Stochastic generation and forecasting of monthly hydrometeorological data based on non-traditional neural network

Edson F. Luque Mamani
University of San Agustin-University of Sao Paulo
edluquem@gmail.com
José Herrera Quispe
University of San Agustin
jherreraq@unsa.edu.pe

***Abstract*—The benefits of well-informed water management systems are related to the forecasting skills of hydrological variables. These benefits can be reflected in reducing economic and social losses to come. Therefore, the optimal design of water management projects frequently involves finding the methods or techniques that generate long sequences of hydrological data. These sequences considered as time series can be used to analyze and optimize the performance of the project designed. In order to cover these requirements, this work presents a new model of the stochastic process applied in problems that involve phenomena of stochastic behavior and periodic characteristics. Two components were used, the first one, a type of recurrent neural network relatively recent introduced in the literature and conceptually simple called ESN (echo state network) as the deterministic component, an interesting feature of ESN is that from certain algebraic properties, training only the output of the network is often sufficient to achieve excellent performance in practical applications. The second part of the model incorporates the uncertainty associated with hydrological processes, the model is finally called ESN-RNN. This model was calibrated with time series of monthly discharge data from four different river basins of MOPEX data set. The performance of ESN-RNN is compared with two feedforward neural networks ANN-1, ANN-2 (with one and two past months respectively) and the Thomas–Fiering model. The results show that the ESN-RNN model provides a promising alternative for simulation purposes, with interesting potential in the context of hydrometeorological resources.**

***Index Terms*—Echo state, forecasting, Recurrent, Stochastic process, Neural Network.**

I. Introduction

In probability theory an stochastic process is defined as a set of models that allow the study of problems with random components. Observing a phenomenon with random characteristics for a period of time, it is possible to obtain a trajectory of this observed process. When carrying out the same observation in a different period of time it is possible to obtain another trajectory, different from the first one. Thus, stochastic process corresponds to the set of all possible trajectories that can be observed of this phenomenon. Each trajectory are called a time series. Therefore, time series is considered one realization of stochastic process.

Natural phenomena such as precipitation and streamflow discharge have nonlinear, complex and chaotic characteristics. In order to model the behavior of these phenomena, initially linear approximation was used [1][2]. Afterwards, were developed methods using self-correcting models [3] such as the PAR(p) model [4]. However, these models are statistical and linear, which means that their application in time series of chaotic behavior such as hydrometeorological series, cannot capture real characteristics of this time series being sometimes inadequate [5].

Among the approaches that attempt to model complex non-linear behavior, the Artificial Neural Networks (ANN) are highlighted as machine learning methods in recent years, they may be adequate to deal with such problems. In fact, the ANNs feedforward have been widely used in most research and applications in forecasting models in contrast to Recurrent Neural Networks(RNN)[6]. The RNN's are able to represent dynamic non-linear maps commonly found in time series forecasting tasks[3]. Studies about the performance in forecasting, demonstrate that RNN are better than their peers ANN, in virtually all tests[7]. However, the main reason to prefer to use the feedforward neural networks over recurrent neural networks is that the later generates greater complexity, especially in the neural network training process. This motivated the development of a stochastic process model using Recurrent Artificial Neural Networks in order to take advantage of its aforementioned characteristics. this was possible using an approach called Reservoir Computing(**RC**)[8]. Reservoir Computing is a training approach emerging as simple and fast compared to other approaches used in traditional recurrent ANN, all in order to reduce complexity, and leverage its proven ability to represents the characteristics of time series. Our model also is composed for one non-deterministic component that represents the white noise with a normal distribution, in order to take into account the uncertainty that normally affects natural processes [9]. Therefore, our model could be considered a new proposal in the literature. Finally, as a case study, it was chosen for apply this model in the well-known Model Parameter Estimation Experiment(**MOPEX**) data set[10].

II. History and developments

Non-stationary nature of time series (precipitation and streamflow) is considered one of the most complicated to forecast in hydrology[11][9]. Initial auto-regressive models, as well as models based on Box & Jenkis methodology were used in forecasting problems, for example in [12], the author shows that there is no evidence that multivariate AR(1) models are inadequate as predictors. Studies such as [13][14] describe mathematical models, which can reproduce special characteristics such periodicity, considering the effects of linear correlation. In fact, the non-deterministics concept considered as part of our model was based on Thomas and Fiering work[14]. All these studies propose, the natural behavior of times series can be simulated by simple linear relationship with previous data.

The problem with previous models is that forecasting is naturally a dynamic task. For this reason, artificial intelligence methods have been appearing as alternatives, with good performance as predictors of time series. Works such as [15][16][3][17], they used Artificial Neural Networks (ANN) as forecasting models. Recurrent Neural Networks have show better forecasting ability than feedforward ones, due to their structure[6][8]. This structure allows more parsimonious modeling of the dynamical time series properties. However, it was also demonstrated that, the recurrence structure could cause increased training complexity, and subsequently cause problems of convergence[18]. At the beginning of the 21st century, the concept of Reservoir Computing (RC) was introduced. Reservoir Computing is a training approach that can be remarkably simpler and faster than those traditional applied in RNN, according to [6] few applications of RC in hydrometereology were developed; in [19], the most popular reservoir computing model denominated as **echo state networks**(**ESN**) was used with a Bayesian regularization for forecast short-term energy production of small hydroelectric plants, the results indicate that the proposed model surpasses both RNN's feedforward and ESN in its simple version. Similar works such as [20][21] used ESN to forecast monthly water levels for the four Great Lakes of North America, the authors obtained good performance of ESN networks and attributed it to their highly non-linear and dynamic structure. Echo state networks were found to be valid alternatives to traditional recurrent ANN's, in forecasted water inflow, for example in [11] the authors compare the performance of ESN with SONARX[22] networks, RBF networks and the ANFIS model[23]. In [6], forecasting ability from ESN model was evaluated in a huge variety of large-scale basins. The experiments are carried out by comparing three different ESN variations with two RNN feedforward models. In addition, several aspects of ESN's design are investigated in order to optimize hydrologically relevant information. Only [19] presents a hybrid system as our model proposal, our study consider two components; A deterministic (Recurrent ANN using ESN) and non-deterministic component (Random Noise), in order to take advantage that a hybrid system can provide us in forecasting task.

