

Effects of Model Structural Complexity and Data Pre-Processing on Artificial Neural Network (ANN) Forecast Performance for Hydrological Process Modelling

Martins Yusuf Otache¹, John Jiya Musa¹, Ibrahim Abayomi Kuti¹, Mustapha Mohammed², Lydia Ezekiel Pam¹

¹Department of Agricultural & Bioresources Engineering, Federal University of Technology, Minna, Nigeria

²Raw Materials Research and Development Council, Abuja, Nigeria

Email: martynso_pm@futminna.edu.ng, drotachemartyns@gmail.com

How to cite this paper: Otache, M.Y., Musa, J.J., Kuti, I.A. Mohammed, M. and Pam, L.E. (2021) Effects of Model Structural Complexity and Data Pre-Processing on Artificial Neural Network (ANN) Forecast Performance for Hydrological Process Modelling. *Open Journal of Modern Hydrology*, 11, 1-18.

<https://doi.org/10.4236/ojmh.2021.111001>

Received: September 3, 2018

Accepted: January 26, 2021

Published: January 29, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The choice of a particular Artificial Neural Network (ANN) structure is a seemingly difficult task; worthy of relevance is that there is no systematic way for establishing a suitable architecture. In view of this, the study looked at the effects of ANN structural complexity and data pre-processing regime on its forecast performance. To address this aim, two ANN structural configurations: 1) **Single-hidden layer**, and 2) **Double-hidden layer** feed-forward back propagation network were employed. Results obtained revealed generally that: a) ANN comprised of double hidden layers tends to be less robust and converges with less accuracy than its single-hidden layer counterpart under identical situations; b) for a univariate time series, phase-space reconstruction using embedding dimension which is based on dynamical systems theory is an effective way for determining the appropriate number of ANN input neurons, and c) data pre-processing via the scaling approach excessively limits the output range of the transfer function. In specific terms considering extreme flow prediction capability on the basis of effective correlation: Percent maximum and minimum correlation coefficient ($R_{\max}\%$ and $R_{\min}\%$), on the average for one-day ahead forecast during the training and validation phases respectively for the adopted network structures: **8 7 5 (i.e., 8 input nodes, 7 nodes in the hidden layer, and 5 output nodes in the output layer)**, **8 5 2 5 (8 nodes in the input layer, 5 nodes in the first hidden layer, 2 nodes in the second hidden layer, and 5 nodes in the output layer)**, and **8 4 3 5 (8 nodes in the input layer, 4 nodes in the first hidden layer, 3 nodes in the second hidden layer, and 5 nodes in the output layer)** gave: 101.2, 99.4; 100.2, 218.3; 93.7, 95.0 in all instances irrespective

of the training algorithm (*i.e.*, pooled). On the other hand, in terms of percent of correct event prediction, the respective performances of the models for both low and high flows during the training and validation phases, respectively were: **0.78, 0.96: 0.65, 0.87; 0.76, 0.93: 0.61, 0.83**; and **0.79, 0.96: 0.65, 0.87**. Thus, it suffices to note that on the basis of coherence or regularity of prediction consistency, the ANN model: **8 4 3 5** performed better. This implies that though the adoption of large hidden layers vis-à-vis corresponding large neuronal signatures could be counter-productive because of network over-fitting, however, it may provide additional representational power. Based on the findings, it is imperative to note that ANN model is by no means a substitute for conceptual watershed modelling, therefore, exogenous variables should be incorporated in streamflow modelling and forecasting exercise because of their hydrologic evolutions.

Keywords

Streamflow, Neural Network, Phase-Space, Optimisation, Algorithm

1. Introduction

It is imperative to note as reported in Chibanga *et al.* [1] that the ability to predict events with reasonable accuracy enables one to plan in advance what course of action to take, get the best out of situation and avert probable deleterious ones. But to achieve this, the system analyst might prefer not to waste time and effort required to develop and implement a conceptual model but rather opt for a simpler system-theoretic model (Chibanga *et al.*, [1]). The system-theoretic models embrace artificial neural networks (ANNs); ANNs entail the application of difference or differential equation to identify a direct mapping between inputs and outputs without detailed consideration of the internal structure of the physical processes (Abrahart and See, [2]). Thus, in the views of Chibanga *et al.* [1], ANN model structure, being a flexible mathematical structure capable of identifying complex nonlinear relationships between input and output sets, stands out as one of its principal characteristics of vital importance.

According to Abrahart [3], system-theoretic/black box or neural network forecasting and prediction offer various benefits ahead of the traditional conceptual modelling; some of the benefits in line with the guiding principles are parsimony, modesty, and testability (Hillel, [4]). However, though an artificial neural network (ANN) is a flexible mathematical structure that is capable of identifying complex nonlinear relationships between input and output data sets, its overall performance is dependent on a whole lot of associated variables. But since an ANN does not depend solely upon the physical parameters used in the analytical approach, it could be designed with much different architecture to achieve optimal performance. As reported in Balkhair [5], the performance of ANN is sensitive to its physical architecture, such as the number of input nodes, hidden layer nodes, and output nodes; *i.e.* the appropriate architecture of ANN

is highly problem dependent. This implies that as in the identification of other nonlinear types of models (system-theoretic or conceptual), 1) a model structure must be identified, and 2) the model parameters must be calibrated through an iterative procedure that employs an objective function surface in search of an optimum (Hsu *et al.*, [6]).

The selection of an appropriate architecture is usually problematic. In the views of Abrahart and See [2], there is no single correct procedure to determine the optimum of units or layers, although one or two “rules of thumb” have been put forward (e.g. Sarle, [7]) and various automated growing, pruning and network breeding algorithms exist (e.g. Fahlman and Lebiere, [8]; SNN Group, [9]). In this regard, selection of the optimal number of hidden units is worth mentioning; often, it is considered to be problem dependent. Though the number of hidden units and layers will control the power of the model to perform more complex modelling but as noted by Abrahart and See [2], this is not without its associated trade-off. It has been noted (Abrahart and See, [2]) that the use of large hidden layers also could be counter-productive because an excessive number of free parameters will encourage over-fitting of the network solution to the training data and thus reduce the generalisation capabilities of the final product. Thus, the general concern is how to find a balance between the hidden units between the layers as well as the number of hidden layers.

In light of the preceding sections, considering that the ANN model structure is ideally suited for modelling highly nonlinear input-output relationships, the central thrust of this study, therefore, is to assess the implication(s) of some of the latent issues in the application or adoption of neural network modelling paradigm or approach; specifically, in this regard, the emphasis is on model structural complexity and implicitly bring to the fore the correlation between optimisation algorithm as well as data pre-processing regime.

