the excellent capacity of the artificial neural networks [multilayer perceptron (MLP)] and Volterra model, these models were used to approximate arbitrary non-linear rainfall–runoff processes. The MLP model was trained using two different training algorithms. The Volterra model was applied as a linear model [the first-order Volterra (FOV) model] and solved using the traditional ordinary least-square (OLS) method. Storm events within the Navrood River basin were used to verify the suitability of the two models. The models' performances were evaluated and compared using five performance criteria namely coefficient of efficiency, root mean square error, error of total volume, relative error of peak discharge, and error of time for peak to arrive. Results indicated that the non-linear MLP models outperform the linear FOV model. The latter was ineffective because of the non-linearity of the rainfall–runoff process. Moreover, the OLS method is inefficient when the FOV model has many parameters that must be estimated.

KEY WORDS: Artificial neural network, rainfall–runoff process, simulation, Volterra model.

INTRODUCTION

A model of the rainfall–runoff (R–R) relationship is an essential component in the evaluation of water resource projects. This relationship is known to be highly non-linear and complex due to large spatial and temporal variability of watershed characteristics, temporal and spatial patterns of precipitation, and the number of input variables involved in the model. Based on the degree of representation of the involved physical processes, R–R models are classified as black-box models, conceptual models, and physically based distributed models (Napiorkowski 1986).

Artificial intelligence (AI) techniques are black-box and data-driven bottom-up modeling approaches without any prior assumptions about the model structure. These models are intelligent in the sense that they use a proportion of the data points from the time series to identify their inherent structure and use the remaining data points for validation of predictions. Artificial neural networks (ANNs) are AI techniques that have gained significant attention in recent years and play an important role in hydrology nowadays due to their ability to provide better solutions for modeling complex systems that are poorly described or understood. An ANN learns about its environment through an iterative procedure known as training that adjusts the parameters (weights) of the network. An early work

[1]Department of Water Engineering, University of Tabriz, Tabriz, Iran.
[2]Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran.
[3]To whom correspondence should be addressed; e-mail: mahsakashani2003@yahoo.com

Published online: 19 April 2014

by Halff et al. (1993) involved a three-layer feed-forward ANN using rainfall hyetographs as input and hydrograph as output. Zhu and Fujitha (1994) then compared the performance of fuzzy reasoning in R–R modeling and a feed-forward ANN model in predicting a 3-h lead runoff. Subsequently, a number of studies have employed ANN for R–R modeling (e.g., Smith and Eli 1995; Minns and Hall 1996; Tokar and Johnson 1999; Abrahart and See 2000; Dibike and Solomatine 2001; Sudheer et al. 2002; De Vos and Rientjes 2005; Garbrecht 2006; Kalteh 2008; Suhaimi and Bustami 2009). Many studies have proven further that ANNs can outperform traditional statistical R–R techniques (e.g., Hsu et al. 1995; Shamseldin 1997; Sajikumar and Thandaveswara 1999; Tokar and Johnson 1999; Thirumalaiah and Deo 2000; Toth et al. 2000). In contrast to the application of ANNs, the application of functional series requires the kernel to be evaluated by means of techniques such as the Brandstetter–Amorocho methods (Amorocho and Brandstetter 1971) and the least-squares method. However, the functional series can be calibrated adaptively; although the extra effort in doing so may not be worthy in terms of results. Moreover, the application of ANN in R–R modeling does not require any a priori assumption regarding the processes involved. The advantages of an ANN compared to conventional models are discussed in detail by French et al. (1992).

Conceptual models or physically based models simulate R–R process mainly based on the meteorological and physiographical characteristics of watershed. It is usually difficult for engineers to apply these models as they have insufficient data or knowledge on hydrologic system processes (Chang et al. 2007). To resolve these problems, most hydrologists have adopted the black-box models, which often avoid addressing the mentioned problems. Hence, the Volterra integral series (VIS) and AI techniques, as black-box models, have been introduced (Sajikumar and Thandaveswara 1999; Khatibi et al. 2011). The Volterra model is a general mathematical model for a non-linear black-box system that produces a single output from a serial input (Muftuoglu 1991). Volterra filters have been successfully used in modeling numerous physical applications, especially in signal processing and system identification. Identification involves the determination of an unknown system based on input–output information in an uncertain environment. The identification of non-linear systems depends on an accurate description of such systems. According to Singh (1988), Amorocho and Orlob (1961) were probably the first investigators to have exploited the VIS by introducing the functional series models (FSM) for the analysis of hydrologic systems. A detailed discussion on FSM, as applied to hydrologic system, is available in Amorocho (1973). The application of FSM to R–R process is widely reported in the literature (Muftuoglu 1984; Xia 1991; Sajikumar and Thandaveswara 1999).

Diskin and Boneh (1972) described the non-linear relationship between excess rainfall and direct surface runoff by a Volterra series. Papazafiriou (1976) simulated R–R process using open time-invariant linear and second-order non-linear Volterra models with memory. Napiorkowski (1986) used a model in the form of a second-order approximation of a cascade of non-linear reservoirs to model flow in open channels and surface runoff systems. Such a model was equivalent to the first two terms of the Volterra series. Labat et al. (1999) applied the first- and second-order Volterra convolutions and a new non-linear threshold model based on the frequency distribution of inter-annual mean daily runoff to two karstic watersheds in the French Pyrenees Mountains, using long sequences of rainfall and spring outflow data at two different sampling rates (daily and semi-hourly). Sajikumar and Thandaveswara (1999) applied an ANN paradigm, known as the temporal back propagation neural network (TBP-NN), as a monthly R–R model. The performance of this model in a “scarce data” scenario was compared with Volterra-type FSM. Chou (2007) proposed a wavelet-based efficient modeling of non-linear R–R processes and its application to flood forecasting in a river basin. A discrete wavelet transform was used first to decompose and compress the Volterra kernels, and then Kalman filters were utilized to estimate online compressed wavelet coefficients of the Volterra kernels, and thus model the time-varying non-linear R–R processes. Maheswaran and Khosa (2012) combined wavelet decomposition with Volterra models to forecast one month ahead of stream flow and compared it to wavelet-based linear models, linear regression models, and other non-linear approaches such as the coupled wavelets/ANN-based models. Kamruzzaman et al. (2013) compared the Volterra model and other time series models for stream flow that use lagged rainfall as an exogenous variable, with models that use a small subset of discrete wavelet coefficients of lagged rainfall with cross product and quadratic terms of the wavelet coefficients in three catchments in the Murray Darling Basin.

Although the R–R process is often non-linear, some researchers including Papazafiriou (1976), Labat et al. (1999), and Harun et al. (2002) have compared performances of linear and non-linear models for R–R simulation. The aims of this study are to evaluate the capability of the linear Volterra model (first-order) for modeling the non-linear R–R process, and compare its performance with the non-linear ANN model. The rest of this article is organized as follows. The next section describes the structure of the Volterra model in a discrete time system after having been identified by the conventional matrix method in the time domain. Then description of the ANN is presented. A case study of application of the methods to a watershed in Iran is then presented. Finally, results are described and discussed, and conclusions are drawn.