Section III and IV briefly reviews feedforward and traditional recurrent ANN models and their training methods, after which a short introduction to RC is given and ESN is described more detailed. Section Section V presents details about our proposal. The results of some experiments are presented and discussed in Section VI. Finally, conclusions are drawn in Section VII.

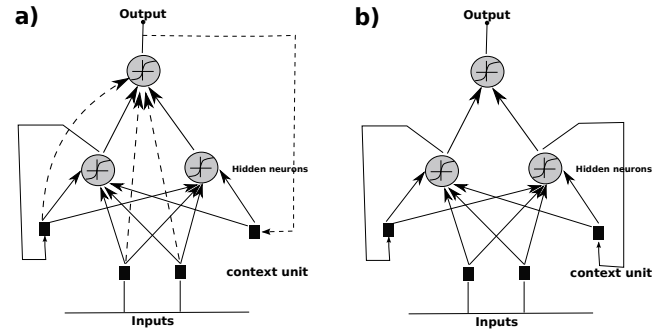


Figure 1. (b) Elman recurrent Artificial Neural Network. a) William-zipser fully recurrent Artificial Neural Network.

III. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a subclass of ANN's characterized by cyclic graphs in its structure. These cycles accumulate previous activities and allow the network stores internal states. These internal states avoid needing to feed the network with the history of previous input and output as the Time-Delay Neural Network[24]. And can use the input sequences in order to perform temporal tasks as forecasting. The RNN output can be described by:

$$Output_{t+1} \cong Forecasting\,(RNNstate, Input_t, Output_t)$$

Figure 1 b) and a) show the two types of traditional recurrent ANN models, the Elman recurrent network [25] and the Willians-Zipser fully recurrent network[26]. These neural networks have cyclic connections on their structure. For instance, the Elman network connects its input to all neurons, including output ones, hidden and output neurons are fully interconnected.

IV. Training Algorithms

Basically, supervised learning means to adjust the network weight matrix **W** using the optimization algorithms , in order to minimize the output error. This is probably the most common approach used among the current types of neural network systems where the input and output are used in the network. A well known training method is the Standard Backpropagation algorithm (BP)[27]. Backpropagation is a method to calculate the gradient of the loss function with respect to the weights. This technique approximates the local minimum by changing these weights along of negative error gradient direction. The objective function E(W) is calculated after BP applies an update to the weights in the network;

$$\triangle\omega_{ji} = -\eta\frac{\partial E}{\partial\omega_{ji}} \tag{1}$$

where η is constant positive value called learning rate. The momentum rate β can be added to the current weight change, this often speeds up the learning process[28]:

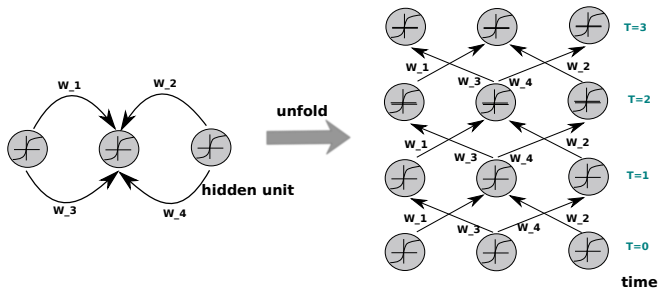


Figure 2. Recurrent Neural Network unfolded in time, the hidden units grouped at time T get inputs from other neurons at previous time steps.

$$\triangle\omega_{ji}^{'} = \beta\triangle\omega_{ji} - \eta\frac{\partial E}{\partial\omega_{ji}} \tag{2}$$

Weight updates can be performed in online mode or based on the mean error over all training data(that is called batch mode). Besides, more sophisticated alternatives to the BP algorithm, such as the Levenberg-Marquardt(LM) have been found faster convergence algorithm[29]. In this algorithm the weight update is obtained by the following equation:

$$\Delta\omega = -\left[H + \mu I\right]^{-1} J^T \rho \tag{3}$$

Where μ is a learning rate, **J** the jacobian matrix, which is the first derivatives of the network error with respect to the weights and biases, and ρ is a vector of network errors. Finally, **H** is an approximation of the Hessian matrix.

Recurrent ANN training

Standard BP algorithm is not suited for networks with cycles in them. Nonetheless, we can apply some artifices and see the RNN like feedforward network by unfolding this RNN network in time as shown in Figure 2. The RNN is interpreted as layered network that keeps the same weights to reusing, we assume the time delay of 1 in each connection in order to create an equivalent feedforward network[30]. This extension of the BP method is called Backpropagation Through Time(BPTT). In BPTT the number of networks copies is equal to time step **T**. It would be impractical in the online training since the memory footprint grows linearly with the time. Therefore, the network unfolding is limited to a chosen truncation depth to keep the method feasible [6].