2. Methodology

2.1. Data Collection and Management

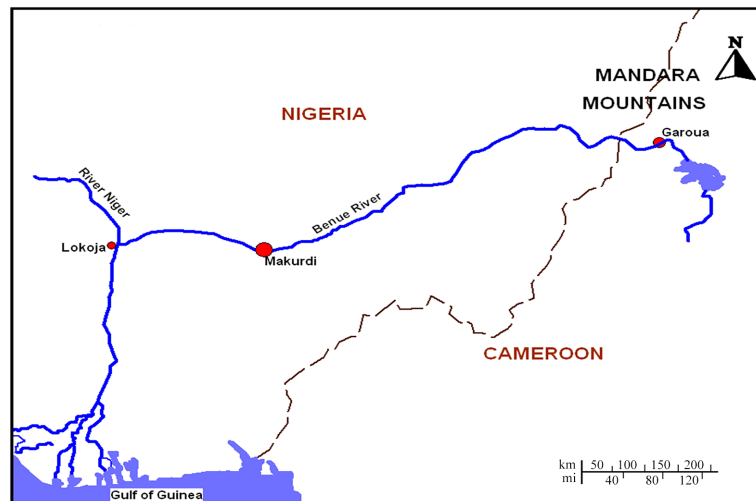
For this study, daily streamflow sequence of the River Benue at the Makurdi hydrometric station was obtained from the Benue State Water Works and National Inland Waterways Authority (Makurdi Office); the data sequence spanned through an entire period of thirty (30) years. Consistency test and continuity tests were done; based on these tests, non-continuous data years were removed thus reducing the length to 26 years (*i.e.*, 9490 data elements). The entire time series of length of 9490 daily values was thus partitioned into two-set constituents of 8670 and 730 data points corresponding to training and validation phases, respectively; *i.e.*, split sampling approach. **Figure 1** shows the traverse of River Benue and the location of the Makurdi hydrometric station. The River is perennial and exhibits high seasonality with peak flow regimes usually in the months of September and October. It takes its source from the Cameroon high lands; in conjunction with River Niger, they drain the larger part of the country and vastly impart the hydrological evolutions of their respective flood plains, albeit the en-

tire country to a great extent.

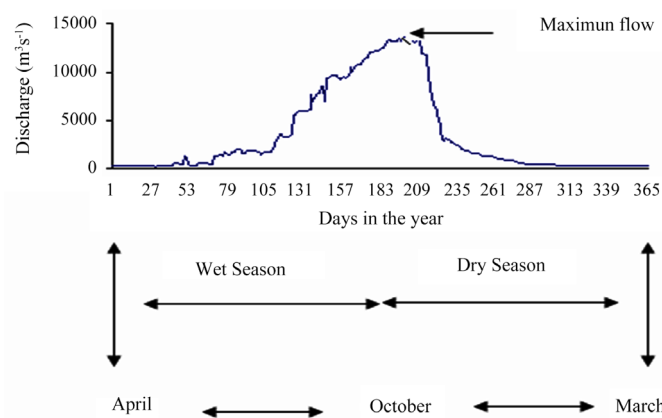
2.2. Development of ANN Model

2.2.1. Preliminary Analysis of the Daily Streamflow/Discharge Dynamics

The mean daily discharges are as shown in **Figure 2**; the discernible features are the flood peaks and the seasonal periodicity, with both large and small discharges during the year. Besides these general characteristics, the study of the autocorrelation and the spectral analysis provide the first important indications on the aperiodicity of the signal under examination. **Figure 3** shows the autocorrelation for values of the delay time (lag) between 1 day and about 5 years; after a rapid decrease, the autocorrelation function displays a regular behaviour, which represents the effect of the seasonal characteristic of the discharges, due, besides the rainfall regime, to other hydrologic forcing which breathes with the season. Contrary to expectation, there is no brisk fall of the autocorrelation as shown in **Figure 4**, which might indicate a complex behaviour, having characteristic timescales of a few days.



(a)



(b)

Figure 1. Basic details of River Benue: (a) Map of Nigeria showing River Benue and its traverse; (b) Flow regime of River Benue at Makurdi hydrometric section.

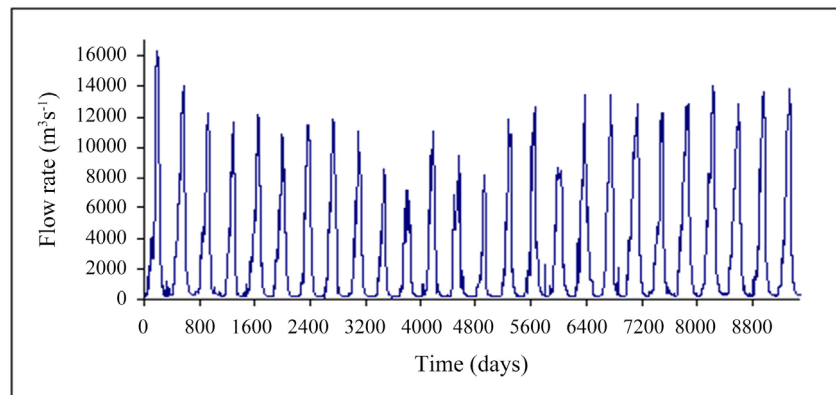


Figure 2. Time series plot of the daily discharge.

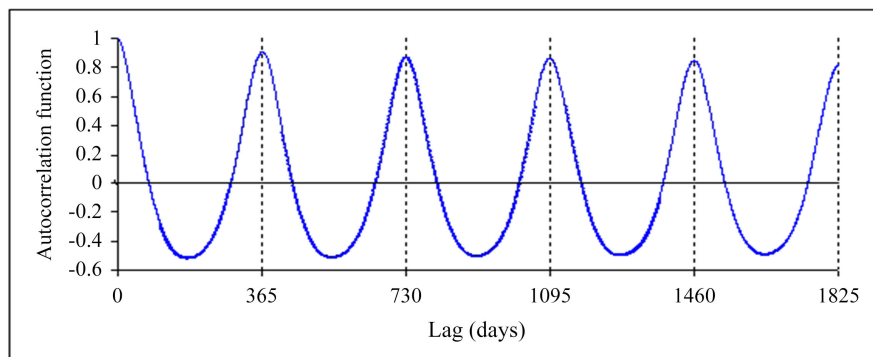


Figure 3. Autocorrelation of the daily discharge series showing sinusoidal pattern.

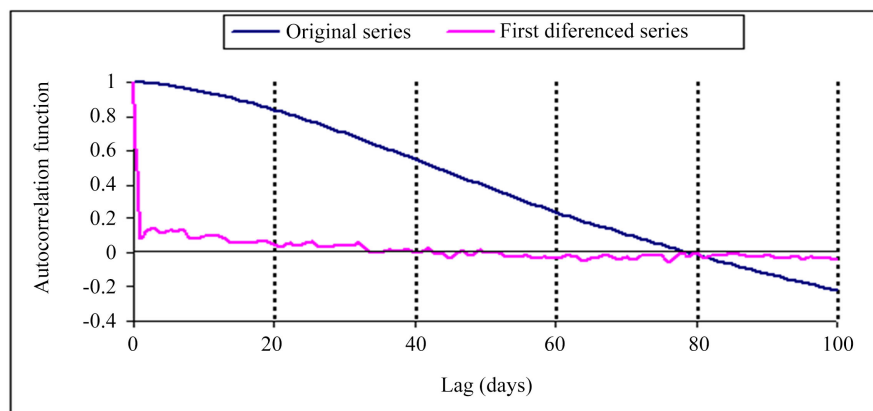


Figure 4. Blow out of the autocorrelation of the discharge series and the first differenced series.

