Implications of Optimisation Algorithm on the Forecast Performance of Artificial Neural Network (Ann) for Streamflow Modelling

Martins Y. Otache,[1] John J. Musa,[1] Abayomi I. Kuti,[1] and Mustapha Mohammed [2]

***Abstract*—** The performance of Artificial Neural Network (ANN) is contingent on a host of factors; for instance, the network optimisation scheme. In view of this, the study examined the general implications of ANN training optimisation algorithm on its forecast performance. To this end, the Bayesian regularisation (Br), Levenberg-Marquardt (LM), and the adaptive learning Gradient descent: GDM (with momentum) algorithms were employed under different ANN structural configurations: (1) Single-hidden layer, and (2) Double-hidden layer feedforward back propagation network. Results obtained revealed generally that the Gradient Descent with momentum (GDM) optimisation algorithm, with its adaptive learning capability, used a relatively shorter time in both training and validation phases as compared to the Levenberg- Marquardt (LM) and Bayesian Regularisation (Br) algorithms though learning may not be consummate; i.e., in all instances considering also prediction of extreme flow conditions for 1-day and 5-day ahead, respectively especially using the ANN model. In specific statistical terms on the average, model performance efficiency using coefficient of efficiency (CE) statistic were Br: 98%, 94%; LM: 98 %, 95 %, and GDM: 96 %, 96% respectively for training and validation phases. However, on the basis of relative error distribution statistics (Mean Absolute Error: **MAE**, Mean Absolute Percentage Error: **MAPE**, and Mean Squared Relative Error: **MSRE**), GDM performed better than the others in the overall. Based on the findings, it is imperative to state that the adoption of ANN for real-time forecasting should employ training algorithms that do not have computational overhead like the case of LM that requires the computation of the Hessian matrix, protracted time, and sensitivity to initial conditions; to this end, Br and other forms of the gradient descent with momentum should be adopted considering overall time expenditure and quality of the forecast as well as mitigation of network overfitting. On the whole, it is recommended that evaluation should consider implications of (i) data quality and quantity and (ii) transfer functions on the overall network forecast performance.

***Keywords*—** Streamflow, neural network, optimisation, algorithm.

I. INTRODUCTION

THE Artificial Neural Networks (ANNs) uses difference or differential equation to capture the relationship between inputs and outputs without detailed attention to the internal structure of the physical processes**[1]**; being generally a nonlinear model, it can be mathematically treated as a universal approximator**[2]**, in other words, a black box model. Basically, ANNs mimic the functioning of the human brain by acquiring knowledge through a learning process that involves finding an optimal set of weights for the connections and threshold values for the nodes. The ANN models have been considerably adopted in flow forecasting; this is due to the fact that implementation and calibration of conceptual models is difficult and requires sophisticated mathematical tools **[3]**. The ANN model structure possesses the capability of modelling complex relations; especially, it stands out in the modelling of nonlinear dynamics **[3]**. Though, physical models, based on continuum mechanics offer one possible forecasting method, simpler approaches offered through black box solutions are fast becoming attractive alternatives. System-theoretic / black box or neural network forecasting and prediction offers various benefits ahead of the traditional conceptual modelling **[4]**. The ANN technology is an alternate computational approach based on theories of the massive inter-connection and parallel processing architecture of biological systems.

However, though an artificial neural network (ANN) has a robust mathematical structure that enhances its modelling capability, its overall performance is dependent on a whole lot of associated variables; for instance, neural network type, network structure, methods of pre- and post-processing of input and output data, training algorithm and training stop criteria **[2]**. 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. On the other hand, it suffices to note that as in the identification of other nonlinear types of models, the calibration (training) procedure is fraught with problems **[5]**. The ANN objective surface is typically highly non-convex with extensive regions; these regions are insensitive to the variations in the values of the network weights (parameters) and do contain numerous multi-local optima of generally high dimension **[6]**. Therefore, the success of the calibration (training) procedure depends largely on the power of the optimisation method used to search for the best parameter estimates **[5]**. Most procedures employed to train ANNs are the back propagation algorithm (BPA) and the step size-adaptive BPA in a Multi-layer perceptron network (MLP); but because these procedures employ a gradient search strategy, their performance is sensitive to the initial starting point **[5]**; in addition, they are easily trapped by local optima and are often ineffective when searching parameter spaces of high dimension.