MATERIALS AND METHODS

Volterra Model

The Volterra series offers an important general representation of a non-linear system. The advantage of Volterra filters is that the output is linearly related to the Volterra kernels. Accordingly, given input u (effective rainfall) and output y (direct runoff hydrograph) sequences, the Volterra kernels can be determined by solving some linear equations. The discrete form of the Volterra model is written as (Chou 2007)

$$y_k = \sum_{j=1}^{m_1} h(j)u_{k-j} + \sum_{i=1}^{m_2}\sum_{j=1}^{m_2} g(i,j)u_{k-i}u_{k-j} + \cdots + \varepsilon_k = y_{1k} + y_{2k} + \cdots + \varepsilon_k, \quad (1)$$

where $h(j)$ and $g(i, j)$ represent the first-order (linear) and second-order (non-linear) kernels of the Volterra filters, respectively, which act as the weighting coefficients of the ordered terms. The number of kernels depends on the desired precision, on system characteristics, and on the set of bounded inputs considered (Goulart and Burt 2012). m_1 and m_2 represent the length of data in the linear and non-linear parts of the Volterra series, respectively. Note that the length of the input u should be greater than $2m_1$ and $2m_2$ (Napiórkowski and Strupczewski 1984). y_{1k} and y_{2k} represent the Volterra model output generated by the linear and non-linear parts of the Volterra series, respectively; ε_k is the output-additive random noise with zero mean.

The major drawback of the Volterra model is the exponential growth of its number of parameters with the order and the memory of the model, which causes severe problems particularly when the modeled system is strongly non-linear (Goulart and Burt 2012). So, an infinite Volterra series is not possible to implement in a practical study; hence, a truncated Volterra model is used. Since in some cases the linear models yield fairly good results, this investigation evaluates just the first part of the Volterra model and adopts a linear first-order Volterra (FOV) model of the non-linear R–R process. The linear Volterra model is defined as (Chou 2007)

$$y_k = \sum_{j=1}^{m_1} h(j)u_{k-j} + \varepsilon_k. \quad (2)$$

Note that the FOV model (Eq. 2) is the same as the equation of the unit hydrograph (UH) method in R–R modeling (Singh 1988).

One of the major tasks in Volterra modeling of non-linear systems is to determine the Volterra kernels in the time domain. In the literature, many methods exist for determination of the Volterra kernels such as the spline functions, regularization method, quasi-solution method, selection method (such as direct optimization of ordinates, orthonormal polynomials (e.g., of Laguerre or Meixner type), and conceptualization method), least-squares method, recursive least-squares method, and different intelligent techniques (Napiórkowski and Strupczewski 1984; Napiórkowski and Kundzewicz 1986; Chou 2007). In this study, the least-squares method is applied to identify the linear Volterra model and its performance is examined.

Assume that the system to be modeled has the following discrete convolution structure (Nikolaou and Mantha 2000):

$$y_k = \boldsymbol{u}_{1k}^{\mathrm{T}}\boldsymbol{h} + \varepsilon_k, \quad (3)$$

where $\boldsymbol{u}_{1k} = [u_{k-1}, u_{k-2} \ldots u_{k-m_1}]^{\mathrm{T}}$. Equation (3) is in the standard linear regression form. Therefore, the ordinary least-squares (OLS) method can be used to yield the unknown parameter $\boldsymbol{h} = [h_1, h_2 \ldots h_{m_1}]^{\mathrm{T}}$. The OLS method involved solving the convolution equation by producing the least-squares error between the observed y_k and simulated $\hat{y}_k$ outputs. The objective function F is defined as (Singh 1988)

$$F = \sum_{k=1}^{N} (y_k - \hat{y}_k)^2. \quad (4)$$

The minimization of the objective function F leads to an estimator of the kernel (Nikolaou and Vuthandam 1998):

$$\hat{\boldsymbol{h}} = [\boldsymbol{u}_{1k}\boldsymbol{u}_{1k}^T]^{-1}[\boldsymbol{u}_{1k}y_k], \quad (5)$$

where $\boldsymbol{u}_{1k}^{\mathrm{T}}$ denotes the transpose of $\boldsymbol{u}_{1k}$.

Artificial Neural Network

ANNs are parallel information processing systems consisting of a set of neurons or nodes arranged in layers and when weighted inputs are used, these nodes provide suitable conversion functions. Any layer consists of pre-designated neurons and each neural network includes one or more of these interconnected layers. Further information on ANNs can be found in Haykin (1999).

The type of ANN used in this study is a multilayer feed-forward perceptron (MLP) trained with the use of back propagation learning algorithm. The MLP network consists of (i) input layer, I; (ii) hidden layer, H; and (iii) output layer, O. The operation of this network is such that the input layer accepts the data and the intermediate layer processes them and finally the output layer displays the outputs of the model. During the modeling stage, coefficients related to present errors in nodes are corrected through comparing the model outputs with recorded input data. Connection weights are first initialized randomly by assigning a small positive or negative random value through the following procedure:

1. Input–output patterns are selected randomly using the training data presented.
2. Actual network outputs are calculated for the current input after application to the activation function.
3. Performance measure is selected, e.g., mean square error (MSE), and the values are calculated.
4. Connection weights are adjusted to minimize the MSE.
5. Steps (2)–(4) are repeated for each pair of input–output vector in the training data, until no significant change in the MSE is detected in the system.

The final connection weights are kept fixed at the completion of training and new input patterns are presented to the network to produce the corresponding output consistent with the internal representation of the input/output mapping.

In this study, two kinds of back propagation learning algorithms are used, namely the Levenberg–Marquardt (LM) and Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms, for training the MLP networks (De Vos and Rientjes 2005). The LM algorithm is often the fastest back propagation algorithm, and has been highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms (Hagan et al. 1996; Adeloye and Munari 2006; Campisi-Pinto et al. 2012). The BFGS algorithm is the most successful algorithm in published studies among the Newton's method (MATLAB 2010), which is an alternative to the conjugate gradient methods for fast optimization. Further information on the back propagation learning algorithms can also be found in Haykin (1999).

STUDY AREA

The Navrood River watershed located in north of Iran is selected here as the study area (Fig. 1). The area of the watershed is about 274 km^2. The watershed is located between 48°35′–48°54′E and 37°36′–27°45′N. The mean annual precipitation is 852.7 mm, and the mean slope of the watershed is 31.16%. Due to the topography of this watershed, the stream flow path lines are almost steep and relatively large floods occur in the watershed, leading to environmental damages. From the Navrood River watershed, hourly flood data were collected from one gaging station located at outlet and hourly rainfall data were collected from three gaging stations within the watershed. The areal averaged rainfall was computed using the Thiessen polygon method (Singh and Chowdhury 1986). Finally, fifteen simultaneous R–R storm events within the watershed were collected for this study. Eleven of these storm events were used for calibration of the Volterra and ANN models, while the other four were used for validation of the results. Details of the calibration and validation storm events are provided in Table 1. The effective rainfall rates were computed using the Φ index for each rainfall hyetograph of the events; and the direct runoff hydrographs

