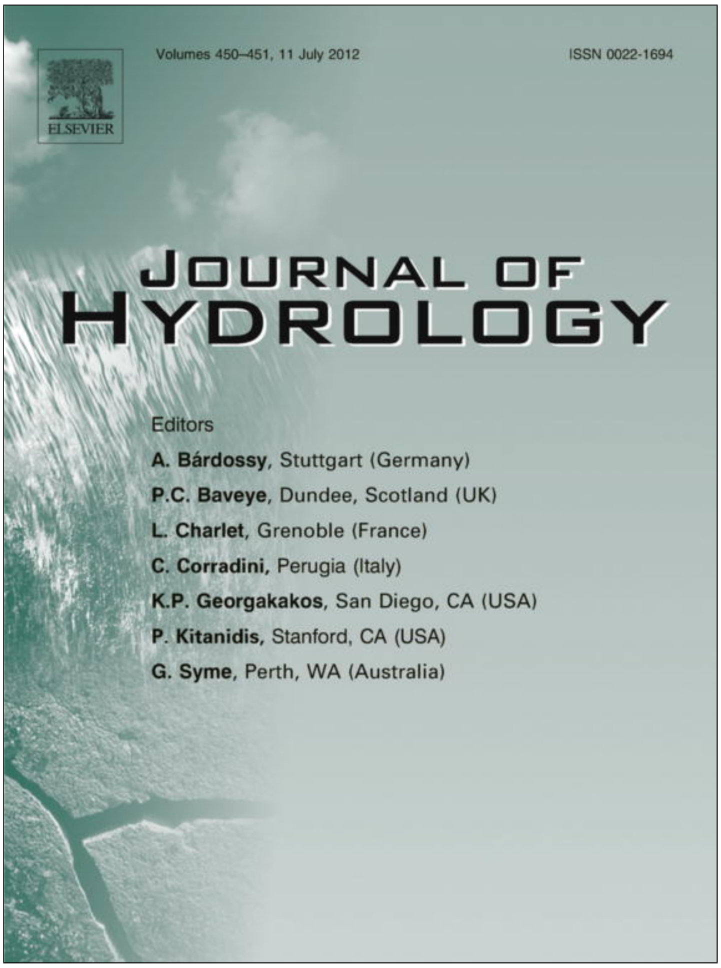
Volumes 450–451, 11 July 2012
ISSN 0022-1694
ELSEVIER
JOURNAL OF HYDROLOGY
Editors
A. Bárdossy, Stuttgart (Germany)
P.C. Baveye, Dundee, Scotland (UK)
L. Charlet, Grenoble (France)
C. Corradini, Perugia (Italy)
K.P. Georgakakos, San Diego, CA (USA)
P. Kitanidis, Stanford, CA (USA)
G. Syme, Perth, WA (Australia)

Journal of Hydrology 450–451 (2012) 293–307

Contents lists available at SciVerse ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Multi-time-step ahead daily and hourly intermittent reservoir inflow prediction by artificial intelligent techniques using lumped and distributed data

V. Jothiprakash *, R.B. Magar

Department of Civil Engineering, Indian Institute of Technology Bombay, Mumbai 400 076, India

ARTICLE INFO

Article history:
Received 15 September 2011
Received in revised form 3 February 2012
Accepted 25 April 2012
Available online 11 May 2012
This manuscript was handled by Andras Bardossy, Editor-in-Chief, with the assistance of Fi-John Chang, Associate Editor

Keywords:
Time-series models
Cause-effect models
Combined models
Daily and hourly
Lumped and distributed data
Artificial intelligent techniques

SUMMARY

In this study, artificial intelligent (AI) techniques such as artificial neural network (ANN), Adaptive neuro-fuzzy inference system (ANFIS) and Linear genetic programming (LGP) are used to predict daily and hourly multi-time-step ahead intermittent reservoir inflow. To illustrate the applicability of AI techniques, intermittent Koyna river watershed in Maharashtra, India is chosen as a case study. Based on the observed daily and hourly rainfall and reservoir inflow various types of time-series, cause-effect and combined models are developed with lumped and distributed input data. Further, the model performance was evaluated using various performance criteria. From the results, it is found that the performances of LGP models are found to be superior to ANN and ANFIS models especially in predicting the peak inflows for both daily and hourly time-step. A detailed comparison of the overall performance indicated that the combined input model (combination of rainfall and inflow) performed better in both lumped and distributed input data modelling. It was observed that the lumped input data models performed slightly better because; apart from reducing the noise in the data, the better techniques and their training approach, appropriate selection of network architecture, required inputs, and also training–testing ratios of the data set. The slight poor performance of distributed data is due to large variations and lesser number of observed values.

1. Introduction

Reservoir inflow forecast is a key component in planning, development, design, operation and maintenance of the available water resources. Inflow forecast models are useful in many water resources applications such as flood control, drought management, optimal reservoir operation, hydropower generation (Yeh, 1985). Thus, the identification of suitable generation model for future inflow is necessary for successful planning and management of water resources structures (Keskin et al., 2006). According to the use of the observational data and to the description of the physical processes large number of empirical, conceptual, physically based and data driven models has been developed and applied to map the rainfall–runoff (RR) relationship (Singh, 1988; Jothiprakash and Magar, 2009; Jothiprakash et al., 2009). However, each model has its own advantages and disadvantages (Sorooshian et al., 1993; Smith and Eli, 1995). These models suffer from problems such as identification, assimilability and uniqueness of parameter estimation (Jain and Indurthy, 2003). Some of the earliest examples of time-series models are given by Box and Jenkins (1976), and includes autoregressive (AR), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA) and autoregressive moving average with exogenous input (ARMAX) methods. All these approaches have employed conventional time-series forecasting and modelling (Thomas and Fiering, 1962; Yevjevich, 1963; Salas et al., 1980; Toth et al., 2000; Mohammadi et al., 2006) assuming that the data taken over time may have an internal structure (such as autocorrelation, trend, or seasonal variation). However, they provide only reasonable accuracy and suffer from the assumptions of stationary and linearity.

Recently artificial intelligent (AI) based data driven techniques such as artificial neural network (ANN), adaptive neuro-fuzzy inference system (ANFIS) and comparatively new technique linear genetic programming (LGP) have emerged as important tools to overcome the drawbacks of conventional modelling technique. These data driven approaches are based on extracting and re-using information implicitly contained in hydrological time-series without directly taking into account the physical laws that underline the process (de Vos and Rientjes, 2005). In the last decade, ANN has been successfully employed in modelling a wide range of hydrologic processes due to their ability to model non-linear system efficiently (Tokar and Johnson, 1999; Thirumalaiah and Deo, 2000; Chang et al., 2002; Sivakumar et al., 2002; Zhang et al., 2009; Googhari et al., 2010). Wang et al. (2006) studied the classic 'divide and conquer (DAC)' paradigm as a top-down black-box technique for the forecasting of the daily streamflows of the upper Yellow River at Tangnaihai in China. The input considered was only

* Corresponding author. Tel.: +91 022 25767302.
E-mail address: vprakash@iitb.ac.in (V. Jothiprakash).

0022-1694/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jhydrol.2012.04.045

streamflow records without employing exogenous variables of the runoff generating process such as rainfall. Three forms of hybrid ANNs were used as univariate time series models, namely, the threshold-based ANN (TANN), the cluster-based ANN (CANN), and the periodic ANN (PANN). For the purpose of comparison of forecasting efficiency, the normal multi-layer perceptron form of ANN (MLP–ANN) is selected as the baseline ANN model. Overall, among the three variations of hybrid ANNs tested, the PANN model performed best for short lead times but vanishes for longer lead times.

Kote and Jothiprakash (2009a) investigated the performance of time lagged recurrent networks (TLRN) with time delay, gamma and laguarre memory structure for predicting seasonal (June–October) reservoir inflow with a monthly time step. The ANN model results were compared with conventional ARIMA models for observed time-series of Pawana reservoir, upper Bhima river basin, India. Kote and Jothiprakash (2009b) studied the performance TLRN with AR, ARMA and ARIMA for intermittent reservoir in series. The models have been developed and applied for monthly time step for Yadgaon reservoir in Upper Bhima River basin India. Jothiprakash et al. (2010) applied ANN models for an intermittent RR process using monthly time stepped lumped and distributed data. The developed ANN models were trained with various algorithms. It was found that both lumped and distributed data ANN models performed equally better. However, from the cause-effect ANN models, it was found that the back propagation (BP) algorithm performed well for lumped data and conjugate gradient (CG) algorithm performed well for distributed data.

Although ANNs do have many attractive features, they suffer from some limitations like the difficulty in choosing an appropriate training algorithm and time-consuming effort involved in developing the structure. But there is still scope to determine the structure of the ANN network and the training algorithm to optimize specific parameters of the network. A comprehensive review of the application of ANN to hydrology can be found in ASCE Task Committee (2000a, 2000b) and (Maier and Dandy, 2000; Maier et al., 2010). To achieve robust learning from the given set of patterns, various kinds of neural networks mechanism are explored in the past. A time-delay recurrent neural network (TDRNN) and Generalized feed forward (GFF) network are adopted in the present study. Wu and Chau (2011) has predicted daily RR transformation from two different watersheds, namely, Wuxi and Chongyang in China. This study attempts to eliminate the lag effect from two aspects: modular ANN (MANN) and data pre-processing by singular spectrum analysis (SSA). Results showed that MANN does not exhibit significant advantages over ANN. However, it is demonstrated that SSA can considerably improve the performance of prediction model and eliminate the lag effect. It was recommended that the ANN RR model coupled with SSA is more promising.

Fuzzy theory appears to be extremely effective in handling dynamic, non-linear and noisy data, especially when the underlying physical relationships are not fully understood. It has also proved to be a very attractive tool enabling the modeller to investigate problems that are uncertain, empiricism and in case of vague data (Mohan and Jothiprakash, 2000; Nayak et al., 2005; Firat and Gungor, 2008). In time-series modelling, the so-called ANFIS overcomes some of the individual weaknesses of ANN and fuzzy and offers some appealing features (Nayak et al., 2004; Chen et al., 2006; Firat, 2008). Jothiprakash et al. (2009) used ANFIS models to develop lumped data RR relationship for monthly data. Intermittent runoff river system namely, Kanand river in Maharashtra state, India was taken as the case study.

A neuro-fuzzy network combines the transparency of fuzzy "if-then" rules with learning capability of neural network. This flexible network is widely recognized because of its accurate non-linear modelling, able to learn from environment (input–output), self organization of its structure and in an adaptive interactive manner (Chen et al., 2006). ANFIS has been proved to be a powerful tool for modelling numerous processes, such as RR modelling, real-time reservoir operation, flood forecasting (Chau et al., 2005; Chidthong et al. 2009; Dastorani et al., 2010; Dorum et al., 2010). However, application of ANFIS for modelling the daily and hourly RR relationship (for an intermittent system, the data having long sequence of zero) has not been reported much.