Against the backdrop of the foregoing, considering that the ANN model structure is ideally suited for modelling highly nonlinear input-output relationships, the emphasis in this study is on the optimisation algorithm, and vis-a-vis intrinsically on

Martins Yusuf Otache is with the Federal University of Technology, Minna, Nigeria (e-mail: drotachemartyns@gmail.com).

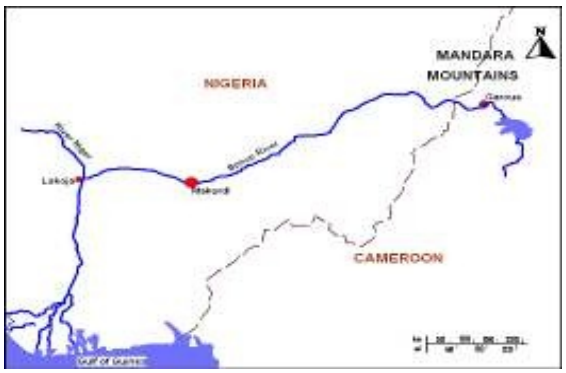
the associated architecture; i.e. how the combination of these two may inadvertently or otherwise affect the overall ANN model forecast performance; especially, knowing that daily streamflow processes exhibit a high degree of nonlinearity **[2]**. In addition, it suffices to note too that in one of the early applications involving streamflows, **[7]** as reported in **[2]**, ANNs and autoregressive moving average models were employed to predict daily and hourly streamflows in the Pyung Chang River basin in Korea. This study in particular confirmed the robustness of ANN as a useful tool for forecasting streamflows. In similar manner, many studies (e.g., **[8]**; **[9]**; **[10]**; **[2]**) attest to the superiority of ANN models over or equality to the traditional statistical and/or conceptual techniques in modelling the hydrological process.

II. METHODOLOGY

A. Data Collection and Management

For this study, daily streamflow sequence of the River Benue at the Makurdi (**Figure 1**) gauging station was obtained from the Benue State Water works and National Inland Waterways Authority (NIWA), Makurdi, Nigeria. **Figure 1** basically shows the traverse of the River and its flow regime for Makurdi section. The data sequence spanned through an entire period of thirty (30) years. To be able to effectively use the data obtained, quality analysis was carried out; to this end, in particular, 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.

(a)



(b)

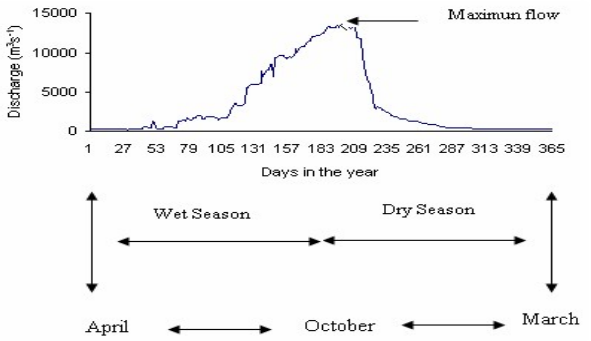


Figure 1: Basic details of River Benue: (a) Map of Nigeria showing River Benue and its traverse, (b) Flow regime of River Benue at Makurdi