Similarly, the autocorrelation of the first difference signal (**Figure 4**) does not show much of a rapid monotonic decrease in autocorrelation. Despite this though, apart from the periodicity, the behaviour of the series appears essentially aperiodic and erratic; for example, once the annual periodic component has been filtered out from the series through least squares methods, the spectral density (**Figure 5**) exhibits a typical broadband behaviour. The spectrum does not show any privileged frequencies, but rather a linear decay that links the

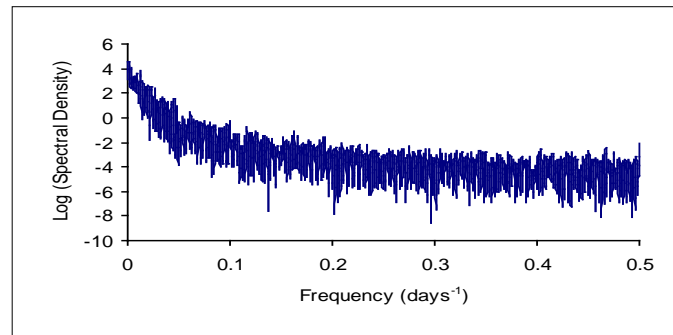


Figure 5. Spectral density of the filtered discharge series (*i.e.*, after the removal of annual periodicity).

whole range of frequency components. The fact that the spectrum is continuous with a pronounced and wide base underscores the aperiodicity of the series; but the problem however is, how much does the character of this complex aperiodicity and irregularity may translate to complex nonlinearity. This therefore portends the need to investigate if the dynamics of the discharges of a river could have a dominant chaotic signature on which high-dimension linear and nonlinear dynamics may be grafted.

2.2.2. Reconstruction of Phase-Space (Attractor) by Time Delay Embedding

The first step in the search for a deterministic behaviour is that of attempting to reconstruct the dynamics in phase space. Having available the time series of only one of the variables present in the phenomenon, that is, the discharge $x(t_i)$, the delay time method proposed by Takens [10] and Packard *et al.* [11] can be used to reconstruct the attractor; this is based on the fact that the interaction between the variables is such that every component contains information on the complex dynamics of the system. Choosing a delay time τ , usually a multiple of sampling period Δt , the method entails the construction of a series of $M = (m-1)\tau/\Delta t$ vectors, of the dimension m , of the form

$$x(t_i) = \{x(t_i), x(t_i + \tau), \dots, x[t_i + (m-1)\tau]\} \quad (1)$$

where, $m(m = 2, 3, \dots)$ is called the embedding dimension.

To construct a well-behaved phase-space by delay time, a careful choice of τ is critical. The delay time τ is commonly selected by using the autocorrelation function (ACF) method where ACF first attains zeros or drops below a small value, say $1/e^4$, or the mutual information (MI) method according to Fraser and Swinney [12] where the MI attains a minimum. Here, the delay time τ is taken as the lag that first generates a zero autocorrelation, which is when the autocorrelation function crosses the zero line (Mpitsos *et al.*, [13]). In practice, the estimate of τ is usually application and author dependent; for instance, some authors take the delay time as 1 day (Porporato and Ridolfi, [14]), 2 days (Jayawardena and Lai, [15]), 7 days (Islam and Sivakumar, [16]), 10 days (Elshorbagy *et al.*, [17]), 20 days (Wilcox *et al.*, [18]), 91 days (Wang, [19]), and 146 days (Pasternack, [20]). These differences may arise from the nature of the autocorrelation function; to

compare the influence of the delay time τ on the construction of state-phase, the state-phase maps can be plotted for differing τ values. The best τ value should make the state-phase plot best unfolded. Towards this end, the state-phase map is constructed for different τ values ($\tau = 1, 7, 10, 30,$ and 78), these are as displayed in **Figure 6** along with the 3-dimensional state-phase map based on τ equals to 78 , *i.e.*, when the autocorrelation function first crossed the zero line (**Figure 6**). **Figure 6** shows that the best unfolding can be obtained when $\tau = 78$; therefore, $\tau = 78$ is adopted for estimating the correlation dimension of the daily streamflow process in this study.

2.2.3. Network Topology

The time delay coordinate method (Packard, *et al.* [11]; Takens, [10]) was used to reconstruct the phase-space from the scalar time series; in this case, because of the nature of the data; *i.e.*, a univariate series. This is informed by the fact that to describe the temporal evolution of a dynamical system in a multi-dimensional phase-space with a scalar time series, there is need to employ some techniques to unfold the multi-dimensional structure using the available data. Thus, the approach in the proceeding section is further complimented by applying the method for the determination of minimal sufficient dimension (m) as proposed by Kennel *et al.* [21], called the “**False Nearest Neighbour (FNN) method**”. That is, supposing the point $X_i = [Y_{i-p+1}, \dots, Y_i]$ has a neighbour $X_j = [Y_{j-p+1}, \dots, Y_j]$ in a p -dimensional space then the distance $\|X_i - X_j\|$ is calculated in order to compute:

$$R_i = \frac{|Y_{i+1} - Y_{j+1}|}{\|X_i - X_j\|} \quad (2)$$

If R_i exceeds a given threshold R_r (a suitable value is $10 \leq R_r \leq 50$), the point X_i is marked as having a *False Nearest Neighbour*. As a consequence, the embedding dimension p is high enough if the fraction of points that have *False Nearest Neighbours* is actually zero, or sufficiently small, say, smaller than a criterion R_f . In this case, the *False Nearest Neighbour* threshold R_r was set to 10 (as reported in Wang [19]). Based on this, the fraction of *False Nearest Neighbours* as a function of the embedding dimension was calculated based on phase-space reconstruction using embedding dimension. Here, the minimal embedding dimension was taken as 8; this implies that the state of the streamflow process can be determined by eight lagged observed values as shown in **Figure 7**.

Following from the analysis, eight lagged values of input variables were used when fitting the ANN model to the series; specifically, this implied that based on the phase-space reconstruction, the discharges $Q_{t-7}, Q_{t-6}, \dots, Q_t$ of day $t-7$ to day t . The eight lagged input values were used to forecast the discharge from time $t + 1$, *i.e.*, the **next day**, to $t + 5$; *i.e.*, **5-ahead** values, using a multiple-output approach rather than a single-output. The idea here is just to explore the ANN model forecast behaviour over a high lead time (**Figure 8**).