In recent years, a particular subset of genetic programming (GP) with a linear structure similar to the Deoxyribose Nucleic Acid (DNA) molecule in biological genomes namely, LGP has been emerged. LGP is a machine learning approach that evolves the programs of an imperative language or machine language instead of the traditional Koza (1992) tree-based GP expressions of a functional programming language (Brameier and Banzhaf, 2007). LGP, the extension of GP provides inherent functional input–output relationships as compared to traditional black box models and found to be more suitable in developing a physical relationship between a set of input and output data in the form of a computer program. For the last one decade, GP have been pronounced as one of the alternatives and robust methods in the field of hydrology and water resources engineering (Whigham and Crapper 2001; Drunpob et al., 2005; Makkeasorn et al., 2008; Elshorbagy et al., 2010a, 2010b). However, LGP is still under its nascent stage. Very few studies have been carried out using LGP in hydrology and water resources (Guven, 2009; Garg and Jothiprakash, 2010). Jothiprakash and Kote (2010) studied the effect of data pre-processing while developing AI based data-driven techniques, such as ANN, model trees (MT) and LGP for Pawana Reservoir in Maharashtra, India. The results showed that AI methods are powerful tools for modelling the daily time-series with appropriate data pre-processing, in spite of many zero values.

Many reservoirs in India are intermittent in nature and inflow into these reservoirs is estimated by a mass balance equation. In all these, reservoir inflow data is computed by subtracting reservoir level of the current day with the previous day reservoir level duly accounting for releases (spill, irrigation, power outlets, etc.), leakages, and evaporation losses from the functional reservoir. If there is an error due to various assumptions of estimation procedure, that error is constant for each computed data in the inflow time-series leading to global shift in the observed values. Generally time-series accounts for the fact that data taken over time may have an internal structure such as autocorrelation, trend and seasonal variation, that should be accounted and analysed before modelling the data series (Peng and Buras, 2000; Kote and Jothiprakash, 2009a).

It is to be mentioned that the data used in most of the above reported studies are from a perennial river and most of time, the data used is time-series stream flow data. Very few studies have been carried to assess the effect of using both continuous lumped and distributed data on daily and hourly basis for the same basin. The main purpose of this study is to investigate the applicability and capability of the above AI techniques to predict the multi-time-step ahead daily and hourly intermittent Koyna reservoir inflow.

2. Time-delay recurrent neural networks (TDRNNs)

Most of the ANN applications in hydrology have used a feed forward neural network, namely, the standard multi-layer perceptron (MLP) trained with the back-propagation (BP) algorithm (Coulibaly et al., 2000). However, MLP is a static and memory less network, and even though it is the most widely used to model water resource variables predictions, it often yields sub-optimal solutions. MLP model does not perform temporal processing and the input vector space does not consider the temporal relationship of the inputs (Giles et al., 1997). For more complicated temporal processing and understanding the dependence of the initial and past states, tapped delay lines (TDLs; internal time-delay operators) can be used within the MLP network. Actually, the use of these internal time-delay operators help the network to behave dynamically and leads to the time-delay neural network (TDNN), which has been used in a variety of applications (Waibel et al., 1989). TDLs and an Elman recurrent connection unit (Elman, 1990) are attached to a static MLP network add to an extended dynamic neural network known as a time-delay RNN (TDRNN). The functioning of TDRNN can be found in (Karamouz et al., 2008; Razavi and Araghinejad, 2009; Kote and Jothiprakash, 2009b; Htike and Khalifa, 2010).

3. Generalized feed forward (GFF) ANN

In many engineering problems, one-time-step-ahead prediction using ANN has been performed and reported with satisfactory results. However, one-time-step-ahead prediction may not provide enough information, especially in situation where it is desirable to understand the behaviour of multi-time-steps in the future. It is found that most of the ANN models on time-series forecasting used the standard MLP trained with BP algorithm. Even though the well-known steepest descent method is a widely used training algorithm, it often yields sub-optimal solutions and its convergence characteristic has encouraged research into faster algorithms. Generalized feed forward (GFF) neural network is extended to form Shunting Inhibitory ANNs (SIANNs). SIANNs are biologically inspired networks in which the synaptic interactions are mediated via a non-linear mechanism called shunting inhibition, which allows neurons to operate as adaptive non-linear filters. The functioning of GFF can be found in (Bouzerdoum and Mueller, 2003; Chang et al., 2004; Hung et al., 2009).

4. Adaptive neuro-fuzzy inference system (ANFIS)

Jang (1993) introduced architecture and learning procedure for the FIS that uses a neural network learning algorithm for constructing a set of fuzzy if-then rules with appropriate membership functions (MFs) from the specified input–output pairs. This procedure of developing a FIS using the framework of adaptive neural networks is called ANFIS. There are two methods that ANFIS learning employs for updating membership function parameters: (1) backpropagation for all parameters (a steepest descent method) and (2) a hybrid method consisting of backpropagation for the parameters associated with the input membership and least squares estimation for the parameters associated with the output membership functions. As a result, the training error decreases, at least locally, throughout the learning process. Therefore, the initial membership functions resemble the optimal ones, the easier it will be for the model parameter training to converge. Human expertise about the target system to be modelled may aid in setting up these initial membership function parameters in the FIS structure. A review of the application of ANFIS to hydrology can be found in (Aqil et al., 2007; Wang et al., 2009; Jothiprakash et al., 2009).

5. Linear genetic programming

In spite of various advantages of applying ANN and ANFIS technique, transferring the knowledge gained through modelling, particularly the relationship between input and output to the field engineers is rather limited. Hence, researchers are seeking other data based techniques where the knowledge can be extracted very easily. The Darwin's natural selection theory of evolution based GP, is relatively a new technique and is the member of evolutionary algorithm family (Koza, 1992). GP is an inductive form of machine learning as it evolves a computer program to perform an underlying process defined by a set of training samples (Whigham and Crapper, 2001). GP has been successfully applied to complex non-linear problems and its solution describes the input–output relationship.

In the concept of GP introduced by Koza (1992), the programs are represented as tree structures and expressed in the LISP functional programming language (Babovic and Keijzer, 2000; Brameier, 2004; Guven, 2009). Later, researchers regarded it as tree based GP (TGP) due to its tree structural solution. Recently, a subset of GP has emerged, which evolves programs in an imperative programming language (C/C++) and represent the graph-based functional structure, termed as LGP (Brameier et al., 1998). The imperative program structure of LGP identifies the non-effective instructions efficiently and executes rapidly (Brameier and Banzhaf, 2001; Foster, 2001). Moreover, in LGP, the maximum size of the program was usually restricted to avoid over-growing programs without any condition (Brameier and Banzhaf, 2001).

The name 'linear' refers to the structure of the (imperative) program representation, and does not stand for functional genetic programs. LGP represents highly non-linear solutions in this meaning (Brameier, 2004; Guven, 2009). The main advantage of LGP is its ability to produce models that build an understandable structure given that the LGP model exhibits a great potential to screen and prioritize the input variables. The various LGP parameters involved are population size, mutation rate and its different types (block mutation rate, instruction mutation rate and data mutation rate), crossover rate, homologous crossover, function set, number of demes and program size. The parameter selection will affect the model generalization capability of LGP. They were selected based on some previously suggested values (Francone, 2004) and after trial and error approach.

Ineffective code in genetic programs, referred as "intron" represents instructions without any influence on the program behaviour. Structural "introns" act as a protection that reduces the effect of variation on the effective code and allow variations to remain neutral in terms of fitness change. Because of the imperative program structure in LGP, these ineffective instructions can be identified efficiently. This allows the corresponding effective instructions to be extracted from a program during runtime. Since only effective programs are executed when testing fitness cases, evaluation is significantly accelerated. The instructions from imperative languages are restricted to operations that accept a minimum number of constants or memory variables, called registers (r) and assign the result to a destination register, e.g. $r0 = r1 + 1$. Where register $r0$ holds the final program output. LGPs can be converted into a functional representation by successive replacements of variables starting with the last effective instruction (Oltean and Grosan, 2003).

For LGP based models, Discipulus software (Francone, 2004), is used which is based on Automatic Induction of Machine code by Genetic Programming (AIMGP) approach. In order to evaluate the capabilities of proposed LGP models, various performance criterias are used between actual and predicted values. The fitness of LGP program is evaluated using mean square error (MSE).

6. Study area and data

The area selected for the present study is Koyna watershed, situated on the West-Coast of Maharashtra, India, lies between the latitude of "$17^0$00′N" to "$17^0$59′N" and longitude of "$73^0$02′E" to "$73^0$35′E". The location of the study area along with

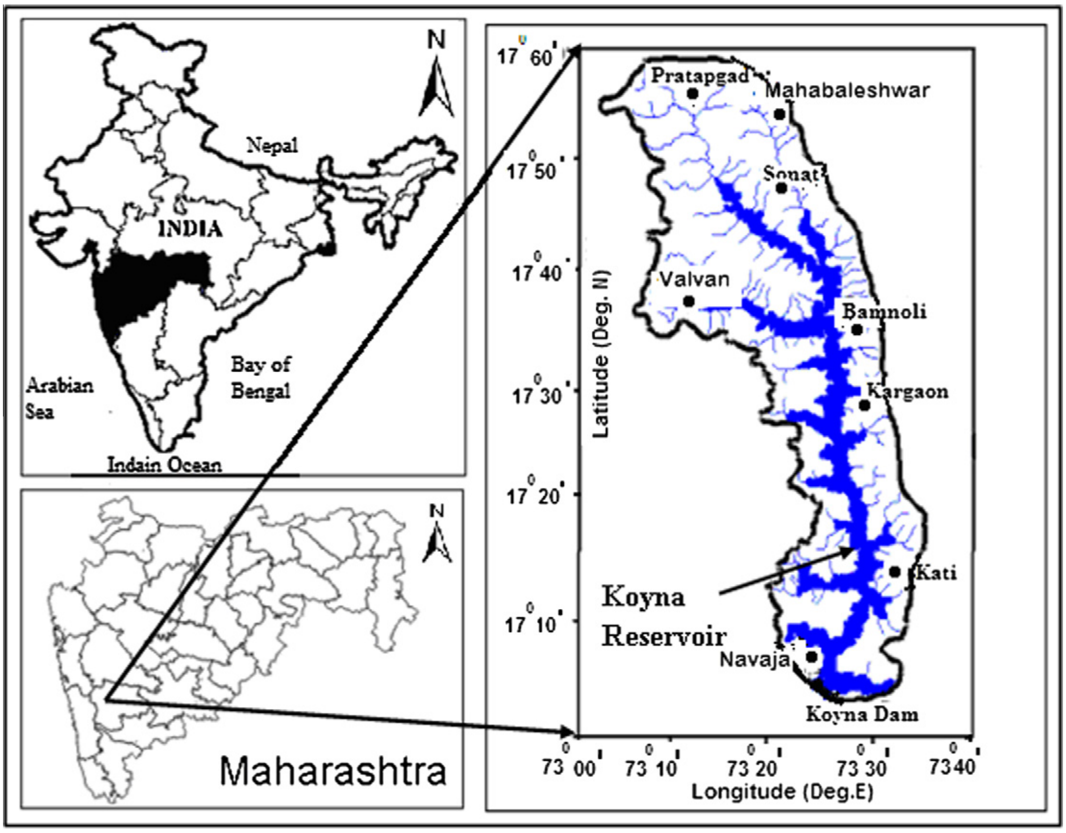

Fig. 1. Location of Koyna Watershed (Magar, 2011).

nine rain-gauge stations in Koyna watershed is shown in Fig. 1. The Koyna Dam is one among the 23,000 large dams in the world with a gross storage capacity of 2797.15×10^6 m^3. The height of the Koyna dam above foundation level is 103 m and the length of the dam at the crest is about 800 m. The Koyna project is a multi-purpose project, but primarily designed as a hydro-electric project that supplies hydro-electric power to Maharashtra state, India with an installed capacity of 1920 MW. Koyna watershed has an elongated leaf shape, about 64 km in length and about 13 km width with an area of 891.78 km^2. The watershed is bounded by hills and broadly consists of 41% forest, 49% cultivated area, 6% waste land and 4% of others (CDO, 1992). The water spread area at full reservoir level is 115.36 km^2 which is about 13% of the total catchment area. Nearly 99% of the annual rainfall in this basin occurs during south-west monsoon (June–October) and varies from 2972 mm to 6694 mm annually over the valley.