B. Development of ANN Model

i. Network Topology

Since network architecture mainly denotes the number of input/output variables, the number of hidden layers and the number of neurons in each hidden layer, and considering that a univariate data type is used for this study, the time delay coordinate method **[12]**; **[13]** is used to reconstruct the phase-space from the scalar time 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 unfold the multi-dimensional structure using the available data. To do this, the optimum lag time or delay time, τ (tau) is set to 78 based on the analysis of the autocorrelation function of the daily streamflow series (i.e., the point where the autocorrelation function plot first crosses the zero line). Detailed implementation of this strategy was effected by applying the method for the determination of minimal sufficient dimension (m), in other words, called the 'false nearest neighbour' method' **[11]**. That is, supposing the point $X_i = \left[Y_{i-p+1}, \cdots, Y_i \right]$ has a neighbour $X_j = \left[Y_{j-p+1}, \cdots, Y_j \right]$ in a p-dimensional space then the distance $\left\| X_i - X_j \right\|$ is calculated in order to compute:

$$R_i = \frac{\left| Y_{i+1} - Y_{j+1} \right|}{\left\| X_i - X_j \right\|} \tag{1}$$

If R_i exceeds a given threshold R_T (a suitable value is $10 \le R_T \le 50$), 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 neighbour is actually zero, or sufficiently small, say, smaller than a criterion R_f. In this case, the false neighbour threshold R_T is set to 10. Based on this, the fraction of false nearest

neighbour as a function of the embedding dimension is calculated on phase-space reconstruction using embedding dimension. Here, the minimal embedding dimension is taken as 8; this implies that the state of the streamflow process can be determined by eight lagged observed values (**Figure 2**). Following from the analysis, eight lagged values of input variables were used when fitting the ANN model to the series; specifically, based on the phase-space reconstruction, this implied the discharges $Q_{t-7}, Q_{t-6}, \cdots, Q_t$ 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 (**Figure 3**).

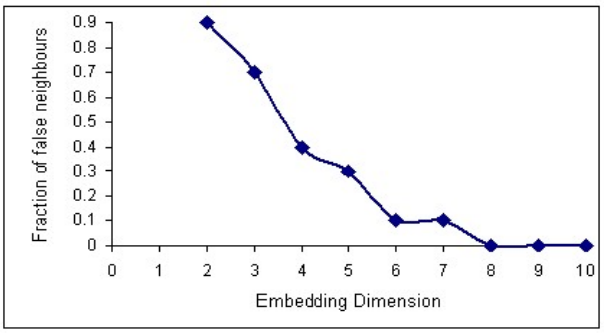


Figure 2: Fraction of false nearest neighbours as a function of embedding dimension

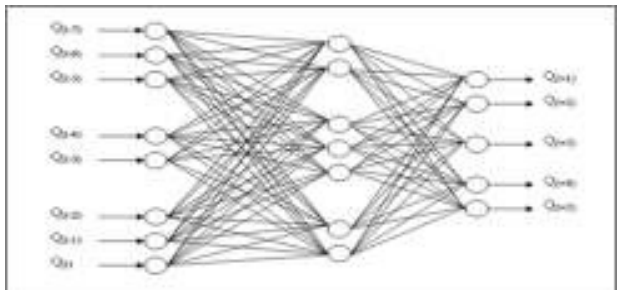


Figure 3: Schematic of three-layer feedforward artificial neural network architecture

To address the thrust of the study, two model types using the MLP were considered corresponding to two model architectures with different nodal configurations in all the instances. Precisely, single and double hidden layers were thus considered to examine the implications of model structural complexity; i.e., ANN models adopted were (i) 8 7 5 single-hidden layer with 7 nodes, 8 input nodes in the input layer while 5 represent five nodes in the output layer (ii) 8 5 2 5 double-hidden layers with 5 and 2 nodes, 8 nodes in the input layer, 5 nodes in the output layer, respectively.