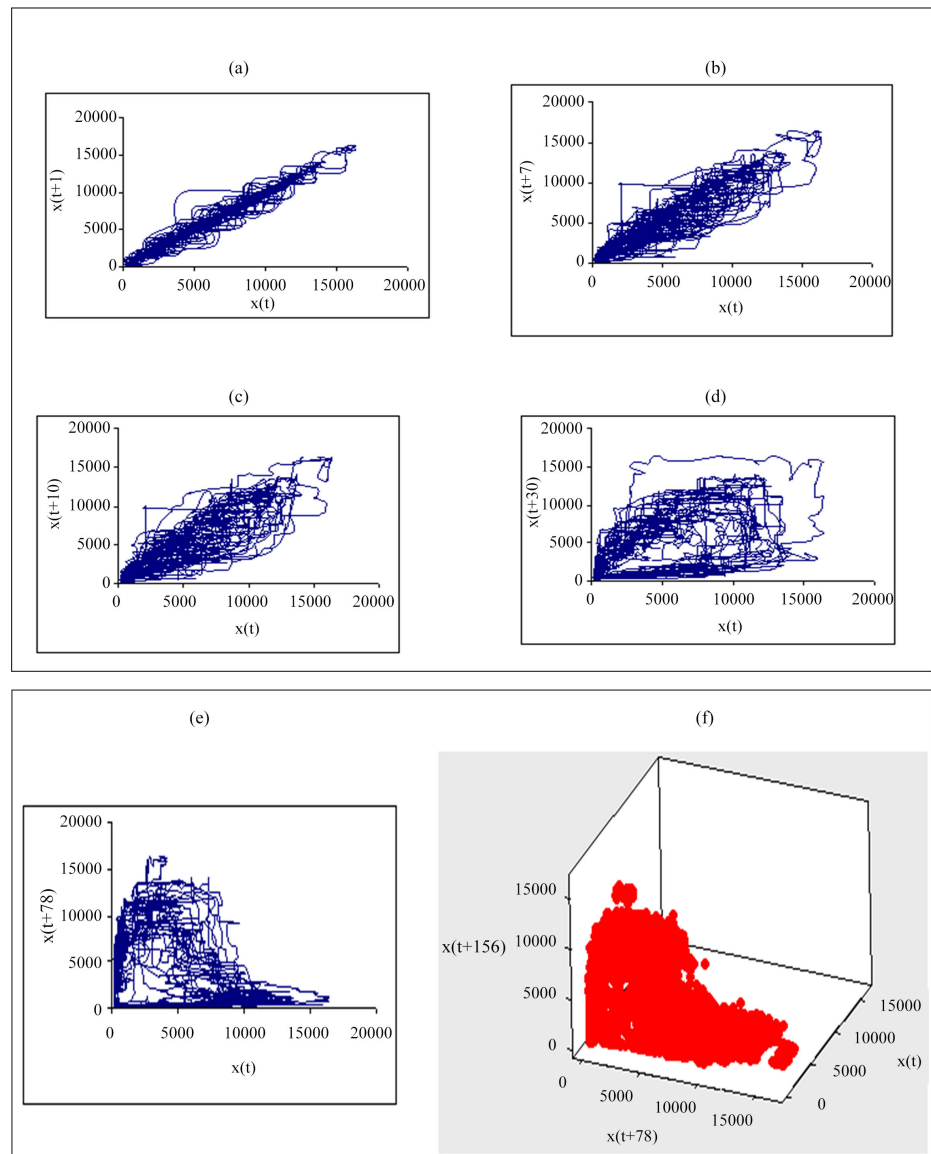


Figure 6. Phase – space schematic of raw average daily discharge series: (a) Delay time = 1 day; (b) Delay time = 7 days; (c) Delay time = 10 days; (d) Delay time = 30 days; (e) Delay time = 78 days; (f) 3-D phase space map using delay time = 78 days.

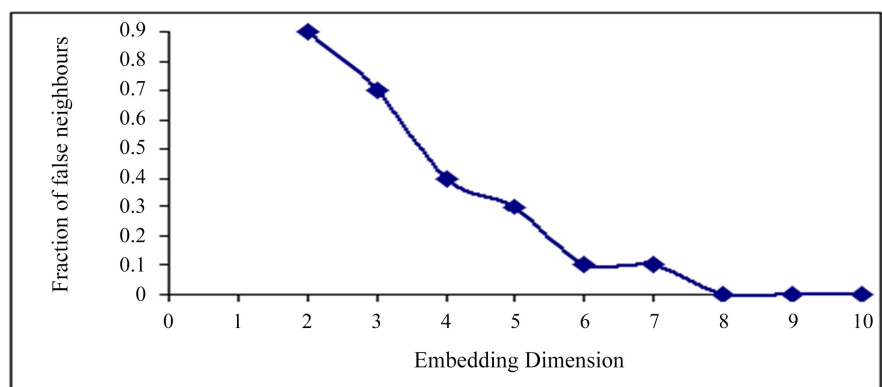


Figure 7. Fraction of false nearest neighbours as a function of embedding dimension.

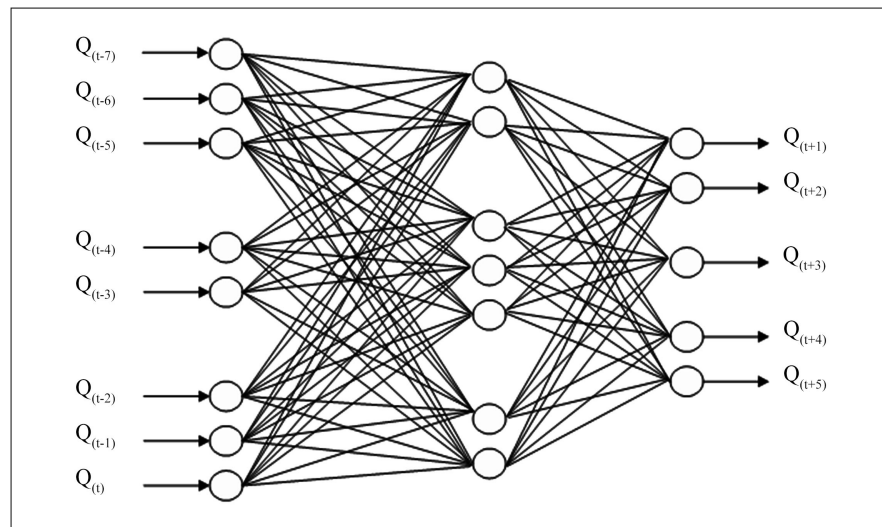


Figure 8. Schematic of three-layer feedforward artificial neural network architecture.

To address the thrust of the study, two model configurations were considered corresponding to two model architectural variants with different nodal configurations. Precisely, single and double hidden layers were thus considered to examine the implications of model structural complexity. The ANN models adopted were 1) **8 7 5** single-hidden layer with 7 nodes; *i.e.*, 8 input nodes in the input layer, 7 nodes in the hidden layer, and 5 output nodes in the output layer 2) **8 5 2 5** double-hidden layers with 5 and 2 nodes, respectively and 3) **8 4 3 5** double-hidden layers with 4 and 3 nodes, respectively; though after several trials in an attempt to choose comparable network structures.