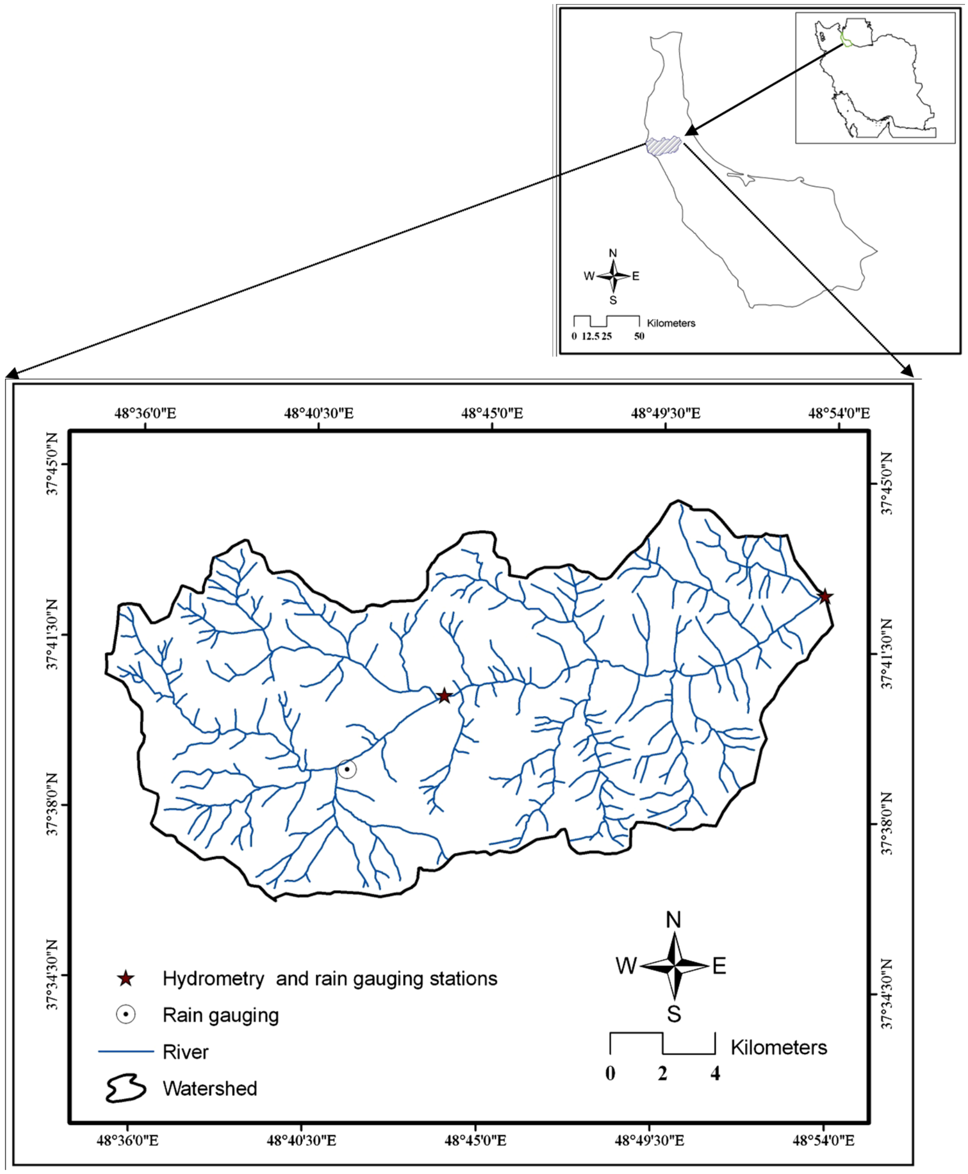


Fig. 1. The map of the study area, Navrood River watershed in northern Iran.

were obtained by separating base flow from flood hydrographs using the constant-discharge method.

COMPARISON OF MODELS

For quantitative comparison of the linear Volterra and ANN (MLP) models, the 1-h-ahead predicted results were evaluated based on the following five performance criteria:

(1) Coefficient of efficiency, *CE*, defined as (Chou 2007)

$$CE = 1 - \frac{\sum_{i=1}^{n}\left(Q_i - \hat{Q}_i\right)^2}{\sum_{i=1}^{n}\left(Q_i - \bar{Q}\right)^2}, \quad (6)$$

where $\hat{Q}_i$ (m^3/s) denotes the discharge of the simulated hydrograph for the time period i, Q_i (m^3/s) is the discharge of the observed hydrograph for the time period i, $\bar{Q}$ (m^3/s) represents the average

Table 1. Details of Storm Events Used in This Study

Events	Year	Duration (h)	Shape of Peak	Peak Discharge (m^3/s)
Calibration storm events				
1	1995	79	Single	9.93
2	1995	84	Multiple	44.40
3	1995	112	Multiple	25.55
4	1996	78	Single	28.93
5	1998	135	Single	127.62
6	1999	130	Multiple	18.5
7	1999	156	Multiple	47.8
8	1999	69	Single	13.7
9	2000	163	Multiple	26.88
10	2000	89	Single	19.55
11	2000	487	Multiple	47.52
Validation storm events				
1	2001	125	Multiple	114.64
2	2001	239	Multiple	22.11
3	2004	133	Multiple	64.40
4	2005	120	Multiple	8.00

discharge of the observed hydrograph for the time period i and n is the number of data points. The closer the CE to one, the better is the fit.

(2) Root mean square error ($RMSE$) (Sajikumar and Thandaveswara 1999)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2}{n}}. \quad (7)$$

(3) Error of total volume (EV) (Chou 2007)

$$EV(\%) = \frac{\sum_{i=1}^{n}(\hat{Q}_i - Q_i)}{\sum_{i=1}^{n} Q_i} \times 100\%. \quad (8)$$

(4) The relative error of peak discharge [EQ_p (%)] (Chou 2007)

$$EQ_p(\%) = \frac{\hat{Q}_p - Q_p}{Q_p} \times 100\%, \quad (9)$$

where $\hat{Q}_p$ (m^3/s) denotes the peak discharge of the simulated hydrograph and Q_p (m^3/s) is the peak discharge of the observed hydrograph.

(5) The error of the time for peak to arrive (ET_p) (Chou 2007)

$$ET_p = \hat{T}_p - T_p, \quad (10)$$

where $\hat{T}_p$ (hours) and T_p (hours) denote the time for the simulated and observed hydrograph peak to arrive, respectively.

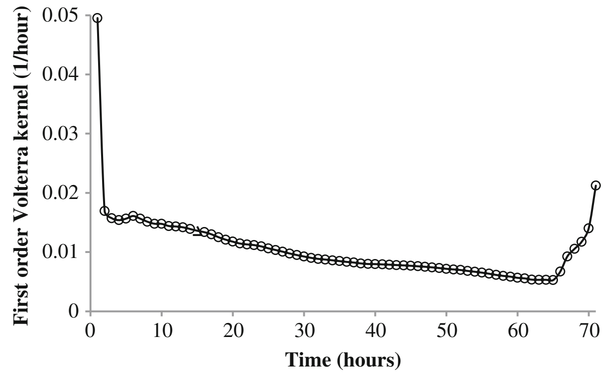


Fig. 2. The first-order kernel of the calibrated Volterra model using OLS for 11 R–R events within the Navrood River watershed.

RESULTS AND DISCUSSION

Volterra Model

In this study, the FOV model cannot be calibrated using individual storm events because it includes many unknown parameters to be estimated. Therefore, the 11 storm events within the Navrood River watershed were connected in series and treated as a single storm event for the calibration. The time lag used for computations is 1 h, which is the same as the duration of the effective rainfall pulses. The number of parameters in the linear part of the Volterra model was 71 using trial and error to minimize the objective function (F) value. Since the data length (1,582 h) should be much greater than the memory length for noise smoothing, 71 parameters were considered. The first-order kernel of the Volterra model calibrated using the OLS for the 11 events demonstrates that the OLS method does not yield a complete smooth kernel (Fig. 2). This result is in agreement with the results obtained by Chou (2007), who found that conventional methods such as OLS for identifying the Volterra model are inefficient and inaccurate because many parameters (here, 71) must be estimated.