ii. Network Training

The ANN training is a problem of nonlinear optimisation; it minimises the error between the network output and the target output according to a pre-determined algorithm. In this regard, error back-propagation (BP) is by far the most popular algorithm for optimising feedforward ANNs **[2]**, **[14]**, **[6]**; this basically is gradient descent technique that minimises the network error function **[2]**. For the purposes of the stated aim of the study, the multi-layer feedforward back propagation network was used by considering a number of optimisers for objectivity. Specifically, network training was implemented using the **trainbr** function (Bayesian regularisation), **traingdm** function (Gradient descent with momentum), **trainlm** function (Levenberg-Marquardt) in MATLAB Neural Network Toolbox. The choice of these optimisers is informed by the following facts: (i) Networks trained with the back-propagation algorithm are sensitive to initial conditions and may get stuck in local minima of error surface, (ii) **trainlm** unlike others, requires the computation of the Hessian matrix and thus, time involving **[6]**; (iii) on the other hand, the **traingdm** enhances stability of the network training whereas the Bayesian regularisation guides against network overfitting [**6**]; **[2]**.

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. On the other hand, as reported in **[14]**, not only training with the hyperbolic tangent function (tansigmoid) faster than training with the sigmoidal transfer function, but predictions obtained using networks with the hyperbolic tangent are slightly better than those with sigmoid transfer functions. Before applying the ANN, both input and output data were pre-processed and normalised.in the range [-1 1]. The scaling strategy was adopted; 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

$$x_t = \frac{\left(U_x - L_x\right)x_t^{'} + \left(M_x L_x - m_x U_x\right)}{M_x - m_x} \qquad (2)$$

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

$$y_t = \frac{\left(U_y - L_y\right) y_t^{'} + \left(M_y L_y - m_y U_y\right)}{M_y - m_y} \quad (3)$$

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, one needs to rescale the output $\hat{y}_t$ back to its original domain; this is by inverting Equation (3) and using $\hat{y}_t^{'}$ as

$$\hat{y}_t^{'} = \frac{\left(M_y - m_y\right) \hat{y}_t - \left(M_y L_y - m_y U_y\right)}{U_y - L_y} \quad (4)$$

C. Forecast analysis

In order to draw conclusions on the ANN model performance, parameters for statistical analyses such as **RMSE** (Root Mean Squared Error), **MSRE** (Mean Squared Relative Error), **MAE** (Mean Absolute Error), **CE** (Coefficient of Efficiency), and r^2 (Coefficient of Determination): (**See Appendix I**) as well as duration of training period, were employed to evaluate the ANN model predictions. The choice of these statistical performance indices is based on the following facts: (i) the MSE or equivalently RMSE, provide a good measure of the goodness-of-fit at high flows **[2]**, whereas (ii) MSRE provide a more balanced perspective of the goodness-of-fit at moderate flows **[2]**; on the other hand, (iii) CE and r^2 give useful comparisons between studies since they are independent of scale of data used **[2]**. In addition, r^2 measures the variability of observed flow that is explained by the model while CE provides a measure of the ability of a model to predict flows that are different from the mean **[6]**, **[2]**.

III. DISCUSSION

Table 1 shows the performance of the ANN model considering some basic attributes. It is obvious from the Table that the mean squared error (MSE) in all instances using the different optimisation algorithms (Bayesian regularisation: Br, Levenberg- Marquardt: LM, and Gradient descent with momentum: GDM) does not indicate any staggering differences as well as the maximum epoch for both training and validation phases except for Br. It is instructive to note that considering the relative time expended in the overall, two notable features are discernible; i.e., inadequate learning and over fitting. The GDM with its adaptive learning capability only used a matter of seconds as compared to both LM and Br, respectively and thus much appealing but learning may not be consummate or smooth. This gives room for circumspect in terms of the overall performance of the Network especially as noted in the reports of **[2]** and reaffirmed by **[6]**, the LM has computational overhead. The implication is that more time is used during computation and may unnecessarily lead to over fitting though the final error or forecast uncertainty might be minimal unlike the case with Br that is designed purely to take care of over fitting problems.