Daily rainfall of 47 years (January 1961–December 2007), hourly rainfall of four years (2005–2008) and corresponding inflow data has been collected from Koyna irrigation division office, Government of Maharashtra, India and is used in this study. Table 1 shows the details of raingauge stations and the length of the data available.

7. Model development

Historically observed inflow values represent the hydrological state of the catchment, which greatly determines a catchments response to a rainfall event. In the basin there are nine rain-gauge stations measuring the rainfall data. Hence, both rainfall (P) and inflow (Q) are considered as critical input to the model development. According to different input combinations to the models, various types of models developed in the present study are time-series models (forecast multi-time-step ahead future inflow values based on observed current and past inflow values), cause-effect models (the reservoir inflow is affected by precipitation alone over the catchment area) and combined models (output is affected by current and delayed rainfall as well as inflows) (Tokar and Johnson, 1999; Tokar and Markus, 2000; Jothiprakash et al., 2010).

7.1. Daily and hourly lumped input data models

Even though each of the nine rain-gauge station has a time-series of rainfall data, all the rainfall data (in mm) are lumped using Thiessen polygon method with respect to space and a single time-series rainfall data has been used to predict the inflow and the model is considered as lumped data models. The average daily rainfall data length is 47 years. The different lengths are taken care while averaging them spatially. The data was divided into two sets: a training set consisting of first 70% and testing set consist of the remaining 30%. In the present study, an attempt is made to develop daily and hourly time stepped relationship between the inflow at the catchment outlet (at reservoir), using average rainfall and inflow data available up to the current time 't'. Therefore, all the developed models are basically approximators of the general function

$$Q_{(t+n)} = f\{P_{obs(t)}, P_{obs(t-1)} \ldots P_{obs(t-m)}, Q_{obs(t)}, Q_{obs(t-1)} \ldots Q_{obs(t-n)}\} \quad (1)$$

$P_{obs(t)}$ and $Q_{obs(t)}$ represent observed rainfall and inflow during time period 't'. $Q_{(t+n)}$ is inflow to be predicted for the next time step. The prediction is done up to three days ahead for daily data and up to ten hours for hourly data. Shanker et al. (1996), Luk et al. (2000) and Aqil et al. (2007) reported that networks trained on transformed data achieve better performance and faster convergence in general. The present study data is found to be non-normal hence,

Table 1
Details of rain-gauge stations and the length of the daily data available (Magar, 2011).

Sr. no.	Rain-gauge station	Longitude (E)	Latitude (N)	Length of the data available
1	Mahabaleshwar	$73^0 40'21''$	$17^0 55'23''$	47 Years
2	Valvan	$73^0 35'43''$	$17^0 44'17''$	37 Years
3	Pratapgad	$73^0 34'43''$	$17^0 56'02''$	41 Years
4	Navaja	$73^0 43'24''$	$17^0 25'37''$	36 Years
5	Sonat	$73^0 42'30''$	$17^0 50'14''$	41 Years
6	Kati	$73^0 49'36''$	$17^0 29'18''$	41 Years
7	Kargaon	$73^0 76'47''$	$17^0 39'17''$	15 Years
8	Bamnoli	$73^0 45'43''$	$17^0 43'46''$	41 Years
9	Koyna	$73^0 44'28''$	$17^0 23'33''$	47 Years
10	Inflow $\times$ 10^6 m^3	$73^0 45'08''$	$17^0 25'24''$	47 Years

Table 2
Statistical properties of raw and logarithmic transformed hourly data set.

Statistical properties	Average hourly rainfall (mm)				Inflow (m^3/s)			
	Entire data set (observed) (01/06/05–0 h)–(31/10/2008–23 h)	Entire data set (transformed) (01/06/2005–0 h)–31/10/2008–23 h)	Training data set (transformed) (01/06/05–0 h)–(01/10/07–8 h)	Testing data set (transformed) (01/10/07–9 h–31/10/2008–23 h)	Entire data set (observed) (01/06/05–0 h)–(31/10/2008–23 h)	Entire data set (transformed) (01/06/2005–0 h)–31/10/2008–23 h	Training data set (transformed) (01/06/05–0 h)–(01/10/07–8 h)	Testing data set (transformed) (01/10/07–9 h–31/10/2008–23 h)
$\overline{X}$	2.621	0.133	0.152	0.100	359.18	0.905	0.882	0.961
S_x	2.982	0.161	0.168	0.0865	593.16	0.526	0.580	0.364
C_{sx}	3.712	1.101	0.910	−0.177	2.74	−0.539	−0.463	−0.384
K_x	4.193	0.320	−0.163	−1.175	3.10	−0.843	−1.184	−0.145
X_{min}	0.00	0	0	0	0.00	0	0	0
X_{max}	39.334	0.8	0.802	0.22	4976	1.85	1.848	1.785
C_v	1.845	1.211	0.802	0.86	1.65	0.581	0.657	0.073
No. of data points	14688	14688	10,281	4407	14,688	14,688	10,281	4407
No. of zeros	9379	9379	6271	3108	9546	9546	6785	2761
% Of zero	63.85	63.85	60.99	70.52	64.99	64.99	65.99	62.65

$\overline{X}$ – Mean, S_x – standard deviation, C_{sx} – skewness, K_x – kurtosis, X_{min} – minimum observed value, X_{max} – maximum observed value, C_v – coefficient of variation.

a logarithmic transformation has been used to bring the observed data to near normal distribution.

The descriptive statistics of total observed, transformed data set as well as training and testing data set for daily data can be found in literature (Magar and Jothiprakash, 2011) whereas for hourly data set are shown in Table 2. From Table 2 it can be observed that the standard deviation, skewness, kurtosis are very high in observed data and found to be greatly reduced after logarithmic transformation. From Table 2 it can also be observed that the training, testing, and entire data set (both rainfall and inflow) are statistically similar revealing that the data are from same population and will not cause over fitting difficulties. This conclusion is supported by many authors (Aytek and Alp, 2008). Based on cross-correlation, autocorrelation, partial auto correlation function twenty daily lumped data models and twelve hourly lumped data models with various input combinations have been formulated to develop the real-time rainfall-reservoir inflow relationship. As an example, sample of daily and hourly lumped input data models (two in each type) are shown in Table 3.

7.2. Daily and hourly distributed input data models

Daily and hourly-distributed input data models have been developed for establishing relationship between rainfall and inflow for Koyna watershed. Unlike lumped data model, in distributed data model, the input variables, rainfall from different stations are considered as individual inputs (used "as it is") for the model development. The rainfall from nine rain-gauge stations are valued as $P_{1(t)}$, $P_{2(t)}$, $P_{3(t)}$, $P_{4(t)}$, $P_{5(t)}$, $P_{6(t)}$, $P_{7(t)}$, $P_{8(t)}$, $P_{9(t)}$ and inflow as $Q_{(t)}$. Twenty daily data models and twelve hourly data models with various combinations of input have been formulated to develop the distributed rainfall inflow relationship. As an example sample of daily and hourly distributed data input models, (two in cause-effect and two in combined) are shown in Table 4. The general form of the distributed input data model is

$$Q_{t+n} = f\{(P_{1_{(t)}}, P_{2(t)}, \ldots, P_{9(t)}), (P_{1(t-1)}, P_{2(t-1)} \ldots P_{9(t-1)}), \ldots, (P_{1_{(t-m)}}, P_{2(t-m)}, \ldots P_{9_{(t-m)}} \ldots Q_{obs(t)}, \ldots, Q_{obs(t-1)}, \ldots, Q_{obs(t-n)})\} \quad (2)$$

8. Model performance criteria

The performance of the lumped and distributed input data models are assessed based on statistical performance criteria like correlation coefficient (R), nash sutcliffe efficiency (E), root mean square (RMSE), akaike information criteria (AIC), bayesian information criteria (BIC) (Srinivasulu and Jain, 2006; Nash and Sutcliffe, 1970). The details of performance criteria are listed in Appendix A.

9. Results and discussion

9.1. Daily lumped and distributed data ANN models (ANN)

Multi-input multi-output (MIMO) ANN architecture was selected, for capturing the complex, dynamic, and non-linear, rainfall–inflow process in the basin. The number of neurons in the input and the output layer can be specified according to the number of predictors and predictants, respectively. The output vector in the output layer is three neurons representing the inflow at 1 day,

Table 3
Model types and input combinations (daily and hourly lumped data).

Model type	Model inputs	No. of input variables	Output variables				
Daily lumped data models							
Time-series models							
DL[a] model 3	$Q_{(t-2)}$, $Q_{(t-1)}$, $Q_{(t)}$	3	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
DL model 6	$Q_{(t-5)}$, $Q_{(t-4)}$, $Q_{(t-3)}$, $Q_{(t-2)}$, $Q_{(t-1)}$, $Q_{(t)}$	6	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
Cause-effect models							
DL model 10	Current average rainfall with two antecedent average rainfall	3	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
DL model 14	Current average rainfall with six antecedent average rainfall	7	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
Combined models							
DL model 19	Current average rainfall and inflow with two antecedent average rainfall and one antecedent inflow	5	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
DL model 20	Current average rainfall and inflow with three antecedent average rainfall and two antecedent inflow	7	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
Hourly lumped data models							
Time-series models							
HL[b] Model 2	Current inflow with one antecedent inflow	2	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
HL model 3	Current inflow with two antecedent inflow	3	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
Cause-effect models							
HL model 5	Current average rainfall with one antecedent average rainfall	2	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
HL model 6	Current average rainfall with two antecedent average rainfall	3	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
Combined models							
HL model 11	Current average rainfall and inflow with one antecedent average rainfall	4	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
HL model 12	Current average rainfall and inflow with two antecedent average rainfall and one inflow	5	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$

[a] DL – Daily-lumped input data, time '*t*' is in days.
[b] HL – Hourly lumped input data, time'*t*' is in hours.

2 days and 3 days ahead. The number of neurons in the hidden layer was varied from 2 to 50 and best architecture was finalized to capture the rainfall-inflow relationship for each category of input data.

For comparison, only best models in each type (time-series, cause-effect and combined models) are selected and shown in Table 5. From Table 5 it is evident that the performances of lumped and distributed data ANN pure cause-effect models are inferior to the time-series models. The reason may be due to the difficulty in capturing the pattern in the input and output data. The time-series models have only one pattern where as the cause-effect models have two different patterns. In time-series models the input and output has same pattern however, in cause-effect models pattern of input and output is different. From Table 5 it can also be seen that distributed combined input models performed better than lumped data pure cause-effect and time-series models. However, the lumped combined input DL-ANN model 18 (3–6–3) obtained best performances during training and testing and is slightly better than distributed input data model especially for peaks and higher lags.