The kernel is characterized by a sharp isolated 1-h peak, followed by a rapid decrease and an upward trend at the end. Such a trend had been seen at the end of the first-order kernel with 32 parameters obtained by Chou (2007) for a second-order Volterra model. Likewise, the first-order kernels obtained by Labat et al. (1999) using different identification methods for the linear Volterra model also were characterized by a sharp isolated one-day peak, followed by a rapid decrease and oscillations near zero. However, Papazafiriou (1976) obtained a completely smooth first-order kernel for the linear Volterra model. The kernel with 96 parameters also showed a sharp 1-h peak and then a rapid decrease.

The calibration results obtained from the FOV model for the 11 storm events show that the Volterra model underestimates the peak flows (Fig. 3). However, it predicts the time to peak more or less precisely. The performance of the FOV model solved using the OLS method applied to 11 storm events connected in series reveals that the Volterra model is not high satisfactory (Table 2). The *CE* value of the model is low and the values of the *EV*, EQ_p, and *RMSE* are high. However, the ET_p value is low and satisfactory.

The Volterra model predicts the peak flow with less accuracy (Figs. 4, 5, 6, 7). However, it estimates the peak value and time to peak successfully for the storm No. 2 (Fig. 5). This model estimates the time to peak more accurately for the storm Nos. 1 and 4 (Figs. 4, 7) than the storm No. 3 (Fig. 6). The model shows a second peak flow at the end of the hydrograph for the storm Nos. 1 and 3 (Figs. 4 and 6, respectively). This second peak can be due to the shape of the first-order kernel obtained at the calibration stage, which showed an upward trend at the end.

The average value of *CE* for the FOV is too low (Table 3). Furthermore, the average value of the *RMSE* for the mentioned model is relatively high. According to the average value of the *EV* criterion, the Volterra model does not perform satisfactorily and shows high total volume error. According to the EQ_p criterion, the FOV model is not a highly suitable model for protection and flood warning systems. However, the ET_p value of the Volterra model (with an average value of 3.5 h) indicates that it predicts the times of peak flows relatively precisely. Generally, the FOV model demonstrated poor performance in the R–R simulation.

ANN Models

The ANN (MLP) model was implemented using the *MATLAB* software and the effective rainfall and runoff data were normalized using *mapstd* function to ensure that their mean is zero and standard deviation (σ) is one (Khatibi et al. 2011):

$$X_n = \frac{X_i - \bar{X}}{\sigma}, \tag{11}$$

where $\bar{X}$ represents the average observed data, and the subscripts i and n represent the ith data and normalization, respectively. The input and output variables selected for the MLP model are the same as those for the FOV model. The MLP model was trained using two different training algorithms, LM and BFGS. The performances of the networks were evaluated for different combinations of input variables, and finally it was found that the selected input variables for the Volterra model [effective rainfall at time $(t-1)$, $(t-2)$ … $(t-71)$] are the best inputs for the networks. This means that the Volterra model has high ability in identifying the most effective input variables of the ANNs. The best fit model structure was determined for both MLP using the performance criteria and a trial-and-error method, which led to the identification of 71–1–1 structure (input layer with 71 neurons, hidden layer with 1 neuron using the tan-sigmoid function, and output layer with 1 neuron and using a linear transfer function in the output layer). Each of the optimal structures obtained for both MLPs have 74 parameters to be estimated. In this case, based on the authors' experience it seems that the size of the training data set (11 events or 1,582 sample data) is somewhat insignificant and this may cause the networks to give some errors in their estimations. This is also true for the Volterra model as a black-box model with 71 parameters.

The calibration results obtained from the MLP (LM) and MLP (BFGS) models for the 11 storm events, in which the output data of the MLPs were denormalized for comparing the networks performance with the observed data, indicate that both MLP models estimate the peak flows more or less precisely (Fig. 3). These models predicted the times to peak with high accuracy. The performances of both models are very similar and satisfactory in simulating the 11 storm events (Table 2). The values of *CE*, *RMSE*, and ET_p for both models are the same and acceptable. Therefore, it can be concluded that the MLP (LM) and MLP (BFGS) models perform almost identically. However, the MLP (LM) is more precise based on values of the *EV* and EQ_p.

Both the MLP models predict the peak flow accurately (Figs. 5, 6); however, both underestimate the peak value for the storm Nos. 1 and 4 (Figs. 4

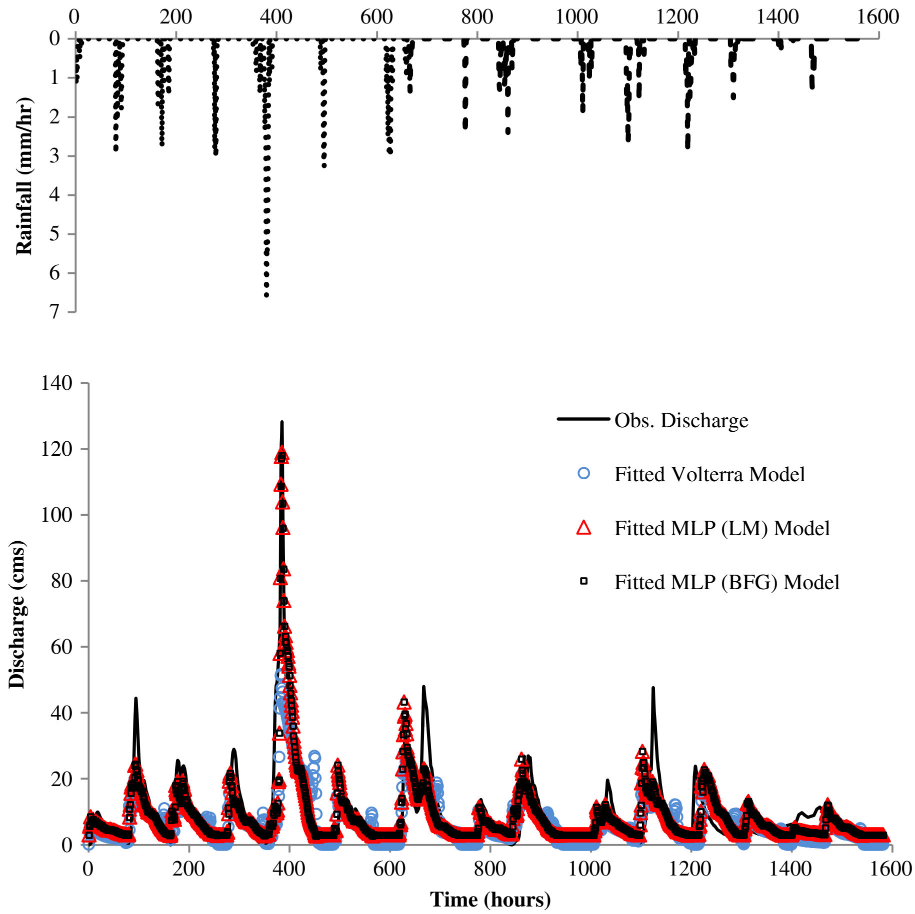


Fig. 3. The calibration results obtained from the models for the 11 events within the Navrood River watershed.