The forecast results are as reported in **Tables 2-5** for both the training and validation periods, respectively. Though the overall accuracy of the model in terms of the statistical parameters: (i) Coefficient of efficiency (CE), (ii) Coefficient of determination (r^2), and (iii) Root mean squared error (RMSE) (Tables 2, 3 and 4) are seemingly good, they do not really reveal the distribution of the forecast errors since there are global statistics. The values of MSRE and MAE in the validation period increase appreciably with the lead time (in days) indicating the distortion in the distribution of the forecast errors. This aspect in the forecast behaviour during the validation period is paramount since from a practical stand point they serve to assess and quantify the forecast errors of the ANN forecast model. **Tables 2-4** succinctly illustrate this distortion with regards to forecast of the entire flow regime. Generally, they provide an intuitive outlook of ANN model prediction when a univariate time series is used. In the same context, the increasing trend in the values of the CE and r^2 in all instances of the validation phase with corresponding decrease in the relative error statistics calls for concern and perhaps cautious optimism. For instance, it suffices to note that the concern should be whether the model is better than seasonal mean values of the series rather than overall observed mean; this finding accords with the results as in **[2]**. The only plausible explanation for this trend is that as the lead time increases, the network input weights begin to stabilise and learning becomes coherent since there is regularity in the data elements due to the lagging process similar to phase-space embedding.

Table 1: Training and Validation attributes in terms of training algorithm

Optimisation algorithm: Bayesian regularisation (Br)		
Attributes	Training	Validation
MSE	0.0031	0.0028
Time (Mins)	0:17:15	0:02:21
Iteration	2000	271
Effective number of parameters	101	101
Optimisation algorithm: Levenberg-Marquardt (LM)		
Attributes	Training	Validation
MSE	0.0031	0.0034
Time (Mins)	17:14	15:20
Iteration	2000	2000
Optimisation algorithm: Gradient descent with momentum (GDM)		
Attributes	Training	Validation
MSE	0.0077	0.0065
Time (Mins)	0:00:37	0:00:37
Iteration	2000	2000

There could be issues associated with data intermittency and redundancy in characteristically phase-spaced embedding; for instance, if the series is embedded some points that are actually far from each other will appear as neighbours because the geometric structure of the attractor has been projected down to a smaller space. Though Br has a high capacity to overcome generalisation problems due to over fitting unlike LM which is fraught with excessive computational overhead (e.g., computation of Hessian matrix) leading to instability problems, on the average, the GDM performed relatively better considering all the statistics.

It is clear from the snapshot of the performance statistics that one major problem in assessing ANN solutions is the use of global statistics. For one-day lead time predictions, the solution will in most cases produce a high or near perfect goodness-of-fit statistic. It suffices to note as seen here that the measures do not give an indication of what the network is getting right or wrong or where improvements could be made. Comparative scatter/correlation diagram (**Figure 4**) of the observed and forecasted streamflow for one-day-ahead and 5-day-ahead depicts the goodness-of-fit for the network trained using the ANN structure as determined in this case; i.e., in terms of effective overall correlation as noted in the values of the r^2 of the fitted trend line. A cursory look at **Figure 4** reveals that the same pattern is exhibited except for GDM in (c) as the r^2 for the fitted trend line decreased with increasing lead time in an expected fashion. It suffices to also note that the ANN model prediction of medium to high flows as noted here is not good enough. The findings suggest global measures are not good indicators of peak predictions due probably to the overwhelming presence of a large number of low flow situation.

In summary, as noted in **Tables 2-5** and **Figure 4,** the statistics clearly reveal the power of each of the respective optimisation algorithm. However, because most of the back propagation algorithms as noted here employ a gradient search strategy, their performance is sensitive to the initial starting point. It is imperative to also note that they are easily trapped by local optima and thus often ineffective when searching parameter spaces of high dimension.