In fact averaging by Thiessen polygon method produced a smoothing of non-stationarities by averaging the fluctuations recorded at each rain-gauge station (Burlando et al., 1993; Toth et al., 2000). Apart from this the major reason for better performance of ANN lumped models are due to the better techniques and their training approach, appropriate selection of network architecture, required inputs, and also training–testing ratios of the data set. DL-ANN model 18 with lead period of 1 day has obtained maximum *R* (0.94), *E* (0.90) minimum *RMSE* (8.55), exhibited minimum *AIC* (11051.7) and *BIC* (11058.24) respectively during testing and appears to be a parsimonious model. Hence, DL-ANN model 18 is selected as the best model among all the ANN models. The scatter plot of the observed inflow and the multi-time-step ahead daily predicted inflow by DL-ANN model 18 during testing period is depicted in Fig. 2. The identified lumped ANN model resulted in a reasonably accurate prediction of medium inflow but not the peak inflows. The reason is that the rainfall inflow relationship may be highly non-linear at peak inflow. In ANN technique daily lumped models performed slightly better than daily distributed models, because of reduced noise in the input due to lumping the inputs. Also in daily lumped model it is only two patterns to be recognized (one input series and one output series) where as in distributed model it is ten patterns (nine rainfall series and one inflow series).

9.2. Daily lumped and distributed data ANFIS models

In the present study various daily lumped and distributed input data ANFIS models have been developed with input and output parameters same as that of ANN models. In case of ANFIS models, the number of membership function (MF) associated with each input variable is fixed by trial and error. Excess number of MFs on the input variable will increase the number of "if-then" fuzzy rules and simultaneously increases the model complexity and hence affect the model parsimony, hence numbers of MFs are varied up to four. In case of distributed data model, the complexity is further increased due to large number of input. Hence, in distributed input data case subtractive fuzzy clustering method has been used to classify the input data because of large input variables (Chang and Chang, 2006).

The parsimonious structure that resulted in minimum error and maximum efficiency during training and testing were selected as the final form of ANFIS model. Due to smoothness and concise notation, the bell membership functions are increasingly popular for specifying fuzzy sets. The advantage of bell shaped membership functions is that it has one more parameter than Gaussian membership functions, thus fuzzy set can be approached when the free parameter is tuned, and the same is adopted in this study. The statistical performances of the best performed model in each type of input like time-series ANFIS, cause-effect ANFIS, and combined ANFIS models are presented in Table 6.

Among the ANFIS models, it is found that lumped input data cause-effect and time–series models performed equally better, however, combined input data models performed better than the

Table 4
Model type and input combinations (daily and hourly distributed data).

Model type	Inputs variables	Input var.	Outputs variables				
Daily Distributed data models							
Cause-effect models							
DD[a] model 6	$P_{1(t-5)}$, $P_{1(t-4)}$, $P_{1(t-3)}$, $P_{1(t-2)}$, $P_{1(t-1)}$, $P_{1(t)}$, $P_{2(t-5)}$, $P_{2(t-4)}$, $P_{2(t-3)}$, $P_{2(t-2)}$, $P_{2(t-1)}$, $P_{2(t)}$, $P_{3(t-5)}$, $P_{3(t-4)}$, $P_{3(t-3)}$, $P_{3(t-2)}$, $P_{3(t-1)}$, $P_{3(t)}$, $P_{4(t-5)}$, $P_{4(t-4)}$, $P_{4(t-3)}$, $P_{4(t-2)}$, $P_{4(t-1)}$, $P_{4(t)}$, $P_{5(t-5)}$, $P_{5(t-4)}$, $P_{5(t-3)}$, $P_{5(t-2)}$, $P_{5(t-1)}$, $P_{5(t)}$, $P_{6(t-5)}$, $P_{6(t-4)}$, $P_{6(t-3)}$, $P_{6(t-2)}$, $P_{6(t-1)}$, $P_{6(t)}$, $P_{7(t-5)}$, $P_{7(t-4)}$, $P_{7(t-3)}$, $P_{7(t-2)}$, $P_{7(t-1)}$, $P_{7(t)}$, $P_{8(t-5)}$, $P_{8(t-4)}$ $P_{8(t-3)}$, $P_{8(t-2)}$, $P_{8(t-1)}$, $P_{8(t)}$, $P_{9(t-5)}$, $P_{9(t-4)}$, $P_{9(t-3)}$, $P_{9(t-2)}$, $P_{9(t-1)}$, $P_{9(t)}$	54	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
DD model 7	Current rainfall from nine raingauge station with six antecedent rainfall from nine raingauge station	63	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
Combined models							
DD model 19	Current rainfall from nine raingauge station and current inflow with three antecedent rainfall from nine raingauge station and two antecedent inflows	39	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
DD model 20	Current rainfall from nine raingauge station and current inflow with four antecedent rainfall from nine raingauge station and two antecedent inflows	48	$Q_{(t+1)}$	$Q_{(t+2)}$	$Q_{(t+3)}$	–	–
Hourly distributed data models							
Cause-effect models							
HD[b] model 5	Current rainfall from nine raingauge station with four antecedent rainfall from nine raingauge station	45	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
HD model 6	Current rainfall from nine raingauge station with five antecedent rainfall from nine raingauge station	54	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
Combined models							
HD model 11	Current rainfall from nine raingauge station and current inflow with three antecedent rainfall from nine raingauge station and three antecedent inflow	40	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$
HD model 12	Current rainfall from nine raingauge and current inflow with two antecedent rainfall from nine raingauge station and three antecedent inflow	31	$Q_{(t+2)}$	$Q_{(t+4)}$	$Q_{(t+6)}$	$Q_{(t+8)}$	$Q_{(t+10)}$

[a] DD – Daily-distributed input data, time '*t*' is in days.
[b] HD – Hourly distributed input data, time '*t*' is in hours.

above models. It is also found that distributed input data models performed slightly inferior to lumped data, because of clustering effect as well as more complexity in fuzzifying large number of input. All these ANFIS models, especially the peak and the higher lead period prediction are better than the best ANN models. Thus from Table 6 it is found that DL-ANFIS model 18 performed better than any other ANFIS model. It is also seen that all the lumped time-series and lumped combined ANFIS models performed better with three membership functions whereas cause-effect models performed better with two memberships function. The combined DL-ANFIS model 18 with 1 day lead time obtained best values of *R* (0.96), *E* (0.91), *RMSE* (8.07), *AIC* (10753.42) and *BIC* (10759.97). Thus from the above results it may be concluded that the lumped DL-ANFIS model 18, which used three inputs outperformed the other ANFIS models. The scatter plot of DL-ANFIS model 18 with lead time of 1 day, 2 day and 3 day is shown in Fig. 3. The superiority of the ANFIS to ANN method may be due to fuzzy partioning of the inputs space and creating a rule-base to generate the output. From the scatter plots, it can be seen that low and medium inflows are well predicted by the ANFIS model but peak inflows are still under predicted. It is also observed that prediction of ANFIS model at higher lead period (3 day) is found to be better than ANN models. The initial and final membership function for DL-ANFIS model 18 is shown in Fig. 4a and b respectively. From this membership functions it is seen that the inflows between 60 and 200×10^6 m^3 (that are high frequency inflows) has large number of fuzzy rules and predicted well. Still the peak inflows are under predicted. In this type also the model performance has deteriorated as the lead period increases from 1 day to 3 days.

9.3. Daily lumped and distributed data LGP models

For LGP modelling, Discipulus software package Pro Version Lite 4.0, developed by Francone (2004), was applied. Unlike ANN, in LGP each multi-time step is modelled separately. Thus it is three runs for each LGP model. The codes are defined in terms of functions and terminal sets that modify the contents of internal memory and program counter. LGP algorithm produces multiple lists of programs representing models with the best fit to its training and calibrating data. Twenty different models, same as that of previous techniques with various input combinations have been developed using LGP technique. After several trials of LGP models, the functional set and operational parameters are finalized and are given in Table 7. The population size of 500 provided high search space for LGP solution. The parameter "initial program size" and "maximum program size" indicate the maximum size of the program of the initial population and of the population from subsequent generations, respectively. From various trials, it was observed that a large initial program size, which leads to good initial exploration of the search space, resulted better. GP usually results in higher mutation rate (Kisi and Guven, 2010; Guven and Kisi, 2011) over the generation. In GP model, a single solution evolves itself thoroughly before going for next generation. Thus for higher evolution within the generation, a higher probability of mutation is required. The objective function was to generate the computer program with least MSE. The statistical performance resulted from the best models in each type of input like time-series; cause-effect and combined input are shown in Table 8.

On analysing Table 8, it is apparent that the performance of LGP is better than ANN and ANFIS model. In contrast to ANN in LGP models the lumped cause-effect model is better than time-series but distributed input cause-effect is equally good as that of time-series. In both the case, the combined input performed better, however, the lumped data model has edge over the distributed data model. The reason may be due to the difficulty in capturing the more patterns in the input and output data, same is advantage as that of ANN models. From Table 8, it can be seen that distributed LGP combined models performed better than lumped data cause-effect and time-series models. From the Table 8 it can be revealed that DL-LGP model 18 with 1 day ahead obtained the best statistics of *R* (0.98), *E* (0.93), *RMSE* (6.95), *AIC* (9989.05), and *BIC* (9995.60) respectively. Lumped combined LGP model produced excellent results for 1 day ahead and acceptable results for 2 days ahead and reasonably accurate results for 3 days ahead prediction and is better than ANN and ANFIS technique. There is no significant difference in *R* and *E* value from 1 day, 2 days and 3 days ahead inflow prediction. The scatter plot of this best combined DL-LGP model 18 for 1 day, 2 days and 3 days ahead inflow prediction is shown

Table 5
Performance measures of daily lumped and distributed data ANN models.

Model	Model type	Performance criteria	Training			Testing		
			Lead period			Lead period		
			1 Day	2 Days	3 Days	1 Day	2 Days	3 Days
Time-series model								
DL-ANN model 6	(6–4–3)	R	0.88	0.78	0.75	0.89	0.76	0.73
		E	0.81	0.69	0.63	0.82	0.66	0.60
		RMSE	12.79	12.35	11.00	11.06	11.25	11.06
		AIC	13127.6	12947.3	12351.1	12378.68	12467.86	12379.10
		BIC	13134.1	12953.8	12357.7	12385.23	12474.41	12385.65
Lumped data models								
Cause-effect models								
DL-ANN model 15	(8–4–3)	R	0.84	0.78	0.78	0.82	0.75	0.73
		E	0.75	0.72	0.67	0.76	0.72	0.64
		RMSE	11.84	11.56	12.27	11.80	11.89	12.20
		AIC	29699.35	29411.7	30128.0	12711.42	12752.35	12885.09
		BIC	29706.7	29419.1	30135.4	12717.96	12758.90	12891.63
Combined models								
DL-ANN model 18	**(3–6–3)**	**R**	**0.95**	**0.94**	**0.91**	**0.94**	**0.92**	**0.92**
		E	**0.92**	**0.87**	**0.83**	**0.90**	**0.85**	**0.82**
		RMSE	**8.37**	**10.45**	**10.93**	**8.55**	**10.56**	**10.74**
		AIC	**25532.96**	**28194.2**	**28743.6**	**11051.70**	**12142.23**	**12230.26**
		BIC	**25540.36**	**28201.6**	**28751.0**	**11058.24**	**12148.78**	**12236.80**
Distributed data models								
Cause-effect models								
DD-ANN model 3	(27–18–3)	R	0.82	0.81	0.78	0.75	0.72	0.70
		E	0.71	0.70	0.68	0.73	0.71	0.65
		RMSE	11.72	11.90	12.07	12.48	12.76	12.56
		AIC	9441.07	9499.52	9553.92	4149.14	4185.60	4159.64
		BIC	9447.33	9505.78	9560.17	4154.55	4191.00	4165.04
Combined models								
DD-ANN model 15	(25–15–3)	R	0.92	0.85	0.81	0.90	0.82	0.77
		E	0.88	0.82	0.78	0.84	0.79	0.65
		RMSE	13.00	10.78	11.14	12.22	10.79	11.06
		AIC	9838.58	9120.45	9246.43	4114.55	3910.07	3950.68
		BIC	9844.83	9126.70	9252.68	4119.95	3915.48	3956.08