Table 2. The Performance Measures of the FOV and ANN Models at the Calibration Stage Applied to 11 Storm Events Connected in Series Within the Navrood River Watershed

Events	Data Length (h)	*EV* (%)			*CE*			*RMSE* (m^3/s)		
		FOV	MLP (LM)	MLP (BFGS)	FOV	MLP (LM)	MLP (BFGS)	FOV	MLP (LM)	MLP (BFGS)
11 in series	1,582	−24.71	−0.01	−0.66	0.51	0.71	0.71	8.33	6.38	6.38

		EQ_p (%)			ET_p (h)		
		FOV	MLP (LM)	MLP (BFGS)	FOV	MLP (LM)	MLP (BFGS)
11 in series	1,582	−59.57	−6.87	−7.60	−1	0	0

and 7, respectively). Both the MLP models estimate the time to peak more satisfactorily for the storm Nos. 1, 2, and 4 (Figs. 4, 5, and 7, respectively) than for the storm No. 3 (Fig. 6). Both of the MLP models have almost the same capability in simulating the four storm events at the validation stage (Figs. 4, 5, 6, 7).

The *CE* for the MLP models is relatively acceptable (Table 3). The average value of *RMSE* is relatively high for both the models. The average values of *EV* for the MLP (LM) and MLP (BFGS) models show their low total volume errors. The average values of EQ_p for the MLP (LM) and MLP (BFGS) models indicate that these models have high

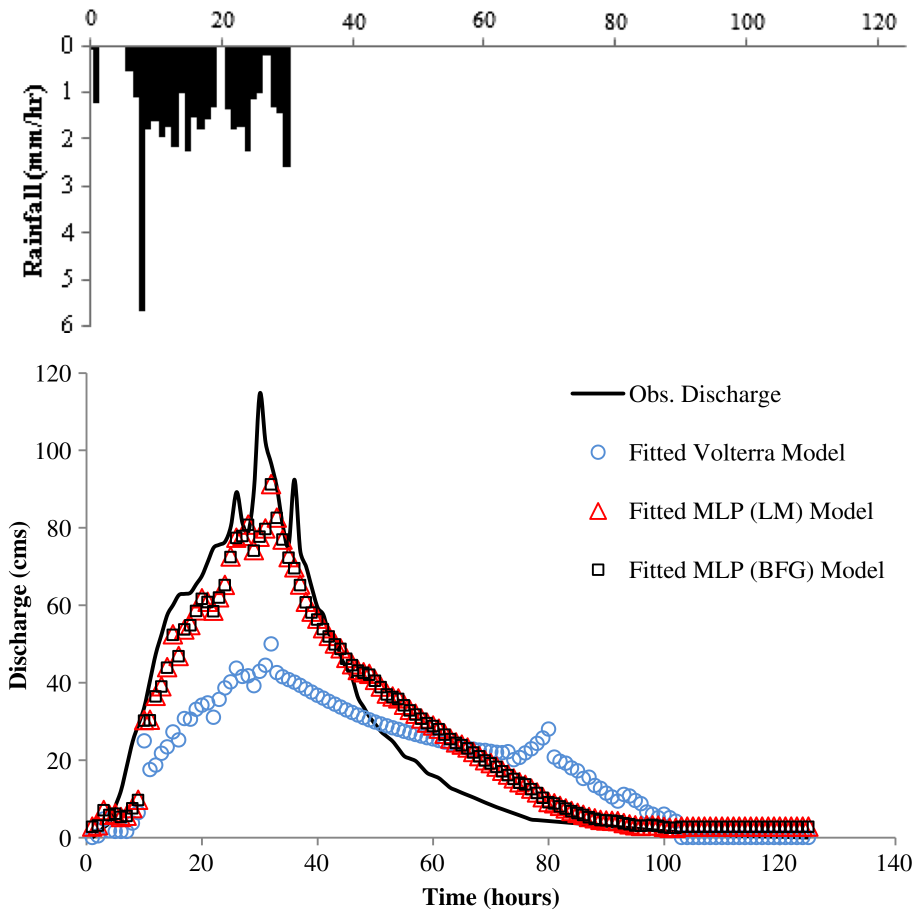


Fig. 4. The validation results of the models for the storm No. 1, 2001 within the Navrood River watershed.

ability to be applied in flood protection and warning systems. The ET_p values of the MLP models reveal that the models predict the times to peaks reasonably. It can be concluded that the MLP models generally have high ability in simulating the R–R process. This is mainly due to the non-linear nature of the ANN models. Like the results obtained by other studies (e.g., Dawson and Wilby 1998; Shamseldin et al. 2007; Lohani et al. 2011), AI models such as ANN models are very powerful in modeling more non-linear and complex systems.

Comparison of the Performance of the Models

The FOV model shows lower capability than the MLPs in predicting the peak flows and the recession limbs of the hydrographs at the calibration stage (Fig. 3). However, the accuracies of the three models in estimating the times to peak flows and the arising limbs are almost the same. In addition, the *RMSE* and ET_p values of the FOV model do not differ strongly from those of the MLPs (Table 2). However, the FOV model shows poor results based on the *EV*, *CE*, and EQ_p criteria. The performance of the MLP (LM) and MLP (BFGS) models are almost identical. However, the MLP (LM) is more powerful based on the *EV* and EQ_p criteria.

The two MLP models and the FOV model have the same performance in predicting the time to peak for storm Nos. 1–3 (Figs. 4, 5, 6). All the models have poor performance in estimating the time to peak in the case of storm No. 3 (Fig. 6). However, the MLP models are more satisfactory than the FOV model in predicting the time to peak for storm No. 4 (Fig. 7). The MLP models predict the peak flow for storm Nos. 1 and 3 (Figs. 4, 6) much better than the FOV model. For storm Nos. 2 and 4, the FOV model

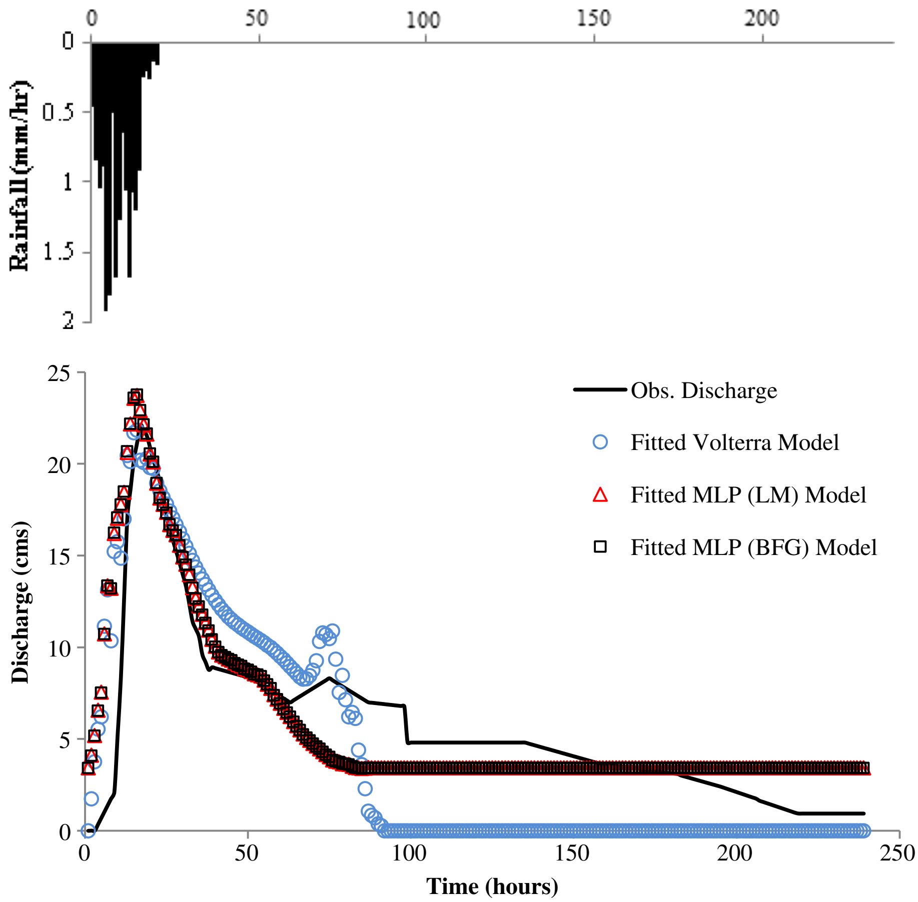


Fig. 5. The validation results of the models for the storm No. 2, 2001 within the Navrood River watershed.

shows relatively high potential in estimating the peak values. The MLP models often predict the limbs of the hydrographs more satisfactorily than the FOV. The reason for this is that the shape of the hydrographs estimated by the FOV model depends on the shape of its first-order kernel. Since the kernel obtained showed some oscillations and errors, the shape of the hydrographs and their limbs contain some errors.