More sophisticated methods were developed to overcome BPTT limitations, for example Real-Time Recurrent Learning[26], Clockwork Recurrent Network (CW-RNN) that splits hidden layer into M modules running at different clocks[24] and the extended Kalman Filter(EKF) method, which each time estimates optimal weights, given a series of observed outputs, for more details see[31]. However, these methods suffer shortcomings related to the modeling complexity and optimization(gradient)[8]. That means, many updates may be necessary and it could be computationally expensive, the gradient information might becomes useless by weight updates procedure[32].

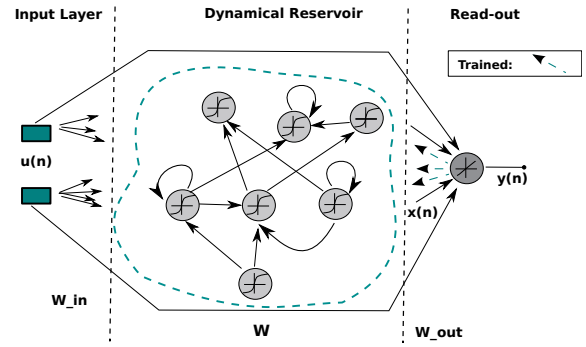


Figure 3. Echo State Network(ESN-RNN), the dynamical reservoir is called "echo states" W, the output layer W_{out} is connected to the hidden reservoir(black arrows), and the input layer is also connected to the output.

Echo State Network

Echo state network(ESN) is a reservoir computing model has been introduced by Jaeger in [8], in order to addressing the difficulties to train RNN networks. Basically ESN-RNN is a clever way to train a RNN where a "reservoir" of hidden units are sparsely connected to each other and the inputs are connected to this reservoir, the hidden connections are not trained, they are randomly initialized. The state of the dynamical reservoir is called "echo states", these states can be understood as an "echo" generated by the input history. The output layer is connected to the hidden reservoir3. Due that the recurrent topology has fixed connections, only the linear readout can be trained, in order to extract the desired response from reservoir states. The idea behind this, is that the "reservoir" retains the system dynamics (current and historical data), which are rich enough to enable the readout learn the functional dependence between inputs and outputs[6].

In accordance to the notation used in [11] the activation of the internal echo states is updated following the equation:

$$x\left(n+1\right) = f\left(\mathbf{W}^{in}u\left(n\right) + \mathbf{W}x\left(n\right)\right) \tag{4}$$

Where $u(n)$ is the input signal, $\mathbf{W}^{in}$ are the weights between the input and the internal states, **W** is the recurrent connection weight matrix, and $x(n)$ is the echo state vector. Finally, f is the activation function of internal units(linear or hyperbolic tangent function).

In order to scale the weight matrix $\boldsymbol{W}$. One condition(echo state property) is needed, that condition asserts that the network states are strongly coupled with the historical input. This is defined in terms of the spectral radius(largest absolute eigenvalue of $\boldsymbol{W}$) ρ(W)<1, this condition is generally satisfied[33].

The optimal output weights W^{out} for the output connections can be found by multiplying the pseudo inverse with the target output Y^{target}:

$$W^{out} = Y^{target}X^{+} \tag{5}$$

Where X is the collected states matrix from ESN. This matrix is the activation states of the reservoir for each training input:

$$X^{+} = \left(XX^{T}\right)^{-1}X^{T} \tag{6}$$

The output $y(n)$ is then computed with equation 7:

$$y(n) = W^{out}x(n) \tag{7}$$

V. Hidrometereological Data Generation Using ESN-RNN With Random Component

We propose a mixed model (stochastic and deterministic) for hydrological synthetic data generation, in terms of monthly series of basic hydrological variables (precipitation, streamflow, and discharge) using ESN-RNN. The model consists of two components;

The **First Component:** is the stochastic part of Thomas-Fiering modeling[14], $T_{v,t+1}$:

$$T_{v,t+1} = \epsilon_{v,t} S_{t+1}\sqrt{1 - r_t^2} \tag{8}$$

Where:

- $\epsilon_{v,t}$ = is a random normal deviate with zero mean and unit variance
- s_{t+1} =is the standard deviation of the normalized and standardized precipitation and discharge in $t+1^{th}$month.
- r_t = the correlation coefficient between data in the $t^{th}$and $(t+1)^{th}$months.

The **Second Component:** $E_{v,t}$ is the deterministic component which is represented by the ESN-RNN architecture. The model can be resumed as the sum of both components above:

$$H_{v,t} = f(E_{v,t} + T_{v,t}) \tag{9}$$

Where, $H_{v,t}$ is the synthetic value obtained by our model. The function f is the inverse of pre-processing transformations defined in equations 10 and 12.

A. Data Pre-processing

In Probability and Statistical theory applied to hydrologic time series the basic assumption is which the variables, have to be normally distributed. Therefore, a transformation is required if time series do not meet this basic assumption. This transformation aims to remove the seasonality from the mean and the variance. In the literature, this operation is called seasonal standardization or deseasonalizing, which results in normally distributed variables with zero-mean and unit standard deviation[9].