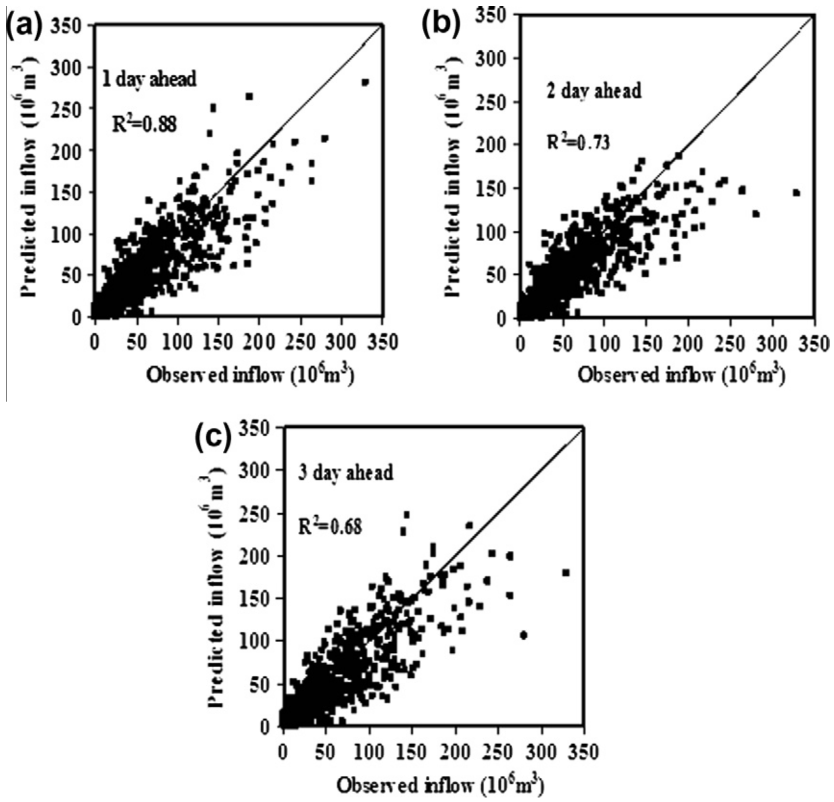

Fig. 2. Scatter plot of observed and multi-time-step ahead predicted inflow by best DL-ANN model 18 during testing period (combined input).

Table 6
Performance measures of daily lumped and distributed data ANFIS models.

Models	No. of inputs	Membership function	Performance criteria	Training			Testing		
				Lead period			Lead period		
				1 Day	2 Days	3 Days	1 Day	2 Days	3 Days
Time-series models									
DL-ANFIS model 5	5	3	R	0.92	0.80	0.66	0.90	0.81	0.68
			E	0.86	0.63	0.43	0.86	0.65	0.65
			RMSE	10.96	11.50	12.47	10.56	11.21	11.63
			AIC	28775.76	29348.33	30319.16	12142.69	12450.11	12639.18
			BIC	28783.15	29355.73	30326.55	12149.24	12456.65	12645.73
Lumped data models									
Cause-effect models									
DL-ANFIS model 14	7	2	R	0.93	0.90	0.81	0.90	0.83	0.76
			E	0.89	0.85	0.65	0.87	0.67	0.65
			RMSE	11.41	11.55	12.05	11.80	12.731	12.44
			AIC	29257.69	29403.06	29915.46	12711.42	13102.88	12986.68
			BIC	29265.08	29410.45	29922.85	12717.96	13109.43	12993.26
Combined models									
DL-ANFIS model 18	**3**	**3**	**R**	**0.94**	**0.92**	**0.90**	**0.96**	**0.95**	**0.93**
			E	**0.91**	**0.89**	**0.87**	**0.91**	**0.91**	**0.90**
			RMSE	**7.94**	**7.88**	**8.37**	**8.07**	**8.26**	**8.43**
			AIC	**24899.67**	**24809.44**	**25535.54**	**10753.42**	**10874.04**	**10982.75**
			BIC	**24907.07**	**24816.83**	**25542.93**	**10759.97**	**10880.59**	**10989.30**
Distributed models									
Cause-effect models									
DD-ANFIS model 4	36	2	R	0.91	0.84	0.82	0.87	0.85	0.81
			E	0.87	0.82	0.78	0.79	0.73	0.73
			RMSE	11.16	11.58	12.11	12.59	12.79	12.89
			AIC	9253.31	9394.99	9566.61	4163.56	4189.45	4202.25
			BIC	9259.56	9401.24	9572.86	4168.96	4194.86	4207.65
Combined models									
DD-ANFIS model 16	28	3	R	0.94	0.89	0.88	0.92	0.88	0.85
			E	0.85	0.76	0.72	0.89	0.76	0.69
			RMSE	10.34	11.91	13.73	11.31	13.15	13.19
			AIC	8960.64	10780.68	11204.07	4757.26	4861.19	5062.56
			BIC	8966.89	10786.93	11210.32	4762.66	4866.59	5067.96

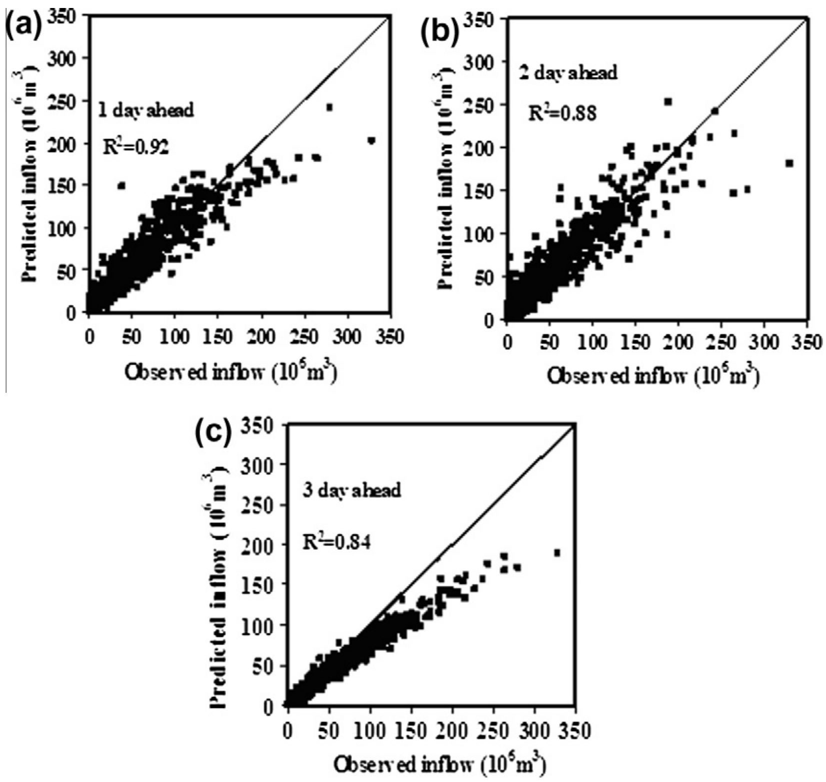

Fig. 3. Scatter plot of observed and multi-time step ahead predicted inflow by DL-ANFIS model 18 during testing period (combined input).

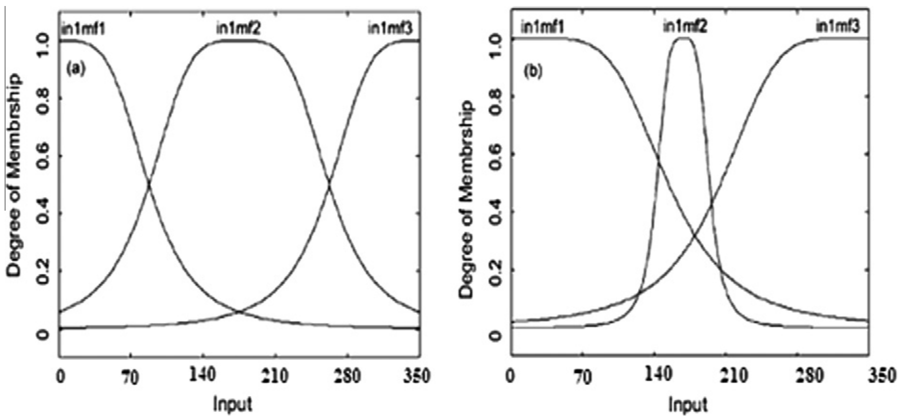

Fig. 4. (a) Initial membership function and (b) Final membership function of combined DL-ANFIS model 18.

Table 7
Parameters of the LGP model.

Parameter	Values
Population size	500
Function set	+, −, ∗, /, √, ln(x), sin, cos, tan
Initial program size	80
Maximum program size	512
Crossover rate (%)	50
Homologous crossover (%)	50–95
Mutation rate (%)	90

in Fig. 5. From scatter plots, it is observed that LGP technique performed very well for inflow prediction with shorter lead prediction and slightly better for higher lead period. The Fig. 5b, shows that the values are mostly underpredicted for higher lead prediction. LGP has other certain advantages over the ANN and ANFIS such as fewer controlling mathematical functions (and hence more flexibility in data mining) and a built-in capacity to handle a large amount of data. It is observed that a combination of rainfall and inflow variables in the input vector significantly improved the model performances.

Table 8
Performance measures of best daily lumped and distributed data LGP models.

Models	No. of inputs	Performance criteria	Training			Testing		
			Lead period			Lead period		
			1 Day	2 Days	3 Days	1 Day	2 Days	3 Days
Time-series models								
DL-LGP model 6	6	R	**0.94**	0.88	0.86	0.90	0.88	0.85
		E	0.87	0.85	0.82	0.84	0.81	0.79
		RMSE	10.28	11.19	11.54	10.60	11.39	11.54
		AIC	28004.05	29020.59	29392.70	12159.02	12528.56	12596.78
		BIC	28011.44	29027.99	29400.09	12165.57	12535.11	12603.32
Lumped data models								
Cause-effect models								
DL-LGP model 14	7	R	0.94	0.90	0.88	0.92	0.87	0.81
		E	0.87	0.85	0.79	0.86	0.76	0.69
		RMSE	10.58	11.01	11.71	10.46	11.89	12.22
		AIC	28344.84	28829.49	29915.46	12091.65	12752.35	12885.09
		BIC	28352.23	28836.89	29922.85	12098.20	12758.40	12891.63
Combined models								
DL-LGP model 18	**3**	**R**	**0.97**	**0.96**	**0.96**	**0.98**	**0.96**	**0.95**
		E	**0.95**	**0.92**	**0.92**	**0.93**	**0.92**	**0.92**
		RMSE	**6.80**	**7.43**	**7.69**	**6.95**	**7.43**	**7.69**
		AIC	**23043.53**	**24102.04**	**24518.15**	**9989.05**	**10331.1**	**10509.50**
		BIC	**23050.9**	**24109.44**	**24525.54**	**9995.60**	**10337.71**	**10516.05**
Distributed data models								
Cause-effect models								
DD-LGP model 5	45	R	0.87	0.74	0.66	0.89	0.77	0.68
		E	0.74	0.54	0.41	0.81	0.57	0.42
		RMSE	14.56	14.96	16.48	16.70	15.37	16.15
		AIC	10273.20	10377.13	10748.24	4627.72	4491.36	4572.69
		AIC	10273.20	10377.13	10748.24	4627.72	4491.36	4572.69
Combined models								
DD-LGP model 16	28	R	0.94	0.92	0.93	0.95	0.90	0.91
		E	0.87	0.86	0.86	0.90	0.84	0.82
		RMSE	10.22	10.83	11.01	10.89	12.66	13.91
		AIC	8915.87	9138.20	9201.41	3925.23	4172.67	4327.37
		BIC	8922.12	9144.45	9207.67	3930.63	4178.07	4332.78

The impact of each input vector was analysed and is presented in Table 9. A value of 100% in the frequency column indicates that the particular input variable appeared in set of all best programs. The average and maximum effect of removing all the instances of a particulars input from each best program is also presented. The results are scaled between 0 and 1. A value of 1 represents the largest impact value possible. It is seen that the lag one rainfall and inflow has the highest impact in predicting the future inflow. Thus based on the results it may be concluded that the LGP model has a great ability to learn from input–output patterns and work efficiently for reservoir inflow prediction with greater accuracy. In addition, there is only marginal difference between DL and DD-LGP models. Thus, LGP models overcome the problem of lumping the input data.