The applied methods rely on rainfall observations and changes significantly with the rainfall characteristics, such as type of storm, rainfall intensity, and duration. When the type of storm is intermittent, the estimated runoff hydrograph is uneven (Figs. 4, 6). In contrast, when the type of storm is successive, the estimated runoff hydrograph is smooth (Figs. 5, 7). For almost equal rainfall duration, the stronger the rainfall intensity is the thinner and more concentrative the estimated runoff hydrograph will be. For example, the estimated runoff hydrograph in Fig. 4 is thinner and more concentrative than that in Fig. 6. In addition, the stronger the rainfall intensity is, the earlier the time to peak appears.

According to Table 3, the validation criteria values demonstrate that the *CE* for the MLPs (with an average value of 0.67) exceeds that for the FOV (with an average value of 0.38). The MLPs outperform the FOV model based on the *RMSE* (with average values of 6.19 and 10.49 m^3/s, respectively). According to the *EV* criterion, the MLP (LM) and MLP (BFGS) (with average values of 2.31 and 2.34%, respectively) perform better than the FOV model (with an average value of 24.82%). In flash flood forecasting, the accurate prediction of the peak discharge is very important for protection and flood warning systems. The average EQ_p values for the FOV, MLP (LM), and MLP (BFGS) models are 33.01, 18.93, and 19%, respectively. Therefore, the MLPs bring real improvement to the prediction of

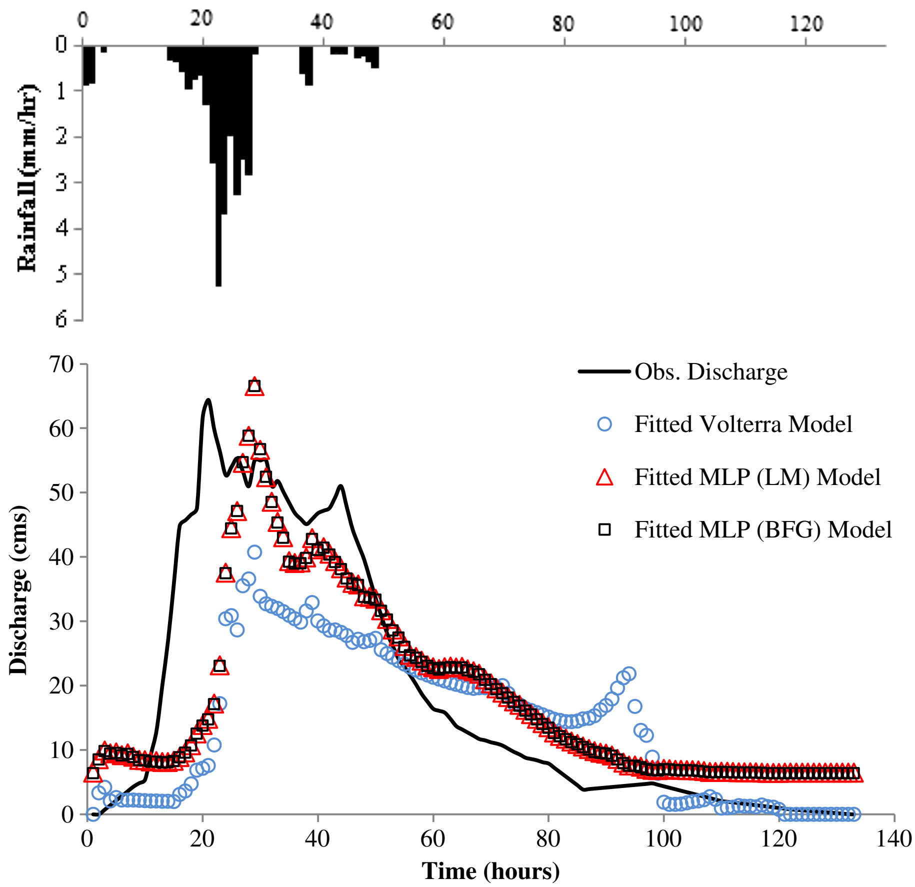


Fig. 6. The validation results of the models for the storm No. 3, 2004 within the Navrood River watershed.

peak discharge, which is important information from a practical point of view. The ET_p values indicate that the errors of the ANNs and FOV models in predicting time to peak flow are almost the same. For protection and flood warning systems, estimating the times of peak flows with less error is also very important. Comparing the two MLP models, it can be concluded that the MLP (LM) and MLP (BFGS) show similar results. However, the MLP (LM) is more satisfactory based on the EV and EQ_p values.

It has been proven that the LM and BFGS algorithms are the most successful algorithms among the ANN training algorithms. In general, the results demonstrate that the FOV model has a more loss of accuracy in the prediction than the MLPs. These findings in this study are in agreement with those in the studies of Papazafiriou (1976), Sajikumar and Thandaveswara (1999), and Labat et al. (1999). Sajikumar and Thandaveswara (1999) did not recommend the linear Volterra model for use in a basin with prominent non-linearity. The linear nature of the FOV model makes it unsuitable for simulating non-linear R–R processes. However, ANN models are quite satisfactory for R–R modeling. Sajikumar and Thandaveswara (1999) confirmed the ANN model as the most efficient of the black-box models for R–R simulation.

Here, the case study is a forested watershed in which rate of interception, infiltration, and transpiration may be high and considerable; therefore, the relationship between the rainfall and runoff may be more non-linear and complicated. Therefore, non-linear models such as ANNs can be more capable than linear models in simulating the R–R process of the watershed. However, since the R–R process is variable in time and space, it is reasonable that both the time-invariant and lumped FOV and ANN models show more or less loss of accuracy in predictions. Lumped models are those that the parameters do not vary spatially within the entire