In this study, the series of MOPEX data set for the period of January, 1948 to 2000 from 4 river basins (see Table I) were employed. We found that MOPEX time series have to be transformed to reduce their biased skewness. The skewness coefficient was reduced using the following equation (log-transformation):

$$X_{v,t} = \log\left(H_{v,t} + c_t\overline{H_t}\right) \tag{10}$$

$$c_t = \frac{a}{g_t^2} \tag{11}$$

Where, $H_{v,t}$ is monthly observed data for month $t(t = 1,\ldots,12)$ and year $v(\nu = 1,\ldots,N)$. N is number of years from records series, $\bar{H}_t$ is the monthly average inflow for month t, a is a constant, for this study the value adopted was 0.8 resulted by trial and error data test, g_t is the skewness coefficient for the set $H_{1,t}, H_{2,t}, ..., H_{N,t}$ and $X_{v,t}$ are the normalized data, for year v and month t.

Table I
Geographical characteristics and statistics of four MOPEX river basins

MOPEX id	River	Area(km2)	Mean Annual precip(mm)	Mean Annual discharge(mm)
3179000	Bluestone	1020	1018	421
3364000	East Fork White	4421	855	376
3054500	Tygart Valley	2372	1166	745
1541500	Clearfield Creek	2170	827	179

The data used by stochastic component (Thomas-Fiering model) were standardized in monthly basis through:

$$Y_{v,t} = \frac{X_{v,t} - \bar{X}_t}{S_t} \tag{12}$$

Where, $Y_{v,t}$ is the standardized value for month t and year v. $\bar{X}_t$ and S_t are mean and standard deviation for month t.

1) Neural Network Architectures: The performance of our approach was compared with three models, two based on feedforward ANN and once statistical model such Thomas and Fiering. These models often require previous measured samples which must be considered as input data, that is why the first model (ANN-1) generates hydrological data for the present month using the data of past month, similarly, the second model (ANN-2) use two previous months in order generates data to the present month.

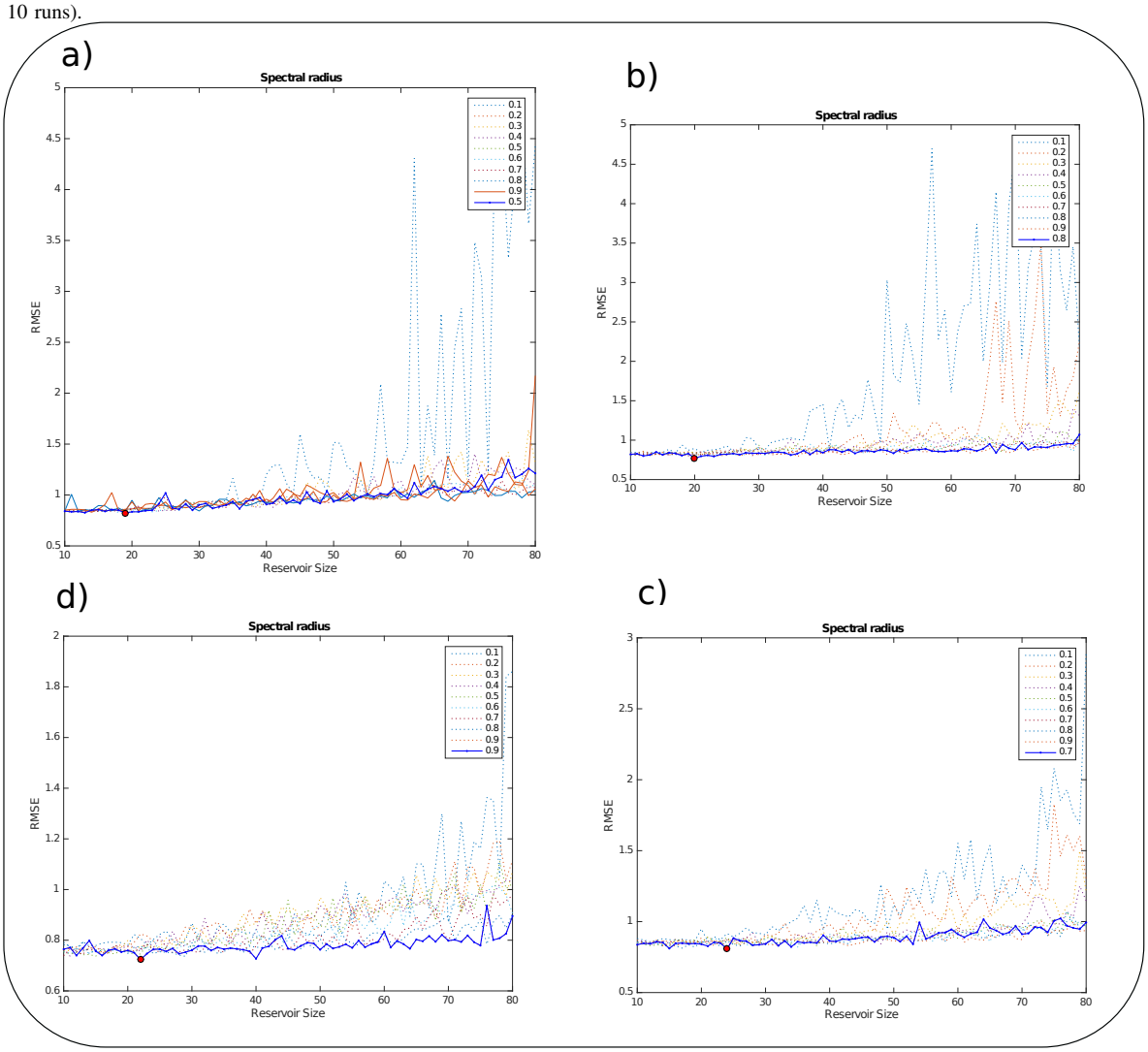
A typical three layer feedforward perceptron network architecture was adopted for ANN-1 and ANN-2. The tan-sigmoid functions were used as activation functions for which the output falls in$[-1, +1]$ range. The number of the hidden layers was decided by trial and error procedure.