2.2.4. Network Training

For the purposes of the stated aim of the study, the multi-layer feedforward back propagation network was used. Specifically, network training was implemented using the *trainbr* (**Bayesian regularisation: Br**) function, *traingdm* (**Gradient descent with momentum: gdm**) function, *trainlm* (**Levenberg-Marquardt: lm**) function in MATLAB Neural Network Toolbox. Since in neural network training, the transfer function is of critical relevance and predictability of future behaviour is a direct consequence of the correct identification of it, for the identified network structure, the *tansigmoid* and *purelin* transfer functions were used in the hidden and output layers, respectively. The *purelin* transfer function was considered for the output layer because it allows the network outputs to take on any value, whereas the last layer of a multi-layer network with sigmoid neurons constrains the network outputs to a small range.

Before applying the ANN, both input and output data were pre-processed and normalised in the range $[-1 \ 1]$. The scaling strategy was adopted based on the findings of Wang [19] and Otache [22]; rescaling was done to scale the data series to fall within this bound. Scaling of the original data, say x'_t to the network range was done by

$$x_t = \frac{(U_x - L_x)x'_t + (M_x L_x - m_x U_x)}{M_x - m_x} \quad (3)$$

where, x'_t = the original input data, x_t = the input data scaled to the network range, M_x and m_x are respectively the maximum and the minimum of the original input data, while U_x and L_x are the upper and the lower network ranges for the network input, respectively. Similarly, the original output, say y'_t is scaled to the network range by

$$y_t = \frac{(U_y - L_y)y'_t + (M_y L_y - m_y U_y)}{M_y - m_y} \quad (4)$$

where, y_t the systems' output is scaled to the network range, M_y and m_y are respectively the maximum and minimum values of the original output data y'_t , whereas U_y and L_y are respectively the upper and the lower network ranges for the network output. After scaling the inputs and outputs, the resulting output, say \hat{y}_t is in the scaled domain. Hence, there is need to rescale the output \hat{y}_t back to its original domain; this is by inverting Equation (3) and using \hat{y}'_t as

$$\hat{y}'_t = \frac{(M_y - m_y)\hat{y}_t - (M_y L_y - m_y U_y)}{U_y - L_y} \quad (5)$$

2.3. Forecast Analysis

In order to draw conclusions on the ANN model performance, attention is on the ANN model performance in terms of extreme events, that is, maximum and minimum flows. In this regard, the coefficient of correlation R as in Equation (6) was employed.

$$R = \frac{\frac{1}{v} \sum_{t=1}^v [y_t - \mu_y][\tilde{y}_t - \tilde{\mu}_y]}{\left[\frac{1}{v} \sum_{t=1}^v [y_t - \mu_y]^2 \right]^{1/2} \left[\frac{1}{v} \sum_{t=1}^v [\tilde{y}_t - \tilde{\mu}_y]^2 \right]^{1/2}} \quad (6)$$

where, v = the number of output data points, y_t = the observed flow, \tilde{y}_t = predicted flow, μ_y = mean of observed flow, and $\tilde{\mu}_y$ = mean of predicted flows. In terms of the measures of forecast accuracy with respect to extreme values, the ratio of the forecasted maximum to the observed maximum (peak) was determined as

$$R_{\max} = \frac{\tilde{y}_t}{\max\{y_t\}} \times 100 \quad (7)$$

where, $\max\{y_t\} = \max\{y_1, \dots, y_v\}$ and \tilde{y}_t is the forecast corresponding to such maximum; and $R_{\max} = 100\%$, means that the observed peak is perfectly reproduced by the model. Forecasts with values of R_{\max} about 100% are considered to be very accurate, while $R_{\max} < 100\%$ indicates that the model underestimates the peak value; and $R_{\max} > 100\%$ indicates overestimation. Similarly, the ratio of the forecasted to the observed minimum

$$R_{\min} = \frac{\tilde{y}_t}{\min\{y_t\}} \times 100 \quad (8)$$

where, \tilde{y}_t represents the forecast corresponding to the minimum observed value was used to judge the forecasting capability of the model. In addition, specific-event prediction was also considered by looking at low and high flows.

3. Discussion

3.1. Effects of Model Structural Complexity

Figures 9-12 and **Table 1, Table 2** clearly show the performance of the network configurations in terms of correlation, extreme flow and event-specific evaluation for lead times of one and five-day ahead predictions, respectively. It is glaring that the complexity of the network structure may impair the integrity of the network performance as depicted by the contrasting results both in the training and validation phases. The overall performance, looking at **Figures 9-12** shows that while the network demonstrated the capacity to predict low flows fairly well as well as elements of high flows, the converse is the case for a range of medium flows. In view of the statistics as in **Table 1** and **Table 2** as well as **Figures 9-12**, the aggregate performance of **ANN Model: 8 4 3 5** is relatively better followed by **ANN Model: 8 7 5**. The **ANN Model: 8 5 2 5** performed abysmally, especially with staggering values of R_{\max} % (489.5 and 423) and R_{\min} % (0.0 and 0.0) in the validation phase (See **Table 1**); **Figure 10** clearly shows this behaviour, especially **Figure 10** where there is strong evidence of zero flow prediction as well as high spikes probably due to poor data quality leading to outliers; though in all instances, as noted in **Figure 9** and **Figure 11**, the models could explain between 60% and 97% (**exemplified by R^2 values**) of the variability in the streamflow dynamics.

Based on the results obtained, it suffices to note that the structural complexity is defined here to connote the size of the hidden layers against the traditional one-hidden layer commonly employed. It is thus imperative to state that the selection of the optimal number of hidden units (nodes) for the hidden layer is often considered to be problem dependent. However, intuition suggests that more is better but as reported by Abrahart and See [2], it is not always the case; *i.e.*, the number of hidden units and layers control the power of the network to perform complex modelling but with associated trade-off between training time and network performance. The findings here accord with the submissions of Abrahart and See [2] that the use of large hidden layers could be counter-productive because an excessive number of free parameters will encourage over fitting of the network solution to the training data and invariably reduces the generalisation capabilities of the final product. Hence, worthy of relevance is the effective number of hidden layers and how to establish a balance between the hidden units of these layers as shown by the results in **Table 1** and **Table 2**. Though as pointed by Hornik *et al.* [23], the performance of a one-hidden layer could be compromised due to data quality; *i.e.*, if the data contain an insufficient deterministic relationship. This really places a searchlight on the viability of a univa-

riate series as employed here; the possibility of compromising any meaningful dependence in terms of casual correlation between input variables is high. However, in line with the findings of Openshaw and Openshaw as reported in Abraham and See [2], there could be some advantages in the adoption of more than one-hidden layer; as noted here (See Table 1, Table 2, and Figure 9 and Figure 12), the use of two-hidden layers could provide an additional degree of representational power. It is obvious that the selection of too many hidden units or neurons may increase the training time without significant improvement on training results. Generally, in agreement with the findings of Ranjithan and Eheart [24], since too many hidden neurons probably may encourage each hidden neuron to memorise one of the input patterns, the error of the training set decreases gradually with an increasing number of intermediate units but the generalisation of the network may reach an optimum and does not necessarily improve indefinitely with an increasing number of intermediate units or nodes.