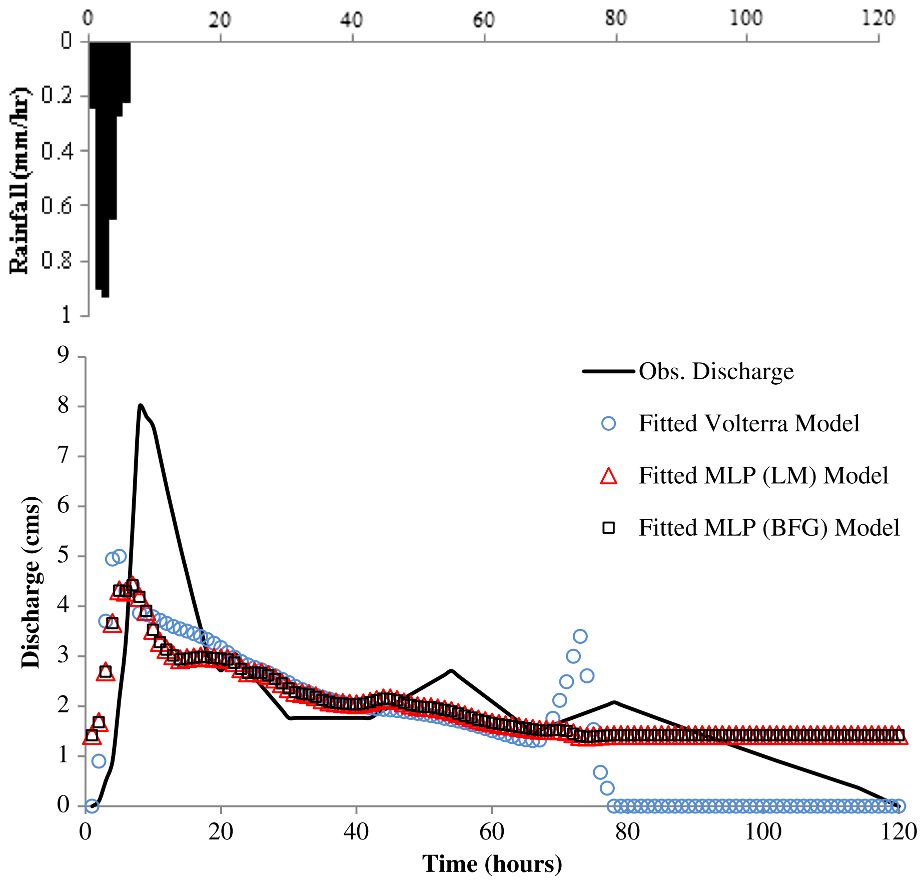


Fig. 7. The validation results of the models for the storm No. 4, 2005 within the Navrood River watershed.

Table 3. The Performance Measures of the FOV and ANN Models at the Validation Stage Applied to Four Storm Events Within the Navrood River Watershed

Events	Year	*EV* (%)			*CE*			*RMSE* (m^3/s)		
		FOV	MLP (LM)	MLP (BFGS)	FOV	MLP (LM)	MLP (BFGS)	FOV	MLP (LM)	MLP (BFGS)
1	2001	−24.82	2.75	2.33	0.51	0.91	0.91	21.46	9.05	9.03
2	2001	−24.83	4.03	3.75	0.33	0.64	0.64	3.77	2.73	2.74
3	2004	−24.84	−0.02	−0.46	0.40	0.64	0.64	15.46	11.91	11.92
4	2005	−24.81	−2.46	−2.82	0.28	0.48	0.48	1.28	1.09	1.09
Mean		24.82	2.31	2.34	0.38	0.67	0.67	10.49	6.19	6.19

Events	Year	EQ_p (%)			ET_p (h)		
		FOV	MLP (LM)	MLP (BFGS)	FOV	MLP (LM)	MLP (BFGS)
1	2001	−56.39	−20.36	−20.45	2	2	2
2	2001	−1.31	7.28	7.49	−1	−1	−1
3	2004	−36.81	3.28	3.27	8	8	8
4	2005	−37.55	−44.81	−44.78	−3	−1	−1
Mean		33.01	18.93	19.00	3.5	3	3

The last rows represent the mean of the absolute values watershed. Moreover, the selected storm events had occurred in late summer and early autumn, in which the watershed is more likely not covered by snow. However, if the watershed was snow covered during that time, the direct runoff generated by excess rainfall may contain snow melt that can cause the models to not show precise results. In this case, it was better to consider the temperature variable as model input and investigate its effect on snow melts and models performance.

Note that although the FOV model did not have high potential for simulating the R–R relationship, one cannot consider it as a totally poor model because its performance depends on some factors such as the type of its identifying method and data amount and quality. Here, since the traditional OLS method is not very efficient (Chou 2007), it may have caused the poor performance of the model. Moreover, the identification of kernels of the Volterra filters is a typical example of an ill-posed problem. It follows that one may find large errors of kernel estimates even if measurement errors are very small (Napiorkowski 1986). Since the data used in this study may contain recording errors, such errors can affect the FOV model performance.

CONCLUSIONS

This study used linear and non-linear models, the FOV and ANN [MLP (LM) and MLP (BFGS)] models, to simulate R–R processes in the Navrood River watershed located in northern Iran. Hourly rainfall and runoff rates pertaining to fifteen storm events were used to study the short-term watershed responses. The calibration results showed that the OLS method used for identifying the Volterra filter did not yield a smooth kernel. The calibration results also indicate that the ANN models outperform the FOV model. Based on five criteria at the validation stage, the average of results indicates that the ANN models outperform the Volterra model. Therefore, ANNs can be used effectively to model the non-linear R–R process because of its non-linear structure. The LM and BFGS algorithms showed almost similar potentials for training the MLP network. This investigation constitutes a useful reference in which the linear FOV model identified using the OLS method is not a very suitable model for simulating the R–R process. In order to improve the capability of the FOV model, it is recommended to use more accurate methods such as the regularization and polynomial methods for identifying the model and use more high quality for calibrating the model. Furthermore, it is recommended to apply intelligent methods such as ANNs and fuzzy inference systems (FIS) to solve the Volterra model and evaluate their potential. It is also suggested to apply the second-order Volterra model to simulate the R–R process in the Navrood River watershed and evaluate its performance. However, the second-order Volterra model may not outperform the FOV model for simulating all R–R systems. It depends also on the applied identifying methods, the amount and quality of data used, and the type of systems to be modeled. It would be desirable to evaluate the capability of the models during the spring months when rivers receive a significant snow melt contribution and coincide with the dominant rainfall season. Likewise, investigations on applying the models for higher lead time forecasting need to be accomplished for better management of water resource systems.

ACKNOWLEDGMENT

We thank Dr. Asadi for helping in collecting data. Thanks are also extended to the anonymous reviewers and editor (Dr. John Carranza) for their constructive and useful comments that helped us improve the quality of the paper.

REFERENCES

Abrahart, R. J., & 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*, 2157–2172.

Adeloye, A. J., & Munari, A. D. (2006). Artificial neural network based generalized storage–yield–reliability models using the Levenberg–Marquardt algorithm. *Journal of Hydrology, 362*, 215–230.

Amorocho, J. (1973). Non-linear analysis of hydrologic systems. In V. T. Chow (Ed.), *Advances in hydroscience* (Vol. 9). New York: Academic Press.

Amorocho, J., & Brandstetter, A. (1971). Determination of non-linear functional response functions in rainfall–runoff processes. *Water Resources Research, 7*(5), 1087–1101.

Amorocho, J., & Orlob, G. T. (1961). *Non-linear analysis of hydrologic systems*. Berkeley, CA: Water Resources Center, University of California.

Campisi-Pinto, S., Adamowski, J., & Oron, G. (2012). Forecasting urban water demand via wavelet-denoising and neural network models. Case Study: City of Syracuse, Italy. *Water Resource Management, 26*, 3539–3558.

Chang, F. J., Chang, L. C., & Wang, Y. S. (2007). Enforced self-organizing map neural networks for river flood forecasting. *Hydrological Processes, 21*, 741–749.

Chou, C. M. (2007). Efficient nonlinear modeling of rainfall–runoff process using wavelet compression. *Journal of Hydrology, 332*, 442–455.