Due to the feedback connections of ESN networks architecture include internal memory(internal states), it is not necessary previous input signals before further processing. We considered input and bias fully connected to the reservoir, the input weights to the reservoir were randomly selected from a uniform distribution where; $W^{in} \in [-0.1, 0.1]$. All other connection weights were selected from a normal distribution. A sparsely and randomly connected reservoir was used.

The number of the internal units of the reservoir and the spectral radius of the reservoir weights are important settings for ESN model[8]. Due to the above, performance tests in training of the networks for each river basin over values of **spectral radius** (from 0.1 to 0.9) and the **reservoir size** were made, see Figure 4.

From Figure 4, we can see that the bigger size of reservoirs generate lower performance in training ESN-RNN, for the 4 basins selected, each river basin was assigned with specific reservoir size(red points) and spectral radius(highlighted in blue), this is due to the characteristics and dynamic nature of the river are different.

Figure 4. ESN training performance of 4 MOPEX rivers, RMSE (y-axis) vs Reservoir size(x-axis), all over a range of values for the spectral radius from 0.1 to 0.9 (over 10 runs).



B. *Performance evaluation of Models*

In this study, 200 synthetic time series were generated with length of 24 months (2 years), The data were split up into training (1948–1999), and test (1998–2000). The performance of the forecasting models was evaluated in accordance with three error criteria: Normalized Root Mean Square Error (NRMSE), Mean Absolute Error (MAD), and Mean Percentage Error (MPE).

VI. Results and Discussion

The models used here (ANN1, ANN2, Thomas-Fiering and ESN-RNN) were trained and adjusted. During the training and testing periods of the ESN model, the entries of W were scaled with different spectral values for the 4 river basins, see Figure4. In order to avoid initial contamination, we have disregarded first responses (the states) of the reservoir. The reservoir sizes were defined as N1=17, N2=20, N3=22, N4=25, which presented the best performance value for each river basins, see the red points in the Figure 4.

We noted that ESN states have a wide dynamic range in forecast applications. These "**echo states**" are sets of functional bases constructed dynamically by the inputs, while the readout simply projects the desired response onto this representation space.

We can see, at Figures 5, 6 and 7, the synthetic data generated by ESN-RNN and Thomas-Fiering models, of discharge for 3 river basins respectively. The blue line refers to the observed measures of real time series, and the black lines refers to the forecast series of streamflow discharge(mm) to the period of 1998-2000.

Based on sections a.1) and b.1) of the Figure 5, 6 and 7, we can say that our model was able to learn and capture most of the variability behavior present on hydrological time series. A lower variance indicates that the training process had the desired effect capturing the dynamic of hydrological time series, which is not an easy task for any predictor. The ESN-RNN model obtained good results.

The Table II shows the forecasting errors obtained by each model used to compare our model(ESN-RNN, ANN-1,ANN-2 and Thomas-Fiering). We can see that ESN-RNN model, and these based on no-recurrent neural networks such as ANN-1 and ANN-2, performed better than Thomas-Fiering

Figure 5. Graphical evaluation of 200 synthetic time series, with testing patterns from 1998 to 2000 using the historical Streamflow discharge data of Bluestone river.