Table 2: Overview of Model Performance Statistics using Bayesian regularisation (Br) optimization Algorithm

Optimisation algorithm: (LM)						
Event: Training	Performance Statistics					
Lead time	MAE	MAPE	RMSE	MSRE	CE	r^2
1	82.24	0.04	211.5	0.01	0.99	0.99
2	135.8	0.07	298.59	0.02	0.99	0.99
3	184.47	0.09	374.21	0.03	0.98	0.98
4	229.93	0.11	442.08	0.05	0.98	0.98
5	272.38	0.14	505.6	0.06	0.97	0.97
Average	180.96	0.09	366.4	0.03	0.98	0.98
Event: Validation	Performance Statistics					
Lead time	MAE	MAPE	RMSE	MSRE	CE	r^2
1	592.31	0.21	1227.1	0.13	0.92	0.92
2	553.11	0.2	1151	0.13	0.93	0.93
3	516.51	0.19	1080.4	0.12	0.94	0.94
4	484.07	0.18	988.14	0.11	0.95	0.95
5	454	0.17	911.16	0.1	0.96	0.96
Average	520	0.19	1071.6	0.12	0.94	0.94

Table 3: Snapshot of Model Performance Statistics based on Levenberg Marquardt (LM) optimisation algorithm

Optimisation algorithm: (LM)						
Event: Training	Performance Statistics					
Lead time	MAE	MAPE	RMSE	MSRE	CE	r^2
1	90.54	0.05	218.45	0.01	0.99	0.99
2	138.63	0.07	301.01	0.02	0.99	0.99
3	186.06	0.09	375.06	0.03	0.99	0.99
4	230.68	0.11	443.4	0.05	0.98	0.98
5	272.79	0.4	506.37	0.07	0.97	0.97
Average	183.74	0.09	368.86	0.04	0.98	0.98
Event: Validation	Performance Statistics					
Lead time	MAE	MAPE	RMSE	MSRE	CE	r^2
1	87.97	0.22	1154.3	0.14	0.93	0.93
2	549.04	0.21	1075.4	0.14	0.94	0.94
3	510.11	0.19	997.71	0.13	0.95	0.95
4	470.59	0.18	915.35	0.13	0.96	0.96
5	434.21	0.17	831.86	0.11	0.96	0.96
Average	510.3	0.19	994.92	0.13	0.95	0.95

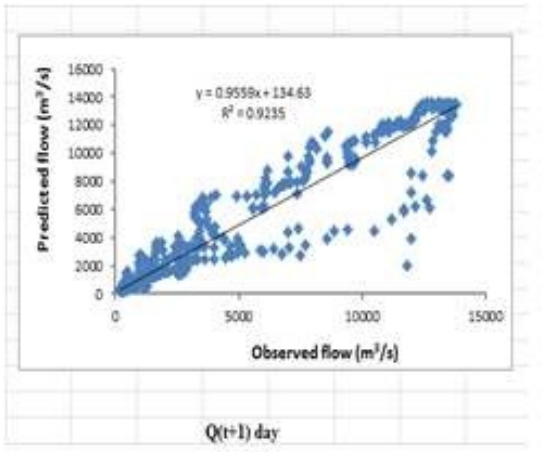
Table 4: Performance Statistics of the Model by employing the Gradient descent optimisation algorithm with momentum (GDM)

Event: Training Lead time	MAE	MAPE	RMSE	MSRE	CE	r^2
1	369.9	0.18	715.3	0.11	0.95	0.95
2	258	0.13	491.5	0.04	0.97	0.97
3	282.7	0.21	488.3	0.08	0.97	0.97
4	366	0.29	617	0.14	0.96	0.96
5	389.2	0.25	653.6	0.11	0.95	0.95
Average	333.1	0.21	593.1	0.1	0.96	0.96
Event: Validation						
Lead time	**MAE**	**MAPE**	**RMSE**	**MSRE**	**CE**	r^2
1	318.8	0.18	631.5	0.08	0.98	0.98
2	510.6	0.13	1062	0.03	0.94	0.96
3	405.6	0.19	807.1	0.08	0.97	0.97
4	495.4	0.21	956.5	0.12	0.95	0.95
5	534.5	0.17	1008	0.07	0.95	0.96
Average	453	0.18	892.9	0.08	0.96	0.96