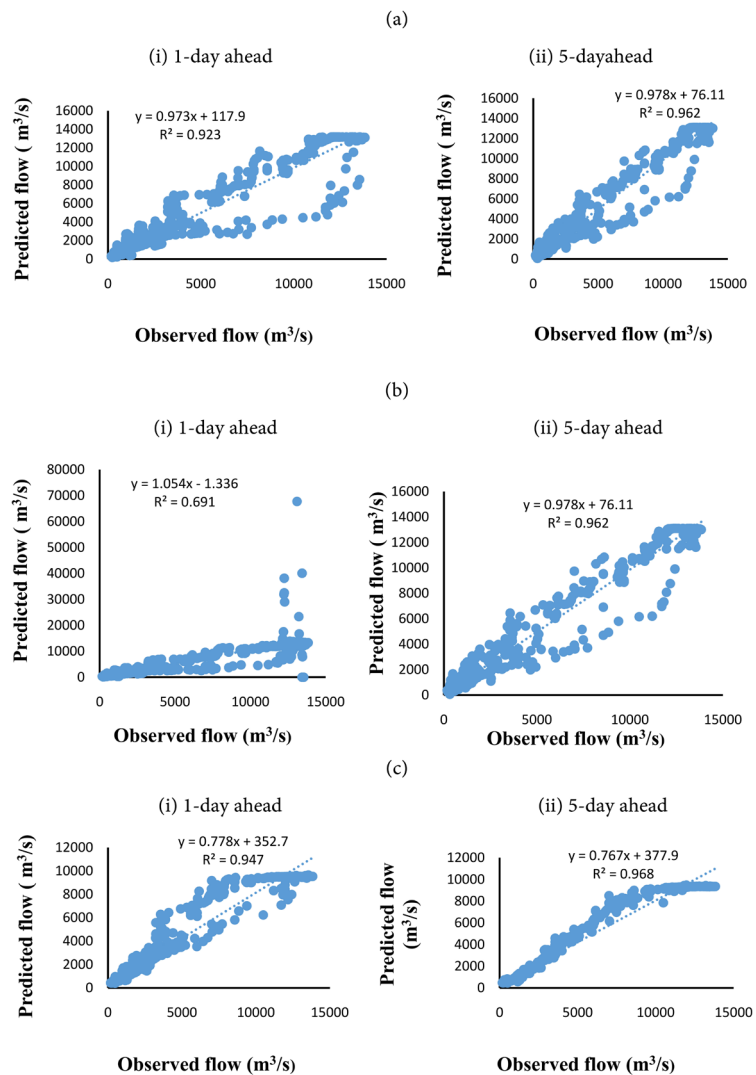


Figure 9. ANN 8 5 2 5 Model prediction correlation plot for (a) Br, (b) LM, and (c) GDM in the validation.

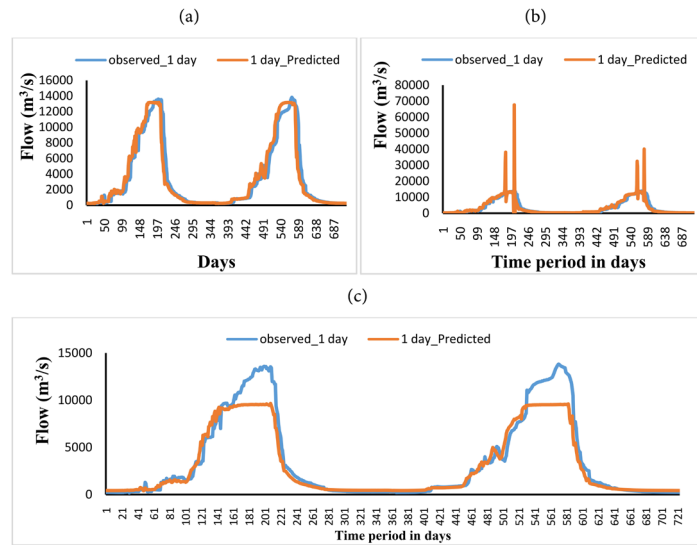


Figure 10. Characteristic ANN 8525 Model flow simulation pattern for 1-day ahead lead time for (a) Br, (b) LM, and (c) GDM optimisation algorithms.

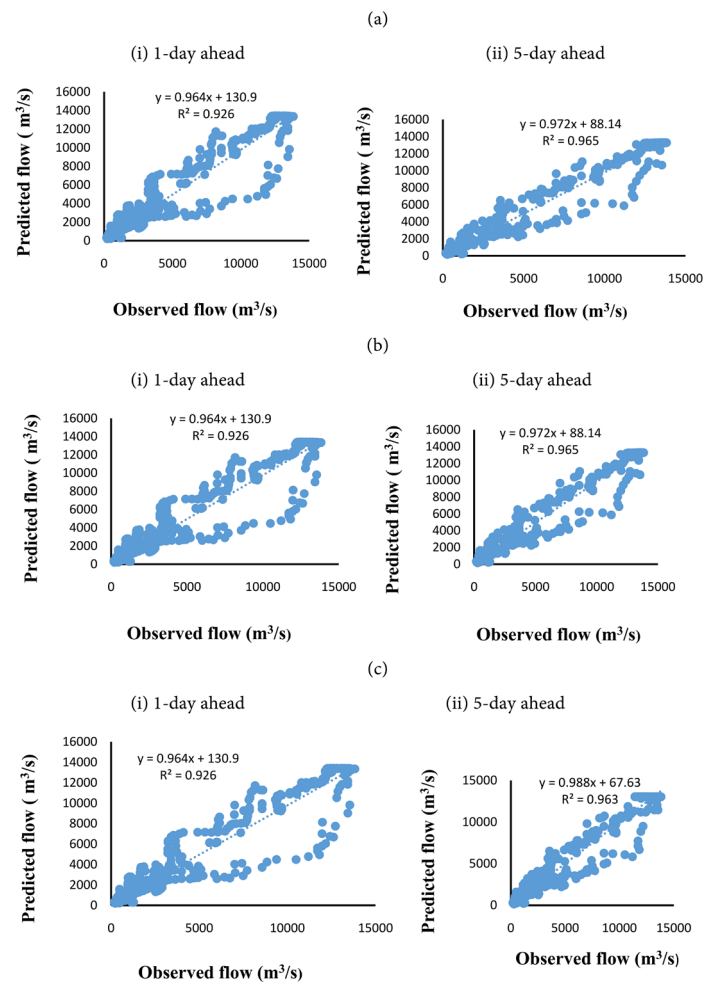


Figure 11. ANN 8 4 3 5 Model prediction correlation plot for (a) Br, (b) LM, and (c) GDM in the validation.