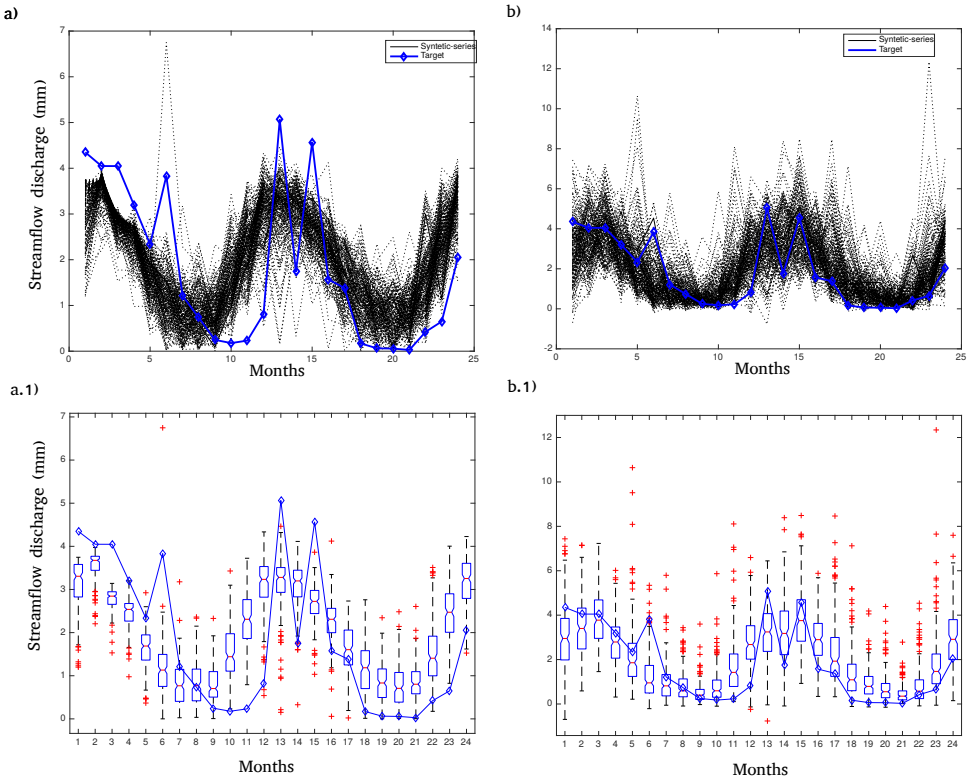


Figure 6. Graphical evaluation of 200 synthetic time series, with testing patterns from 1998 to 2000 using the historical Streamflow discharge data of Tygar Valley river

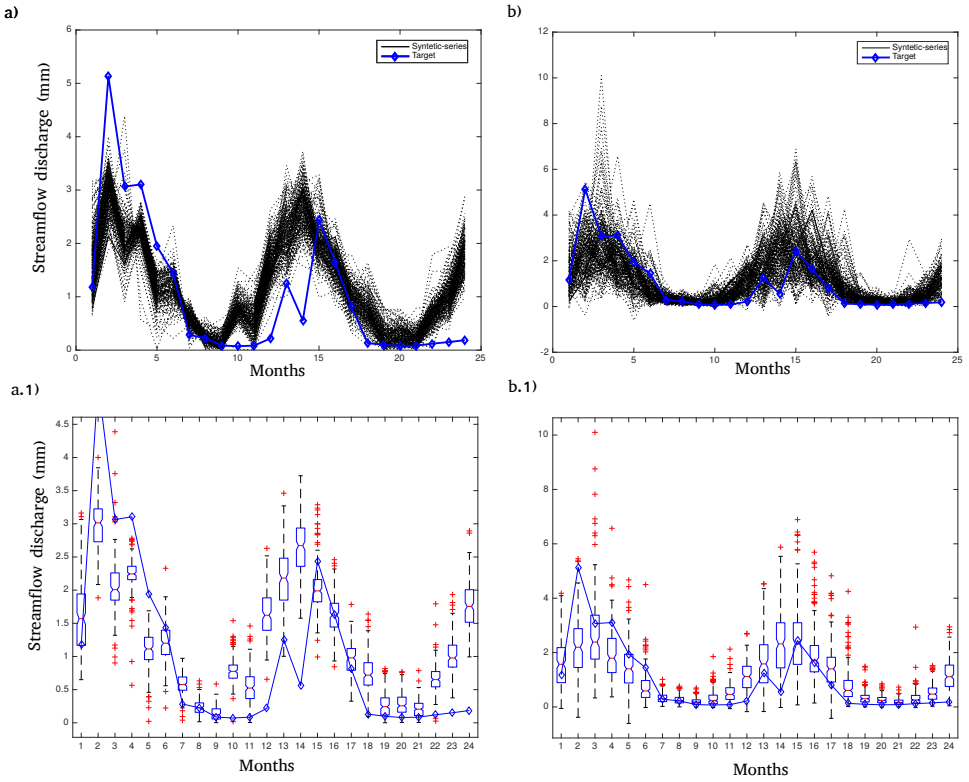
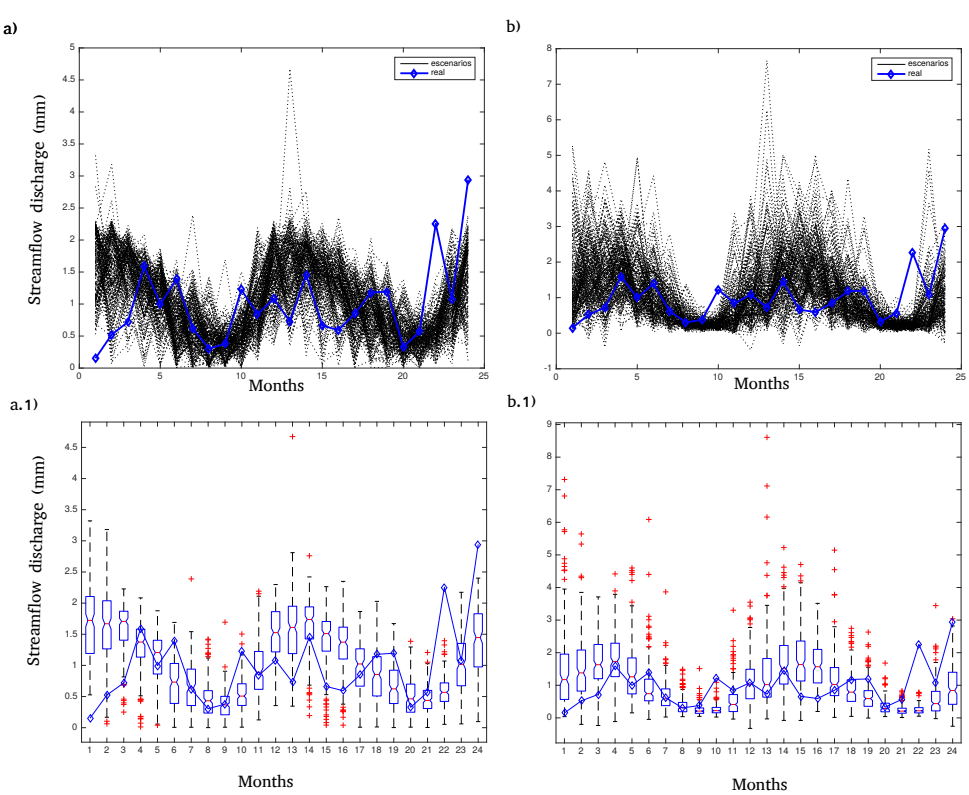