9.4. Hourly lumped and distributed data GFF Models (ANN)

In the present study, Generalized Feed Foreword (GFF) network with conjugate gradient (CG) algorithm is adopted for hourly data models. It is found that the TDRNN worked better for daily data but performed poorly for the hourly data (Magar, 2011). The reason is that the higher cross correlation and autocorrelation among the input variables lead to overtraining of the TDRNN network. GFF with two hidden layer network has performed better and hence adopted in this study. The input vector (number of inputs) for input layer depends upon type of model as shown in Table 3. The neurons in the output layer (five numbers) represent the inflow at 2 h, 4 h, 6 h, 8 h and 10 h ahead hourly inflow being modelled. The optimum number of neurons (N) in the hidden layer is varied from 2 to 30 and best architecture was finalized to capture the rainfall–inflow relationship. The performance summary of best lumped and distributed data ANN models during training and testing periods are displayed in Table 10.

The daily ANN models are better than hourly models due to two reasons (i) the time step and (ii) secondly the length of data in case of daily models is 47 years (17,165 data sets) out of which 12,015 data set used for training and 5150 data set for testing, where as in case of hourly data out of total 4 years (14,688 data sets), 10,281 data set is used for training and 4407 data set is used for testing. This result indicates that even the non-linear ANN model could not predict the peak inflows properly, may be because of very high non-linearity in the observed peak inflows; hence, the other non-linear techniques are attempted.

9.5. Hourly lumped and distributed data ANFIS models

Similar to daily lumped and distributed data models, hourly lumped and distributed data ANFIS models have been developed with input and output parameters. In case of ANFIS models the number of membership functions (MFs) are varied and associated with each input variable are set by trial and error. It is found that large number of MFs on the input variable increased the performance due to the large number of if then fuzzy rules and at the same time increases the model complexity and hence affected the model parsimony. Therefore, numbers of MFs are varied between 2 and 4 to comprise parsimonious models. The resulted statistical performances of the best lumped and distributed data ANFIS models is shown in Table 10. Studying the performance of all the ANFIS models it is found that HL-ANFIS model 12 performs better than any other model. The combined model HL-ANFIS 12 with 2 h lead period displayed best values of *R* (0.93), *E* (0.87), *RMSE* (80.43) and *%MF* as 9.54 respectively. Even though the prediction accuracy of 10 h lead period is less than 2 h, there is no significant reduction in the performance.

9.6. Hourly lumped and distributed data LGP models

The LGP soft computing technique, which provides input–output relationship, has also been employed for prediction of

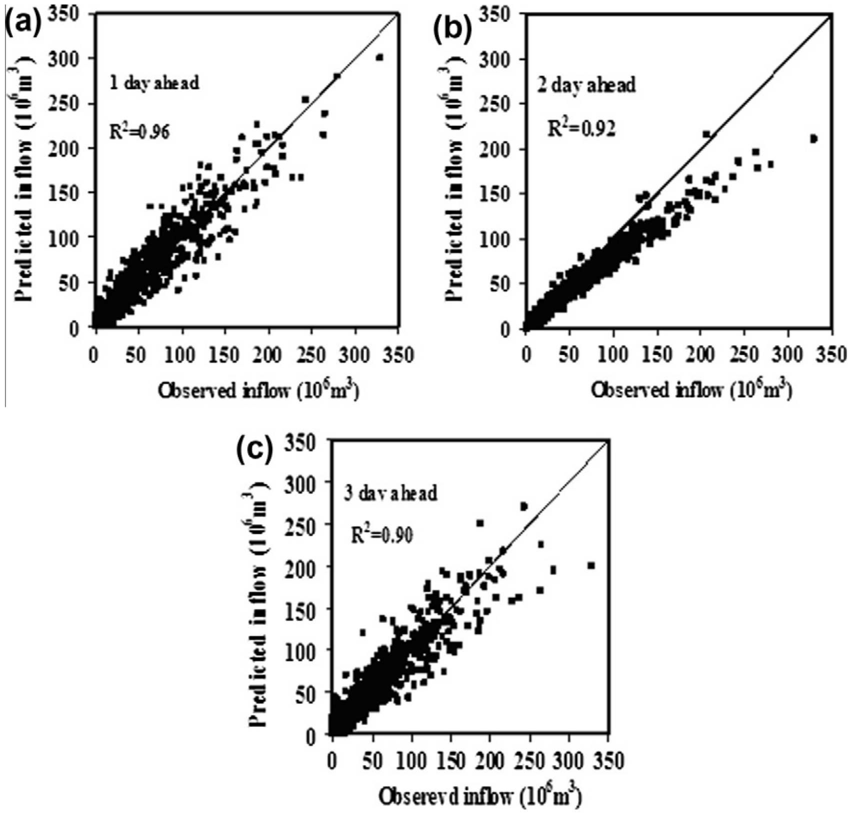

Fig. 5. Scatter plot of observed and multi-time-step ahead predicted inflow by DL-LGP model 18 during testing period (combined input).

Table 9
Impact of each input variable in the combined DL-LGP model 18.

Input parameters	Frequency (%)	Average impact	Maximum
Impact			
$P_{(t-1)}$	63	0.018	0.039
$P_{(t)}$	83	0.132	0.246
$Q_{(t)}$	100	0.255	0.745

multi-time-step hourly inflow prediction. In hourly multi-time step model, it is five runs for each LGP model. The performances of best models in each type are listed in Table 10. From Table 10 it can be observed that time-series HL-LGP model 2 outperformed all the models. In LGP models, also the time-series and cause-effect models performed equally well. However, the combined input models outperformed for higher lead period. Analysing the results of HL combined models, Table 10 shows that all the combined HL-LGP models resulted in acceptable results, showing almost comparable values of goodness of fit criteria and better than other techniques for same number of inputs. From the Table 10 it can be revealed that combined HL-LGP model 11 outperformed all other models especially for of 2 h lead period and obtained the best statistics of *R* (0.98), *E* (0.93), *RMSE* (77.81), and *%MF* 8.81% respectively.

The scatter plot of observed and predicted inflow by the best combined HL-LGP model 11 with 2 h, 4 h, 6 h, 8 h and 10 h prediction is shown in Fig. 6. From the scatter plot, it can be observed that all inflow values such as low, medium, and peak are clustered along the ideal line. It is seen that in hourly data modelling also, the antecedent inflow has highest impact followed by antecedent rainfall. This also indicates that average time of concentration is around 2 h only. However, the LGP model has predicted very well up to 8 h lead prediction (Fig. 6a–d). Thus from the hourly lumped input data LGP model it may be concluded that the LGP models have a great ability to map the relationship from input–output patterns and work strongly in reservoir inflow prediction with greater accuracy.

10. Conclusions

This study investigated the applicability and capability of AI models for inflow forecasting applied to Koyna watershed in Maharashtra, India. Two different set of inputs are studied, firstly a lumped input data and secondly a distributed data input for the same basin. Twenty daily data models and twelve hourly data models are developed based on different input structure combination such as time-series, cause-effect and combined models and also their performances are evaluated. The daily and hourly ANN model results suggested that the choice of the number of inputs, percentage of data for training and testing, number of hidden nodes, network memory structure has an impact on the model prediction efficiency and found to be by trial and error. It is concluded that ANN technique still requires large number of trial and error work and time consuming efforts involved in developing the structures to achieve optimal performance. Also, the time-series models performed better than pure cause-effect models. Since in time-series it is only two patterns, where as in cause-effect it is ten patterns to be recognized.

ANFIS models performed better than ANN models for daily data set due to fuzzification of inputs in terms of membership functions. ANFIS models predicted medium and peak inflows better than ANN models. Thus, it may be concluded that neuro-fuzzy approach has the advantage of reduced training time not only due to its smaller dimensions but also due to the ability of the network to initialize with parameters relating to the problem domain. It may be

Table 10
Performance measures of hourly lumped and distributed data models.

Models	Performmance criteria	Training Lead period					Testing Lead period				
		2 h	4 h	6 h	8 h	10 h	2 h	4 h	6 h	8 h	10 h
Lumped data ANN Combined model											
HL-ANN model 12(5–7–7–5)	R	0.93	0.95	0.93	0.92	0.90	0.91	0.89	0.89	0.88	0.84
	E	0.79	0.54	0.76	0.73	0.70	0.81	0.90	0.88	0.83	0.81
	RMSE	113.08	114.82	122.79	132.40	140.55	113.08	114.82	122.79	132.40	140.55
	%MF	12.45	−12.78	−2.75	−3.12	−12.65	11.39	−1.50	5.90	1.34	−2.11
Distributed data ANN Combined model											
HD-ANN model 8(19–10–10–5)	R	0.92	0.92	0.91	0.89	0.88	0.92	0.91	0.90	0.87	0.86
	E	0.86	0.88	0.86	0.79	0.77	0.85	0.82	0.80	0.78	0.76
	RMSE	162.64	247.71	160.31	297.98	211.72	145.59	269.85	292.28	295.68	298.26
	%MF	16.56	−17.34	−17.89	−15.90	23.23	10.82	−25.46	−16.40	−25.85	25.87
Lumped data ANFIS Combined model											
HL-ANFIS model 12	R	0.95	0.94	0.92	0.88	0.88	0.93	0.92	0.88	0.86	0.85
	E	0.90	0.88	0.87	0.84	0.82	0.87	0.90	0.92	0.85	0.83
	RMSE	79.09	81.32	82.32	83.87	84.11	80.43	82.43	84.76	85.65	86.98
	%MF	10.45	−13.87	−12.75	13.09	14.98	9.54	−12.78	−10.86	11.76	12.76
Distributed data ANFIS Combined model											
HD-ANFIS model 11	R	0.95	0.94	0.92	0.90	0.88	0.93	0.92	0. 90	0.88	0.85
	E	0.92	0.85	0.84	0.81	0.79	0.87	0.86	0.87	0.85	0.83
	RMSE	79.09	81.32	82.32	83.87	84.11	93.12	82.43	84.76	85.65	86.98
	%MF	10.45	−13.87	−12.75	13.09	14.98	12.90	−12.78	−10.86	11.76	12.76
Lumped data LGP Combined model											
HL-LGP model 11	**R**	**0.98**	**0.95**	**0.92**	**0.90**	**0.89**	**0.98**	**0.94**	**0.92**	**0.88**	**0.86**
	E	**0.95**	**0.91**	**0.87**	**0.83**	**0.81**	**0.93**	**0.90**	**0.85**	**0.82**	**0.83**
	RMSE	**75.76**	**76.76**	**77.32**	**77.41**	**77.65**	**77.81**	**78.25**	**81.76**	**84.09**	**84.23**
	%MF	**−5.47**	**−11.06**	**−8.73**	**−9.76**	**−12.06**	**8.81**	**−9.10**	**11.05**	**−1.54**	**−1.41**
Distributed data LGP Combined model											
HD-LGP model 8	R	0.97	0.94	0.92	0.89	0.87	0.95	0.93	0.92	0.88	0.85
	E	0.94	0.88	0.33	0.79	0.83	0.90	0.91	0.87	0.82	0.83
	RMSE	98.64	117.71	120.31	137.98	149.43	87.58	126.52	116.76	132.53	143.33
	%MF	−8.15	−8.62	−8.60	−23.52	−14.54	11.54	−13.84	−14.58	−13.47	14.43