Dawson, C. W., & Wilby, R. (1998). An artificial neural network approach to rainfall–runoff modeling. *Hydrological Sciences Journal, 43*(1), 47–66.

De Vos, N. J., & Rientjes, T. H. M. (2005). Constraints of artificial neural networks for rainfall–runoff modeling: Trade-offs in hydrological state representation and model evaluation. *Hydrology and Earth System Sciences, 9*, 111–126.

Dibike, Y. B., & Solomatine, D. P. (2001). River flow forecasting using artificial neural networks. *Physics and Chemistry of the Earth (B), 26*(1), 1–7.

Diskin, M. H., & Boneh, A. (1972). Properties of the kernels for time invariant, initially relaxed, second order, surface runoff systems. *Journal of Hydrology, 17*(1–2), 115–141.

French, M. N., Krajewski, W. F., & Cuykendall, R. R. (1992). Rainfall forecasting in space and time using a neural network. *Journal of Hydrology, 137*, 1–31.

Garbrecht, J. D. (2006). Comparison of three alternative ANN designs for monthly rainfall–runoff simulation. *Journal of Hydrologic Engineering, 11*(5), 501–505.

Goulart, J. H. M., & Burt, P. M. S. (2012). Efficient kernel computation for Volterra filter structure evaluation. *Signal Processing Letters, 19*(3), 135–138.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural network design*. Boston: PWS Publishing.

Halff, A. H., Halff, H. M., & Azmoodeh, M. (1993). Predicting runoff from rainfall using neural networks. In C. Y. Kuo (Ed.), *Engineering hydrology, proceedings of the symposium sponsored by the hydraulics division of ASCE, San Francisco, CA* (pp. 760–765). New York: ASCE.

Harun, S., Ahmat Nor, N. I., & Kassim, A. H. M. (2002). Artificial neural network model for rainfall–runoff relationship. *Jurnal Teknologi, 37*, 1–12.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). New Jersey: Prentice Hall, Upper Saddle River.

Hsu, K., Gupta, V. H., & Sorooshian, S. (1995). Artificial neural network modeling of the rainfall–runoff process. *Water Resources Research, 31*(10), 2517–2530.

Kalteh, A. M. (2008). Rainfall–runoff modelling using artificial neural networks (ANNs): modeling and understanding. *Caspian Journal of Environmental Sciences, 6*(1), 53–58.

Kamruzzaman, M., Metcalfe, A., Beecham, S. (2013). Wavelet based rainfall-stream flow models for the South-East Murray Darling Basin. Journal of Hydrologic Engineering, 10.1061/(ASCE)HE.1943-5584.0000894.

Khatibi, R., Ghorbani, M. A., Hasanpour, Kashani. M., & Kisi, O. (2011). Comparison of three artificial intelligence techniques for discharge routing. *Journal of Hydrology, 403*(3–4), 201–212.

Labat, D., Ababou, R., & Mangin, A. (1999). Linear and non-linear input/output models for karstic springflow and flood prediction at different time scales. *Stochastic Environmental Research and Risk Assessment, 13*, 337–364.

Lohani, A. K., Goel, N. K., & Bhatia, K. S. (2011). Comparative study of neural network, fuzzy logic and linear transfer function techniques in daily rainfall–runoff modeling under different input domains. *Hydrological Processes, 25*, 175–193.

Maheswaran, R., & Khosa, R. (2012). Wavelet-Volterra coupled model for monthly stream flow forecasting. *Journal of Hydrology, 450–451*, 320–335.

MATLAB. (2010). version 7.10.0 (R2010a), The MathWorks Inc., Natick, Massachusetts.

Minns, A. W., & Hall, M. J. (1996). Artificial neural network as rainfall–runoff models. *Hydrological Sciences Journal, 41*(3), 399–419.

Muftuoglu, R. F. (1984). New models for non-linear catchment analysis. *Journal of Hydrology, 73*, 335–357.

Muftuoglu, R. F. (1991). Monthly runoff generation by non-linear models. *Journal of Hydrology, 125*, 277–291.

Napiorkowski, J. J. (1986). Application of Volterra series to modeling of rainfall–runoff systems and flow in open channels. *Hydrological Sciences Journal, 31*(2), 6.

Napiórkowski, J. J., & Kundzewicz, Z. W. (1986). A discrete conceptualization of a volterra series model for surface runoff. *Water Resources Research, 22*(10), 1413–1421.

Napiórkowski, J. J., & Strupczewski, W. G. (1984). Problems involved in identification of the kernels of Volterra series. *Acta Geophysica Polonica XXXII, 4*, 375–391.

Nikolaou, M., Mantha, D. (2000). Efficient nonlinear modeling using wavelet compression. In: Progress in Systems and Control Theory, vol. 26, Boston.

Nikolaou, M., & Vuthandam, P. (1998). FIR model identification: Parsimony through kernel compression with wavelets. *AIChE Journal, 44*(1), 141–150.

Papazafiriou, Z. G. (1976). Linear and nonlinear approaches for short-term runoff estimations in time invariant open hydrologic systems. *Journal of Hydrology, 30*, 63–80.

Sajikumar, N., & Thandaveswara, B. S. (1999). A non-linear rainfall–runoff model using an artificial neural network. *Journal of Hydrology, 216*, 32–55.

Shamseldin, A. Y. (1997). Application of a neural network technique to rainfall–runoff modeling. *Journal of Hydrology, 199*, 272–294.

Shamseldin, A. Y., O'connor, K. M., & Nasr, A. E. (2007). A comparative study of three neural network forecast combination methods for simulated river flows of different rainfall–runoff models. *Hydrological Sciences Journal, 52*(5), 896–916.

Singh, V.P. (1988). Hydrologic Systems. Rainfall–Runoff Modeling, 1, Prentice Hall Englewood Cliffs, New Jersey, 480 pp.

Singh, V. P., & Chowdhury, P. K. (1986). Comparing some methods of estimating mean areal rainfall. *Water Resources Bulletin, 22*(2), 275–282.

Smith, J., & Eli, R. N. (1995). Neural Networks models of rainfall–runoff process. *Journal of Water Resources Planning and Management ASCE, 121*(6), 499–508.

Sudheer, K. P., Gosain, A. K., & Ramasastri, K. S. (2002). A data-driven algorithm for constructing artificial neural network rainfall–runoff models. *Hydrological Processes, 16*, 1325–1330.

Suhaimi, S., & Bustami, R. A. (2009). Rainfall–runoff modelling using radial basis function neural network for Sungai Tinjar catchment, Miri. *Sarawak. Journal of Civil Engineering, 1*(1), 1–7.

Thirumalaiah, K., & Deo, M. C. (2000). Hydrological forecasting using neural networks. *Journal of Hydrologic Engineering, 5*(2), 180–189.

Tokar, A. Z., & Johnson, P. A. (1999). Rainfall–runoff modeling using artificial neural network. *Journal of Hydrologic Engineering ASCE, 4*(3), 232–239.

Toth, E., Brath, A., & Montanari, A. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology, 239*, 132–147.

Xia, J. (1991). Identification of a constrained non-linear hydrological system described by Volterra functional series. *Water Resources Research, 27*(9), 2415–2420.

Zhu, M. L., & Fujitha, M. (1994). Comparison between fuzzy reasoning and neural networks methods to forecast runoff discharge. *Journal of Hydroscience and Hydraulic Engineering, 12*(2), 131–141.