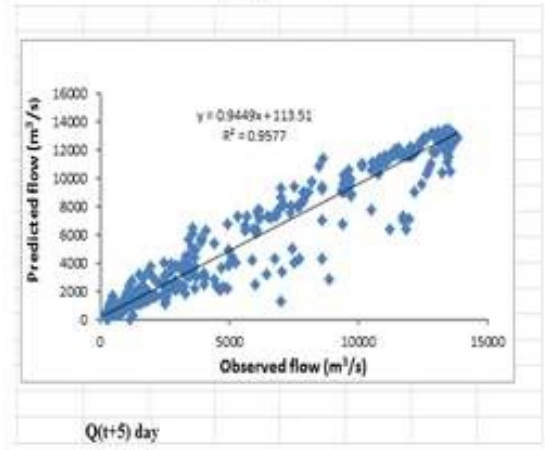
Table 5: Event-specific evaluation

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	
	1day	5day	1day	5day	1day	5day	1day	5day
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	
	1day	5day	1day	5day	1day	5day	1day	5day
Br	0.81	0.73	0.97	0.89	0.65	0.7	0.88	0.87
LM	0.81	0.73	0.98	0.89	0.65	0.69	0.88	0.86
GDM	0.67	0.7	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

(a)

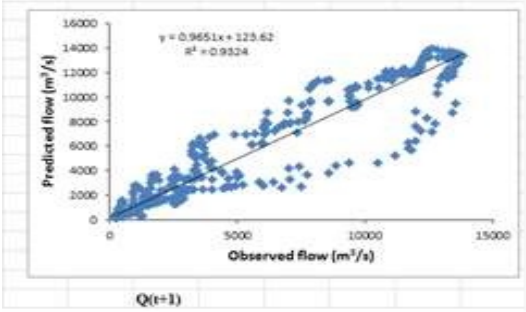


Q(t+1) day

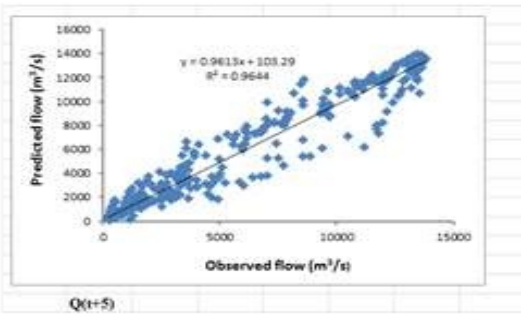


Q(t+5) day

(b)



Q(t+1)



Q(t+5)

(c)