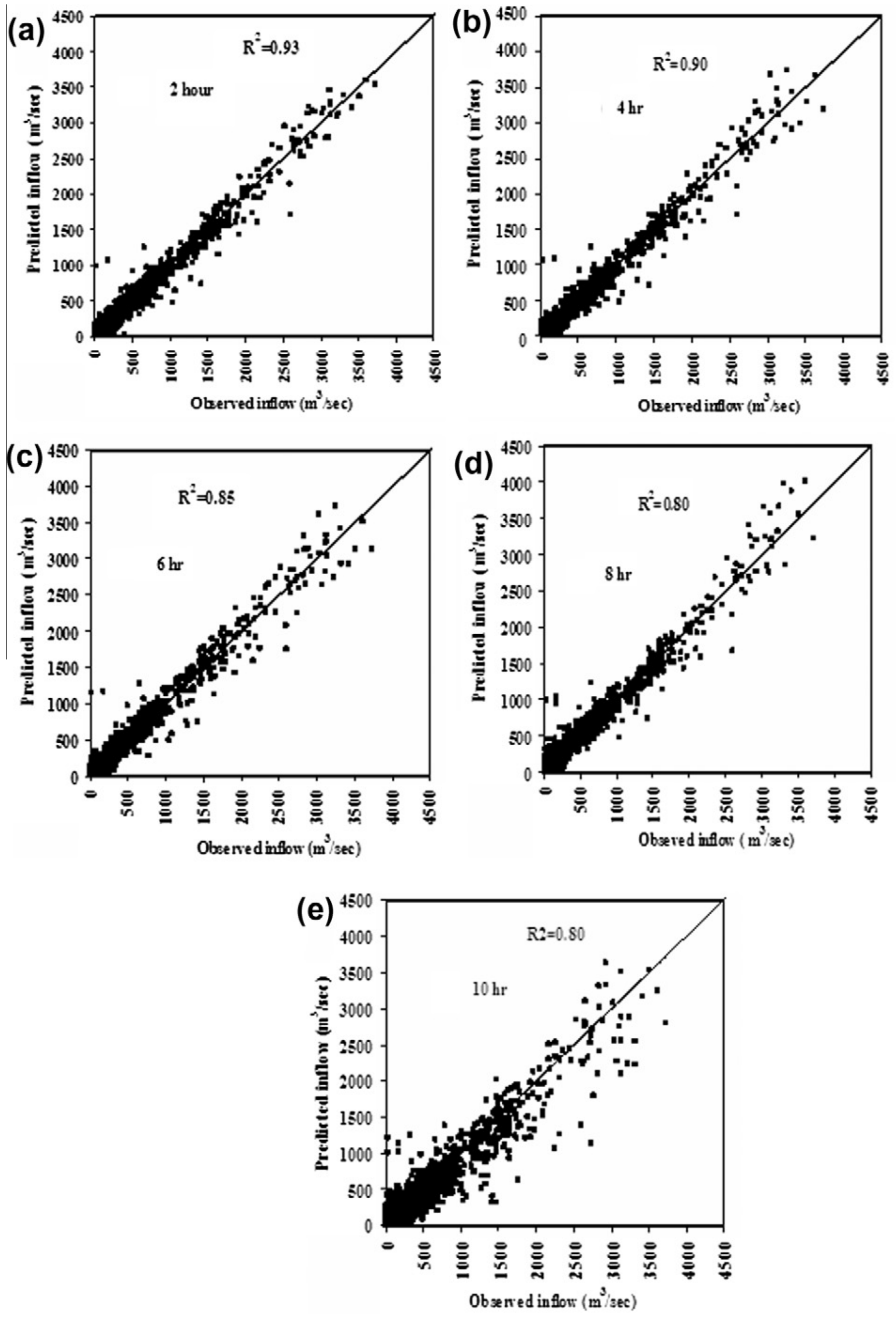

Fig. 6. Scatter plot of observed and multi-time-step ahead predicted inflow by best HL-LGP model 11 during testing period (combined input).

concluded that due to exponential relationship in ANFIS theory, increasing the number of inputs added the model complexity and reduced the performance of the models.

LGP models responded well to most of the fluctuations within the data and have resulted in better prediction of low, medium and peak inflows amongst all the models developed in this study. The reason may be due to fewer mathematical functions and better build in capacity used in LGP. Also, the lumped and distributed model performs equally better since they are highly correlated. LGP models found to be easier to model, less time consuming and are recommended for their wider application to other reservoirs. In case of lumped data models, DL-LGP model 18 obtained the best statistics of *R* (0.98), *E* (0.93), *RMSE* (6.95), *AIC* (9989.05), and *BIC* (9995.60) respectively and in case of distributed data models, DD-LGP model 16 obtained the best statistics of *R* (0.95), *E* (0.90). In case of hourly data models, HL-LGP model 11 (input structure of $P_{(t-1)}$, $P_{(t)}$, $Q_{(t-1)}$, $Q_{(t)}$) outperformed all other models and obtained the best statistics of *R* (0.98), *E* (0.93), *RMSE* (77.81), and *%MF* 8.81% respectively. It may be concluded that the LGP models captured the linear as well as non-linear relationship in reservoir inflow accurately. It is also found that irrespective of the technique used, combined input with three dimensions performed well, but the reason is not known and further research may be explored in this direction.

Acknowledgements

The authors express their sincere thanks to Executive Engineer, Irrigation Department and office personnel of Koyna Dam Division Government of Maharashtra, India for providing necessary data to carry out this work. The authors also acknowledge the Ministry of Water Resources, Government of India, and New Delhi for sponsoring the research project through Indian National Committee on Hydrology.

Appendix A. Model performance criteria

Performance criteria's	Equation
Coefficient of correlation (R)	$\frac{\sum_{i=1}^{N}(obs_i - avgobs_i)(calc_i - avgcalc_i)}{\sqrt{\sum_{i=1}^{N}(obs_i - avgobs_i)^2} \times \sqrt{\sum_{i=1}^{N}(calc_i - avgcalc_i)^2}}$
Nash Sutcliffe efficiency (E)	$E = 1 - \frac{\sum_{i=1}^{N}(obs_i - calc_i)^2}{\sum_{i=1}^{N}(obs_i - avg.obs_i)^2}$
Akaike information criterion (AIC) and Bayesian information criterion (BIC)	$AIC = m\ln(RMSE) + 2n$ $BIC = m\ln(RMSE) + n\ln(m)$
Root mean square error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(obs_i - calc_i)^2}{n}}$
Percentage mean error in estimating peak flow (%MF)	$\%MF = \frac{max.calc_i - max.obs_i}{max.calc_i} \times 100$

Where obs_i = observed inflow, $calc_i$ = calculated/predicted inflow, $avg.obs_i$ = average observed inflow, $avg.calc_i$ = average calculated/predicted inflow. E = Efficiency of the model; 'm' is the number of input–output patterns, and 'n' is the number of parameters to be estimated. max.$calc_i$ = maximum estimated flow, max.obs_i = maximum observed flow

References

Aqil, M., Kita, I., Yano, A., Nishiyama, S., 2007. A comparative study of artificial neural networks and neuro-fuzzy in continuous modelling of the daily and hourly behaviour of runoff. Journal of Hydrology 337 (1–2), 22–34.

ASCE Task Committee, 2000a. Artificial neural networks in hydrology I: Preliminary concepts. Journal of Hydrologic Engineering, ASCE 5 (2), 115–123.

ASCE Task committee, 2000b. Artificial neural networks in hydrology II: Hydrologic applications. Journal of Hydrologic Engineering, ASCE 5 (2), 124–137.

Aytek, A., Alp, M., 2008. An application of artificial intelligence for rainfall-runoff modelling. Journal of Earth System Sciences 117 (2), 145–155.

Babovic, V., Keijzer, M., 2000. Genetic programming as model induction engine. Journal of Hydroinformatics 2 (1), 35–60.

Bouzerdoum, A., Mueller, R., 2003. A generalized feedforward neural network architecture and its training using two stochastic search methods. In: Cantu-Paz, E. et al. (Eds.), GECCO, LNCS 2723. Springer, Verlag Berlin Heidelberg, pp. 742–753.

Box, G.E.P., Jenkins, G.M., 1976. Time Series Analysis Forecasting and Control, second ed. Holden Day, San Francisco.

Brameier, M., 2004. On Linear Genetic Programming. Ph.D. Thesis, University of Dortmund, Germany.

Brameier, M., Banzhaf, W., 2001. Evolving teams of predictors with linear genetic programming. Genetic Programming and Evolvable Machines 2 (4), 381–407.

Brameier, M., Banzhaf, W., 2007. Linear Genetic Programming. Springer Science +Business Media, New York (NY).

Brameier, M., Kantschik, W., Dittrich, P., Banzhaf, W., 1998. SYSGP-AC++ Library of Different GP Variants. Technical Report CI-98/48, Collaborative Research Centre 531, University of Dortmund, Germany.

Burlando, P., Rosso, R., Cadavid, L.G., Salas, J.D., 1993. Forecasting of short-term rainfall using ARMA models. Journal of Hydrology 144 (1–4), 193–211.

CDO, 1992. Final Report on Revised Flood Study for Koyna Dam. Government of Maharashtra, Central Design Office, Irrigation Department, India.

Chang, F.J., Chang, L.C., Huang, H.L., 2002. Real-time recurrent learning neural network for stream-flow forecasting. Hydrological Processes 16 (13), 2577–2588.

Chang, F.J., Chang, Y.T., 2006. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. Advances in Water Resources 29, 1–10.

Chang, L.C., Chang, F.J., Chiang, Y.M., 2004. A two-step-ahead recurrent neural network for stream flow forecasting. Hydrological Processes 18 (1), 81–92.

Chau, K.W., Wu, C.L., Li, Y.S., 2005. Comparison of several flood forecasting models in Yangtze river. Journal of Hydrological Engineering. ASCE 10 (6), 485–491.

Chen, S.H., Lin, H.Y., Chang, L.C., Chang, F.J., 2006. The strategy building a flood forecast model by neuro-fuzzy network. Hydrological Processes 20 (7), 1525–1540.

Chidthong, Y., Tanaka, H., Supharatid, S., 2009. Developing a hybrid multi-model for peak flood forecasting. Hydrological Processes 23 (12), 1725–1738.