Figure 7. Graphical evaluation of 200 synthetic time series, with testing patterns from 1998 to 2000 using the historical Streamflow discharge data of East Fork White river.



model, which is a pure stochastic model. In addition, we note that ANN-1 and ANN-2 models obtain practically the same performance. Moreover, the ESN-RNN model performance was better than all other models, even though it presents a significantly simpler, and faster, training algorithm. This fact supports how powerful and promising this approach is.

VII. Conclusions

A new model is proposed as an alternative and effective tool for generation of monthly hydrological series. This model is composed of one deterministic component, defined as a Recurrent Neural Network trained with ESN algorithm. The other component is the stochastic part, defined as the white noise in order to take into account the uncertainty that normally affects natural processes.

This study presents a type of recurrent neural network called Echo State Network (ESN), which possesses a highly interconnected and recurrent topology. ESN has two interesting properties; the first one is that only the readout is trained, whereas the recurrent topology has fixed connection weights; the second one is that ESN has internal built-in memory, which are the result of feedback connections, thus it is not necessary to embed the input signal before further processing. Here, the ESN was used to forecast discharge. The project MOPEX of average monthly streaming inflows was used as source of training and test data.

For comparison, the Thomas-Fiering model was applied. The proposed hybrid model was also tested using two neural networks architecture, the first architecture (ANN-1) utilizes one previous month, and the second (ANN-2) utilizes two

Table II
Results summary of all methods on MOPEX dataset: 1) each row has the results of a specific method on a particular river basins; 2) each column compares the results of all methods with horizon value of 24 months.

ID River	Models	NRMSE	MAD	MPE(%)
3054500	ESN-RNN	0.67	0.91	216.91
	ANN-1	0.69	0.96	311.34
	ANN-2	0.70	0.96	313.23
	Thomas-Fiering	0.66	0.91	283.00
3364000	ESN-RNN	1.23	0.59	70.83
	ANN-1	1.45	0.69	118.22
	ANN-2	1.48	0.72	110.24
	Thomas-Fiering	1.33	0.62	99.87
3179000	ESN-RNN	0.62	0.56	169.79
	ANN-1	0.75	0.66	180.56
	ANN-2	0.80	0.70	195.43
	Thomas-Fiering	0.83	0.79	257.58
1541500	ESN-RNN	0.65	0.45	159.89
	ANN-1	0.73	0.52	175.23
	ANN-2	0.74	0.53	178.98
	Thomas-Fiering	0.73	0.51	165.37

previous months, in order to get the output of the model which is the next month. The results demonstrated that our model is better in performance compared to ANN-1, ANN-2 and Thomas-Fiering models. In addition the results also revealed that the proposed model performs better than pure stochastic Thomas-Fiering Model. The synthetic series produced by our model (ESN-RNN), they present low variability accompanied by few outliers in contrast with Thomas-Fiering Model. This fact shows that Recurrent model(ESN-RNN) features had the desired effect. It can be concluded that the proposed model is a promising alternative to be considered for more applications, competing with other linear pure stochastic autoregressive and traditional neural networks models. We can add to the above that the architecture of ESN permits to overcome several important drawbacks on traditional training methods of recurrent networks. This approach can lead to more accurate, reliable, and realistic models.

REFERENCES

[1] G. BOX, G.; JENKINS, "Time series analysis: Forecasting and control," 1970.

[2] I. Luna, R. Ballini, and S. Soares, "Tecnicas identificacao de modelos lineares e no-lineares de series temporais," *Sba: Controle e Automacao Sociedade Brasileira de Automatica*, vol. 17, pp. 245 – 256, 09 2006.

[3] L. C. D. Campos, M. M. B. R. Vellasco, and J. G. L. Lazo, "A stochastic model based on neural networks," in *The 2011 International Joint Conference on Neural Networks*, July 2011, pp. 1482–1488.

[4] D. D. J. . D. J. Maceira, M. E. P. ; Penna, "Geracao de cenarios sinteticos de energia e vazao para o planejamento da operacao energética." *In: XVI Simposio Brasileiro de Recursos Hidricos,*, 2005.

[5] H. Raman and N. Sunilkumar, "Multivariate modelling of water resources time series using artificial neural networks," *Hydrolog. Sci. J*, pp. 145–163, 1995.

[6] N. J. de Vos, "Echo state networks as an alternative to traditional artificial neural networks in rainfall runoff modelling," *Hydrology and Earth System Sciences*, vol. 17, no. 1, pp. 253–267, 2013. [Online]. Available: http://www.hydrol-earth-syst-sci.net/17/253/2013/

[7] D. Brezak, T. Bacek, D. Majetic, J. Kasac, and B. Novakovic, "A comparison of feed-forward and recurrent neural networks in time series forecasting," in *2012 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*, March 2012, pp. 1–6.

[8] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, August 2009.