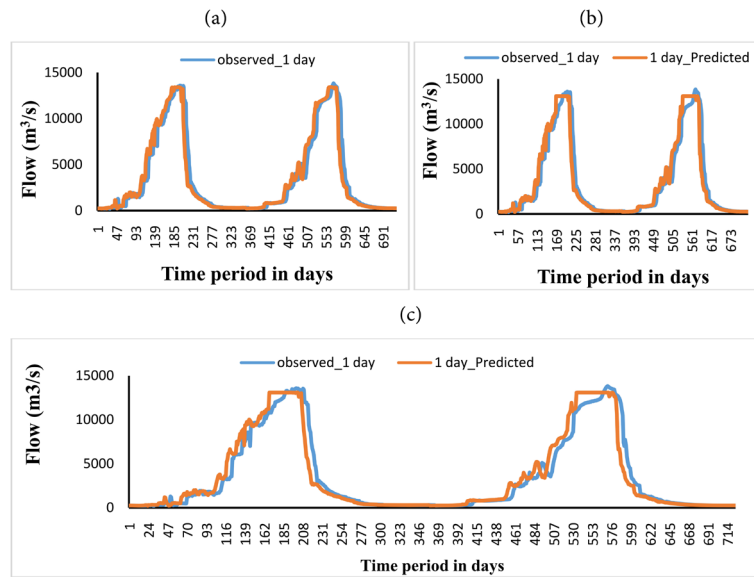


Figure 12. Characteristic ANN 8 4 3 5 Model flow simulation pattern for 1-day ahead lead time for (a) Br, (b) LM, and (c) GDM optimisation algorithms.

Table 1. Effective correlation (R) statistics of extreme flow predictions.

Model Architecture: (8 7 5)									
General	Optimisation algorithm	Training				Validation			
		R_{max} (%)		R_{min} (%)		R_{max} (%)		R_{min} (%)	
		1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
	Br	99.8	96.8	98.5	66.0	98.5	97.3	82.2	22.1
	LM	98.9	96.4	125.8	0.0	100.6	100.9	84.9	60.8
	GDM	104.8	101.2	0.0	0.0	99.2	88.6	16.4	0.0
Average		101.2	98.1	74.8	22.0	99.4	95.6	61.2	27.6
Model Architecture: (8 5 2 5)									
General	Optimisation algorithm	Training				Validation			
		R_{max} (%)		R_{min} (%)		R_{max} (%)		R_{min} (%)	
		1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
	Br	95.9	94.4	10.9	54.9	95.5	94.6	104.6	34.9
	LM	99.9	98.1	84.8	0.0	489.5	423.0	0.0	0.0
	GDM	104.8	101.2	0.0	0.0	69.8	68.5	0.0	0.0
Average		100.2	97.9	31.9	18.3	218.3	195.4	34.9	11.6
Model Architecture: (8 4 3 5)									
General	Optimisation algorithm	Training				Validation			
		R_{max} (%)		R_{min} (%)		R_{max} (%)		R_{min} (%)	
		1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
	Br	98.5	95.5	110.36	66.2	97.0	96.1	115.0	89.7
	LM	101.2	101.2	107.2	0.0	94.6	94.4	110.3	87.1
	GDM	81.5	75.8	70.9	30.7	94.6	94.4	110.3	87.1
Average		93.7	90.8	96.1	32.3	95.4	95.0	111.8	88.0

Table 2. Event-specific evaluation.

Network Topology (8 7 5: Single hidden layer)								
Optimisation algorithm	Training				Validation			
	% correct event prediction				% correct event prediction			
	Low flow		High flow		Low flow		High flow	
	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
Br	0.82	0.73	0.98	0.89	0.65	0.69	0.87	0.86
LM	0.81	0.72	0.98	0.88	0.65	0.72	0.87	0.87
GDM	0.72	0.68	0.91	0.83	0.65	0.72	0.87	0.87
Average	0.78	0.71	0.96	0.87	0.65	0.71	0.87	0.87
Network Topology (8 5 2 5: Double hidden layer)								
Optimisation algorithm	Training				Validation			
	% correct event prediction				% correct event prediction			
	Low flow		High flow		Low flow		High flow	
	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
Br	0.81	0.73	0.97	0.89	0.65	0.70	0.88	0.87
LM	0.81	0.73	0.98	0.89	0.65	0.69	0.88	0.86
GDM	0.67	0.70	0.84	0.84	0.53	0.56	0.73	0.78
Average	0.76	0.72	0.93	0.87	0.61	0.65	0.83	0.84
Network Topology (8 4 3 5: Double hidden layer)								
Optimisation algorithm	Training				Validation			
	% correct event prediction				% correct event prediction			
	Low flow		High flow		Low flow		High flow	
	1-day	5-day	1-day	5-day	1-day	5-day	1-day	5-day
Br	0.82	0.73	0.98	0.89	0.65	0.70	0.88	0.87
LM	0.81	0.72	0.98	0.88	0.65	0.72	0.87	0.87
GDM	0.73	0.68	0.91	0.83	0.65	0.72	0.87	0.87
Average	0.79	0.71	0.96	0.87	0.65	0.71	0.87	0.87

3.2. Implications of Data Pre-Processing Strategy

It is paramount not to only evaluate model forecast performance on the basis of statistical parameters, but to also consider the impact *data pre-processing* may have on ANN model forecasts. It is recognised that data pre-processing can have a significant effect on model performance (e.g. Maier and Dandy, [25]). It is commonly considered that, because the outputs of some transfer functions are bounded, the outputs of a Multi-Layer Perceptron (MLP) ANN will be in the interval [0, 1] or [-1, 1] depending on the transfer function used in the neurons. Reports in literature (e.g. Otache, [22]; Maier and Dandy, [25]; Wang, [19]) suggest that using smaller intervals for streamflow modelling, as [0.1, 0.85], and [0.1, 0.9] could allow extreme (*low and high*) flow events occurring outside the

range of the calibration data may be accommodated. However, the advantage of rescaling the data into a small interval is supported to a varying degree; fairly good in some instances and abysmally poor in the others. The prediction capability of extreme flow regimes by the different models is illustrated clearly in both **Table 1** and **Table 2**.

The behaviour as depicted by **Table 1** and **Table 2** could be explained against the backdrop of the behaviour of transfer functions. For instance, to rescale the input data to $[-1, 1]$ would limit the output range of the $\text{tansig}(x)$ function approximately to $[-0.7616, 0.7616]$. Similarly, to rescale the input range to $[-0.9, 0.9]$ would further shrink the output range approximately to $[-0.7163, 0.7163]$. Both 0.7616 and 0.7163 are still far away from the extreme limits of the $\text{tansig}(x)$ function; such a small output data range will make the output less sensitive to the change of the weights between the hidden layer and output layer, and will therefore possibly make the training process more difficult. As reported by Wang [19] and affirmed in Otache [22], since the neurons in an ANN structure are combined linearly with a lot of weights, any rescaling of the input vector can be offset the more, as corresponding weights and biases are changed.