Coulibaly, P., Anctil, F., Bobee, B., 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. Journal of Hydrology 230 (3–4), 244–257.

Dastorani, M.T., Afkhami, H., Sharifidarani, H., Dastorani, M., 2010. Application of ANN and ANFIS models on dry land precipitation (Case study: Yazd in Central Iran). Journal of Applied Sciences 10 (20), 2387–2394.

de Vos, N.J., Rientjes, T.H.M., 2005. Constraints of artificial neural networks for rainfall–runoff modelling: trade-offs in hydrological state representation and model evaluation. Hydrology Earth System Sciences 9, 111–126.

Dorum, A., Yarar, A., Sevimili, M.F., Onucyildiz, M., 2010. Modelling the rainfall–runoff data of Susurluk basin. Journal of Expert System with Applications 37 (9), 6587–6593.

Drunpob, A., Chang, N.B., Beaman, M., 2005. Stream flowrate prediction using Genetic Programming Model in a semi-arid coastal watershed. In: Walton, Raymond (Ed.), Proceedings of World Water and Environmental Resources Congress, May 15–19, 2005. ASCE, Anchorage, Alaska, USA.

Elman, J.L., 1990. Finding structure in time. Cognitive Science 14, 179–211.

Elshorbagy, A., Corzo, G., Srinivasulsu, S., Solomatine, D.P., 2010a. Experimental investigations of the predictive capabilities of data driven modelling techniques in hydrology – Part 2: Application. Hydrology Earth System Sciences 14, 1943–1961.

Elshorbagy, A., Corzo, G., Srinivasulu, S., Solomatine, D.P., 2010b. Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology – Part 1: Concepts and methodology. Hydrology Earth System Sciences 14, 1931–1941.

Firat, M., 2008. Comparison of artificial intelligence techniques for river flow forecasting. Hydrology and Earth System Sciences 12 (1), 123–139.

Firat, M., Gungor, M., 2008. Hydrological time-series modelling using an adaptive neuro-fuzzy inference system. Hydrological Processes 22 (13), 2122–2132.

Foster, J.A., 2001. Review: discipulus: a commercial genetic programming system. Genetic Programming and Evolvable Machines 2 (2), 201–203.

Francone, F., 2004. Discipulus Lite™ Owner's Manual, Version 4.0. Register Machine Learning Technologies, Inc.

Garg, V., Jothiprakash, V., 2010. Reservoirs trap efficiency estimation using artificial neural networks and genetic programming. Journal of Hydrologic Engineering ASCE 12 (15), 1001–1015.

Giles, C.L., Lawrence, S., Tsoi, A.C., 1997. Rule inference for financial prediction using recurrent neural networks. In: Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, (CIFEr) IEEE, Piscataway, NJ; 253–259.

Googhari, S.K., Feng, H.Y., Ghazali, A.H.B., Shui, L.T., 2010. Neural networks for forecasting Daily Reservoir Inflows. Pertanika Journal Science and Technology 18 (1), 33–41.

Guven, A., 2009. Linear genetic programming for time-series modelling of daily flow rate. Journal of Earth System Science 118 (2), 137–146.

Guven, A., Kisi, O., 2011. Daily pan evaporation modeling using linear genetic programming technique. Irrigation Science 29 (2), 135–145.

Htike, K.W., Khalifa, O.O., 2010. Rainfall forecasting models using focused time-delay neural networks. In: International Conference on Computer and Communication Engineering (ICCCE), 11–12 may, Kuala Lumpur, Malaysia 978-1-4244-6235-3/10© IEEE.

Hung, N.Q., Babel, M.S., Weesakul, S., Tripathi, N.K., 2009. An artificial neural network model for rainfall forecasting in Bangkok, Thailand. Hydrology Earth System Sciences 13, 1413–1425.

Jain, A., Indurthy, S.K.V.P., 2003. Comparative analysis of event based rainfall–runoff modelling techniques – deterministic, statistical and artificial neural networks. Journal of Hydrologic Engineering, ASCE 8 (2), 93–98.

Jang, J.S.R., 1993. Anfis: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23 (3), 665–685.

Jothiprakash, V., Kote, A.S., 2010. Improving the performance of data-driven techniques through data pre-processing for modelling daily reservoir inflow. Hydrological Sciences Journal 56 (1), 168–186.

Jothiprakash, V., Magar, R.B., Kalkutki, S., 2010. Lumped and distributed data rainfall–runoff models using artificial neural network for an intermittent river. Journal of Flood Engineering 1 (2), 159–173.

Jothiprakash, V., Magar, R., Kalkutki, S., 2009. Rainfall–runoff models using adaptive neuro fuzzy inference system (ANFIS) for an intermittent river. International Journal of Artificial Intelligence, Autumn 3 (A09), 1–23.

Jothiprakash, V., Magar, R., 2009. Soft computing tools in rainfall–runoff modelling. ISH Journal of Hydraulic Engineering 15 (SP-1), 84–96.

Karamouz, M., Razavi, S., Araghinejad, S., 2008. Long-lead seasonal rainfall forecasting using time-delay recurrent neural networks: a Case study. Hydrological Processes 22, 229–241.

Keskin, M.E., Taylan, D., Terzi, O., 2006. Adaptive neural-based fuzzy inference system (ANFIS) approaches for modelling hydrological time-series. Hydrological Sciences Journal 51 (4), 588–598.

Kisi, O., Guven, A., 2010. Evapotranspiration modeling using linear genetic programming technique. Journal of Irrigation and Drainage Engineering, ASCE 136 (10), 715–723.

Kote, A.S., Jothiprakash, V., 2009a. Stochastic and artificial neural network models for reservoir inflow prediction. Journal of Institution of Engineers (India) 90 (18), 25–33.

Kote, A.S., Jothiprakash, V., 2009b. Monthly reservoir inflow modelling using time lagged recurrent networks. International Journal of Tomography and Statistics 12 (F09), 64–84.

Koza, J.R., 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA.

Luk, K.C., Ball, J.E., Sharma, A., 2000. A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. Journal of Hydrology 227, 56–65.

Magar, R.B., Jothiprakash, V., 2011. Intermittent reservoir daily-inflow prediction using lumped and distributed data multi-linear regression models. Journal of Earth System Sciences 120 (6), 1067–1084.

Magar, R. B., 2011. Real-Time Reservoir Inflow Prediction Using Soft Computing Techniques. Ph.D thesis Report, Indian Institute of Technology Bombay, Mumbai.

Maier, H., Dandy, G., 2000. Neural networks for the predictions and forecasting of water resources variables: review of modelling issues and applications. Environmental Modelling and Software 15, 101–124.

Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P., 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. Environmental Modelling and Software 25 (8), 891–909.

Makkeasorn, A., Chang, N.B., Zhou, X., 2008. Short-term streamflow forecasting with global climate change implications – a comparative study between genetic programming and neural network models. Journal of Hydrology 352 (3–4), 336–354.

Mohammadi, K., Eslami, H.R., Kahawita, R., 2006. Parameter estimation of an ARMA model for river flow forecasting using goal programming. Journal of Hydrology 331 (1–2), 293–299.

Mohan, S., Jothiprakash, V., 2000. Fuzzy system modelling for optimal crop planning. Journal of Institution of Engineers (India) 81 (CV), 9–17.

Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models. A discussion of principles. Journal of Hydrology 10 (3), 282–290.

Nayak, P.C., Sudheer, K.P., Ramasastri, K.S., 2005. Fuzzy computing based rainfall–runoff model for real time flood forecasting. Hydrological Processes 19, 955–968.

Nayak, P.C., Sudheer, K.P., Rangan, D.M., Ramasastri, K.S., 2004. A neuro-fuzzy computing technique for modelling hydrological time series. Journal of Hydrology 291 (1–2), 52–66.

Oltean, M., Grosan, C., 2003. A comparison of several linear genetic programming techniques. Complex-Systems 14 (4), 285–313.

Peng, C.S., Buras, N., 2000. Practical estimation of inflows into multireservoir system. Journal of Water Resources Planning and Management, ASCE 126 (5), 331–334.

Razavi, S.R., Araghinejad, S., 2009. Reservoir inflow modelling using temporal neural networks with forgetting factor approach. Water Resources Management 23 (1), 39–55.

Salas, J.D., Delleur, V., Yevjevich, V., Lane, W.L., 1980. Applied Modelling of Hydrologic Time-Series. Water Resources Publication, Littleton, Colorado, 484p (2nd printing 1985, 3rd printing, 1988).

Shanker, M., Hu, M.Y., Hung, M.S., 1996. Effect of data standardization on neural network training. International Journal Management Sciences 24 (4), 385–397.

Singh, V.P., 1988. Hydrologic Systems, Vol. 1: Rainfall–Runoff Modelling. Prentice Hall, Eaglewood Cliffs, NJ.

Sivakumar, B., Jayawardena, A.W., Fernando, T.M.K.G., 2002. River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches. Journal of Hydrology 265 (1–4), 225–245.

Smith, J., Eli, R.N., 1995. Neural network models of rainfall–runoff process. Journal of Water Resources Planning and Management, ASCE 121, 499–508.

Sorooshian, S., Duan, Q., Gupta, V.K., 1993. Calibration of rainfall–runoff models: application of global optimization to the Sacramento soil moisture accounting model. Water Resources Research 29 (4), 1185–1194.

Srinivasulu, S., Jain, A., 2006. A comparative analysis of training methods for artificial neural network rainfall–runoff models. Applied Soft Computing 6 (3), 295–306.

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

Thomas, H.A., Fiering, M.B., 1962. Mathematical synthesis of streamflow sequences for the analysis of river basins by simulation. In: Maass, A. et al. (Eds.), Design of Water Resources Systems. Harvard University Press, Cambridge, USA: Mass, pp. 459–493 (Chapter 12).

Tokar, A.S., Markus, M., 2000. Precipitation-runoff modeling using artificial neural networks and conceptual models. Journal of Hydrologic Engineering, ASCE 5 (2), 156–161.

Tokar, A.S., Johnson, P.A., 1999. Rainfall–runoff modelling using artificial neural networks. 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 (1–4), 132–147.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K.J., 1989. Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics, Speech, and Signal Processing 37 (3), 328–339.

Wang, W., Van Gelder, P.H.A.J.M., Vrijling, J.K., Ma, J., 2006. Forecasting daily streamflow using hybrid ANN models. Journal of Hydrology 324, 383–399.

Wang, W., Chau, K.W., Chang, C.T., Qui, L., 2009. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. Journal of Hydrology 374 (3–4), 294–306.

Wu, C.L., Chau, K.W., 2011. Rainfall–runoff modeling using artificial neural network coupled with singular spectrum analysis. Journal of Hydrology 399, 394–409.

Whigham, P.A., Crapper, P.F., 2001. Modelling rainfall–runoff relationship using genetic programming. Mathematical and Computer Modelling 33 (6–7), 707–721.

Yeh, W.W.-G., 1985. Reservoir management and operation models: a state-of-the art review. Water Resources Research 21 (12), 1797–1818.

Yevjevich, V.M., 1963. Fluctuations of wet and dry years, Part I, Research data assembly and mathematical models. Colorado State University Hydrology Paper 1, 55 pp. (Available from Department of Earth Resources, Colorado State University, Fort Collins, CO 80523).

Zhang, J., Cheng, C.T., Li, S., Wu, L.X.Y., Shen, J.J., 2009. Daily reservoir inflow forecasting combining QPF into ANNs model. Hydrology Earth System Sciences Discussions 6, 121–150.