[9] T. A. Awchi and D. K. Srivastava, "Artificial Neural Network Model Application in Stochastic Generation of Monthly Streamflows for Mula Project," 2004.

[10] Q. Duan, J. Schaake, V. Andréassian, S. Franks, G. Goteti, H. Gupta, Y. Gusev, F. Habets, A. Hall, L. Hay, T. Hogue, M. Huang, G. Leavesley, X. Liang, O. Nasonova, J. Noilhan, L. Oudin, S. Sorooshian, T. Wagener, and E. Wood, "Model parameter estimation experiment (mopex): An overview of science strategy and major results from the second and third workshops," *Journal of Hydrology*, vol. 320, no. 102, pp. 3 – 17, 2006, the model parameter estimation experimentMOPEXMOPEX workshop. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002216940500329X

[11] R. Sacchi, M. C. Ozturk, J. C. Principe, A. A. F. M. Carneiro, and I. N. da Silva, "Water inflow forecasting using the echo state network: a brazilian case study," in *2007 International Joint Conference on Neural Networks*, Aug 2007, pp. 2403–2408.

[12] X. Peng, R. Chen, C. Cheng, and X. Yan, "A folksonomy-ontology-based digital gazetteer service." in *Geoinformatics*. IEEE, 2010, pp. 1–6.

[13] K. P. Singh, "Streamflow synthesis by myron b. fiering; 139 pages; harvard university press, cambridge, massachusetts; 1967." *JAWRA Journal of the American Water Resources Association*, vol. 4, no. 2, pp. 70–70, 1968. [Online]. Available: http://dx.doi.org/10.1111/j.1752-1688.1968.tb02941.x

[14] M. B. THOMAS, H. A.; FIERINIG, "Mathematical synthesis of streamflow sequences for the analysis of river basins by simulation," pp. 459–493, 1962.

[15] A. Jain and S. Srinivasulu, "Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques," *Water Resources Research*, vol. 40, no. 4, pp. n/a–n/a, 2004, w04302. [Online]. Available: http://dx.doi.org/10.1029/2003WR002355

[16] N. J. de Vos and T. H. M. Rientjes, "Multiobjective training of artificial neural networks for rainfall-runoff modeling," *Water Resources Research*, vol. 44, no. 8, pp. n/a–n/a, 2008, w08434. [Online]. Available: http://dx.doi.org/10.1029/2007WR006734

[17] Y. Sun, D. Wendi, D. E. Kim, and S.-Y. Liong, "Technical note: Application of artificial neural networks in groundwater table forecasting a case study in a singapore swamp forest," *Hydrology and Earth System Sciences*, vol. 20, no. 4, pp. 1405–1412, 2016. [Online]. Available: http://www.hydrol-earth-syst-sci.net/20/1405/2016/

[18] B. Hammer, B. Schrauwen, and J. J. Steil, "Recent advances in efficient learning of recurrent networks," in *European Symposium on Artificial Neural Networks*, M. Verleysen, Ed. d-facto, 2009, pp. 213–226.

[19] G. Li, B.-J. Li, X.-G. Yu, and C.-T. Cheng, "Echo state network with bayesian regularization for forecasting short-term power production of small hydropower plants," *Energies*, vol. 8, no. 10, p. 12228, 2015. [Online]. Available: http://www.mdpi.com/1996-1073/8/10/12228

[20] P. Coulibaly, "Reservoir computing approach to great lakes water level forecasting," *Journal of Hydrology*, vol. 381, no. 1-2, pp. 76–88, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022169409007471

[21] R. J. Abrahart, N. J. Mount, and A. Y. Shamseldin, "Discussion of reservoir computing approach to great lakes water level forecasting by p.coulibaly[j. hydrol. 381(2010) 76-88]," *Journal of Hydrology*, vol. 422-423, pp. 76–80, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022169411007074

[22] A. A. F. M. C. R. Sacchi and A. F. R. Araujo, "A rbf network trained by the sonarx model and applied to obtain the operation policies of the hydropower systems," in *Brazilian Symposium on Neural Networks - SBRN*, Aug 2004, pp. 2403–2408.

[23] J. S. R. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, May 1993.

[24] K. Kuna, "Time Series Prediction using Neural Networks," p. 47, 2015.

[25] J. L. Elman, "Finding structure in time," *COGNITIVE SCIENCE*, vol. 14, no. 2, pp. 179–211, 1990.

[26] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989. [Online]. Available: http://dx.doi.org/10.1162/neco.1989.1.2.270

[27] D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986.

[28] I. Sutskever, "Training Recurrent neural Networks," *PhD thesis*, p. 101, 2013.

[29] N. J. de Vos and T. H. M. Rientjes, "Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation," *Hydrology and Earth System Sciences*, vol. 9, no. 1/2, pp. 111–126, 2005. [Online]. Available: http://www.hydrol-earth-syst-sci.net/9/111/2005/

[30] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, pp. 490–501, 1990.

[31] J. Sum, L.-w. Chan, C.-s. Leung, and G. H. Young, "Extended kalman filter-based pruning method for recurrent neural networks," *Neural Comput.*, vol. 10, no. 6, pp. 1481–1505, Aug. 1998. [Online]. Available: http://dx.doi.org/10.1162/089976698300017278

[32] K. Doya, "Bifurcations in the learning of recurrent neural networks 3," *learning (RTRL)*, vol. 3, p. 17, 1992.

[33] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004. [Online]. Available: http://science.sciencemag.org/content/304/5667/78