4. Conclusion

Based on the results obtained in all instances, it could be inferred that adoption of large hidden layers could be counter-productive; this is because an excessive number of free parameters will encourage over fitting of the network though it may provide an additional representational power. In the same context, rescaling of the ANN input regime adversely limits the output of the transfer function. Thus resulting from the conclusions drawn, it suffices to note that ANN model is by no means a substitute for conceptual watershed modelling, therefore, exogenous variables should be incorporated in streamflow modelling and forecasting exercise because of their hydrologic evolutions and too, effort should be geared towards using hybrid models like Fuzzy-Neural Network and Wavelet models in a coupling strategy with ANN in the modelling of streamflow; similarly, because of volatility and nonlinear deterministic problems, ARMA-GARCH models should be considered as viable complement in this regard too.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Chibanga, R., Berlamont, J. and Vandewalle, J. (2003) Modelling and Forecasting of Hydrological Variables Using Artificial Neural Networks: The Kafue River Sub-Basin. *Hydrological Sciences*, **48**, 363-379. <https://doi.org/10.1623/hysj.48.3.363.45282>
- [2] Abrahart, J.R. and See, L. (2000) Comparing Neural Network and Autoregressive Moving Average Techniques for Provision of Continuous River Flow Forecasts in Two Contrasting Catchments. *Hydrological Processes*, **14**, 2157-2172.

- [https://doi.org/10.1002/1099-1085\(20000815/30\)14:11/12<2157::AID-HYP57>3.0.CO;2-S](https://doi.org/10.1002/1099-1085(20000815/30)14:11/12<2157::AID-HYP57>3.0.CO;2-S)
- [3] Abrahart, J.R. (1999) Neuro-Hydrology: Implication Options and a Research Agenda. *Area*, **31**, 141-149. <https://doi.org/10.1111/j.1475-4762.1999.tb00179.x>
- [4] Hillel, D. (1986) Modelling in Soil Physics: A critical Review. In: Boersma, L.L., Ed., *Future Developments in Soil Science Research*, Soil Science Society of America, Madison, WI, 35-42. <https://doi.org/10.2136/1987.futuredevelopmentssoil.c5>
- [5] Balkhair, S.K. (2002) Aquifer Parameters Determination for Large Diameter Wells using Neural Network Approach. *Journal of Hydrology*, **265**, 118-128. [https://doi.org/10.1016/S0022-1694\(02\)00103-8](https://doi.org/10.1016/S0022-1694(02)00103-8)
- [6] Hsu, K.L., Gupta, H.V. and Sorooshian, S. (1995) Artificial Neural Network Modelling of the Rainfall-Runoff Process. *Water Resources Research*, **31**, 2517-2530. <https://doi.org/10.1029/95WR01955>
- [7] Sarle, W.S. (1998) FAQ document for Usenet Newsgroup, comp.ai. neural-nets. <ftp://ftp.sas.com/pub/neural/FAQ.html>.
- [8] Fahlman, S.E. and Lebiere, C. (1990) The Cascade-Correlation Learning Architecture. In: Touretzky, D., Ed., *Advances in Neural Information Processing Systems*, Morgan Kaufmann Publishers, USA.
- [9] SNNS Group (1990-98) Stuttgart Neural Network Simulator—User Manual—Version 4.1. <http://www.ra.informatik.uni-tuebingende/SNNS/>
- [10] Takens, F. (1981) Detecting Strange Attractors in Turbulence. In: Rand, D.A. and Young, L.S., Eds., *Lecture Notes in Mathematics*, Springer-Verlag, New York, 366-381. <https://doi.org/10.1007/BFb0091924>
- [11] Packard, N.H., Crutchfield, J.P., Farmer, J.D. and Shaw, R.S. (1980) Geometry from a Time Series. *Physical Review Letters*, **45**, 712-716. <https://doi.org/10.1103/PhysRevLett.45.712>
- [12] Fraser, A. and Swinney, H.L. (1986) Independent Coordinates for Strange Attractors from Mutual Information. *Physical Review A*, **33**, 1134-1140, <https://doi.org/10.1103/PhysRevA.33.1134>
- [13] Mpitsos, G.J., Creech, H.C., Cohan, C.S. and Mendelson, M. (1987) Variability and Chaos: Neuron-Integrative Principles in Self-Organization of Motor Patterns. In: Hao, B.L., Ed., *Directions in Chaos*, World Scientific, Singapore, 162-190.
- [14] Porporato, A. and Ridolfi, L. (1997) Nonlinear Analysis of River Flow Time Sequences. *Water Resources Research*, **33**, 1353-1367. <https://doi.org/10.1029/96WR03535>
- [15] Jayawardena, A.W. and Lai, F. (1994) Analysis and Prediction of Chaos in Rainfall and Stream Flow Time Series. *Journal of Hydrology*, **153**, 23-52. [https://doi.org/10.1016/0022-1694\(94\)90185-6](https://doi.org/10.1016/0022-1694(94)90185-6)
- [16] Islam, M.N. and Sivakumar, B. (2002) Characterisation and Prediction of Runoff Dynamics: A Nonlinear Dynamical View. *Advances in Water Resources*, **25**, 179-190. [https://doi.org/10.1016/S0309-1708\(01\)00053-7](https://doi.org/10.1016/S0309-1708(01)00053-7)
- [17] Elshorbagy, A., Panu, U.S. and Simonovic, S.P. (2001) Analysis of Cross-Correlated Chaotic Streamflows. *Hydrological Sciences Journal*, **46**, 781-793. <https://doi.org/10.1080/02626660109492870>
- [18] Wilcox, B.P., Rawls, W.J., Brakensiek, D.L. and Wight, J.R. (1990) Predicting Runoff from Rangeland Catchments: A Comparison of Two Models. *Water Resources Research*, **26**, 2401-2410. <https://doi.org/10.1029/WR026i010p02401>
- [19] Wang, W. (2006) Stochasticity, Nonlinearity and Forecasting of Stream Flow Processes.

- Deft University Press, Amsterdam, Netherlands, 1-17.
- [20] Pasternack, G.B. (1999) Does the River Run Wild? Assessing Chaos in Hydrological Systems. *Advances in Water Resources*, **23**, 253-260.
[https://doi.org/10.1016/S0309-1708\(99\)00008-1](https://doi.org/10.1016/S0309-1708(99)00008-1)
- [21] Kennel, M.B., Brown, R. and Abarbanel, H.D. (1992) Determining Embedding Dimension for Phase-Space Reconstruction using Geometrical Construction. *Physical Review A*, **45**, 3403-3411. <https://doi.org/10.1103/PhysRevA.45.3403>
- [22] Otache, Y.M. (2008) Contemporary Analysis of River Benue Flow Dynamics and Modelling. Ph.D. Dissertation, Hohai University, Nanjing, China, 30-35, 60-79, 83-90.
- [23] Hornik, K., Stinchcombe, M. and White, H. (1989) Multi-Layer Feed forward Networks Are Universal Approximator. *Neural Networks*, **2**, 359-366.
[https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- [24] Ranjithan, S. and Eheart, J.W. (1993) Neural Network-Based Screening for Groundwater Reclamation under Certainty. *Water Resources Research*, **29**, 563-574.
<https://doi.org/10.1029/92WR02129>
- [25] Maier, H.R. and Dandy, G.C. (2000) Neural Networks for the Prediction and Forecasting of Water Resources Variables: A Review of Modelling Issues and Applications. *Environmental Modelling & Software*, **15**, 101-124.
[https://doi.org/10.1016/S1364-8152\(99\)00007-9](https://doi.org/10.1016/S1364-8152(99)00007-9)