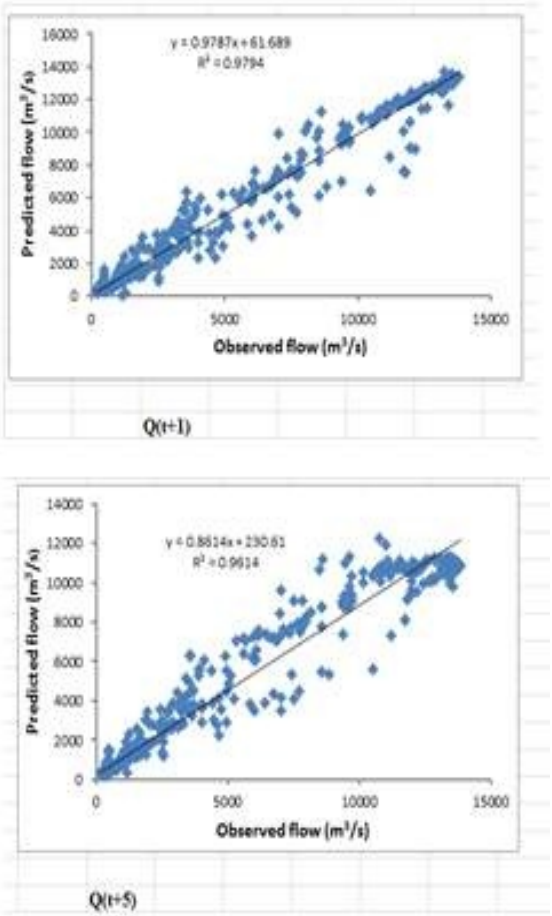


Figure 4: Correlation/Scatter plots for ANN model (8 (***Input nodes***), 7 (***nodes in the Hidden layer***), 5 (***Output nodes***)) in terms of one and five-day ahead prediction using (a) Br, (b) LM, and (c) GDM training optimisation algorithms, respectively.

IV. CONCLUSIONS

Based on the results obtained in all instances, it could be concluded that the success of an ANN training procedure depends largely on the power of the optimisation method employed to search for the best parameter estimates and not specifically on the complexity of the neural network structure.

Thus resulting from the general conclusions drawn, the following recommendations are proffered, namely: -

1. Effort should be geared towards using adaptive Neural Networks. Similarly, because of volatility and nonlinear deterministic problems, ARMA-GARCH models should be considered as viable complement
2. To allow for an enhanced exploration of the parameter space in terms of optimising time expenditure and robustness of the forecast, radial-basis network, hybrid models like Wavelet-ANN, FuNN (Fuzzy Neural Network) as well as Particle Swarm Optimisation coupled ANN should be considered as good options.
3. Despite the popularity of the CE, for hydrological time series which usually have strong seasonality, Seasonally Adjusted Coefficient of Efficiency: **SACE** should be considered.

REFERENCES

[1] J. R. Abrahart., and L. See (2000). Comparing Neural Network and Autoregressive Moving Average techniques for provision of continuous river flow forecasts in two contrasting catchments. J. Hydro. Processes, Vol. 14, pp: 2157-2172.

[2] W. Wang (2006). Stochasticity, nonlinearity and forecasting of streamflow processes. Deft University Press, Amsterdam – Netherlands, ISBN 1-58603-621-1; pp: 1-17, 2006.

[3] R. Chibanga., J. Berlamont., and J. Vandewalle (2003). Modelling and forecasting of hydrological variables using artificial neural networks: The Kafue River sub-basin. Hydrological Sciences, Vol. 48(3), PP: 363-364.

[4] J. R. Abrahart (1999). Neuro-hydrology: implication options and a research agenda. Area, Vol. 31(2); PP: 141-149.

[5] K. L. Hsu., H. V. Gupta., and S. Sorooshian (1995). Artificial neural network modelling of the rainfall-runoff process, Water Resour. Res., Vol. 31, pp: 2517-2530

[6] M. Y. Otache (2008). Contemporary Analysis of River Benue Flow Dynamics and Modelling. PhD dissertation, Hohai University, Nanjing, P. R. C. China; PP: 30-35, 60-79, 83-90.

[7] K. W. Kang., J. H. Kim., C. Y. Park., and K. J. Ham, K. (1993). Evaluation of hydrological forecasting system based on neural network model. In: Proc. 25th Congress of Int. Assoc. for Hydr. Res. IAHR, Deft, The Netherlands, pp: 257-264.

[8] A. S. Tokar., and M. Markus (2000). Precipitation-runoff modelling using artificial neural network and conceptual models. J. Hydrological Process, 14, pp: 1362-1376.

[9] S. Birikundavyi., R. Labib., H. T. Trung., and J. Rouselle (2002). Performance of neural networks in daily streamflow forecasting. J. Hydrol. Eng. 7(5), pp: 392-398.

[10] Y. B. Dibike., and D. P. Solomatine (2001). River Flow Forecasting Using Artificial Neural Networks, Journal of Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere, 26(1), pp: 1-8.

[11] M. B. Kennel., R. Brown., and H. D. Abarbanel (1992). Determining embedding dimension for Phase-space reconstruction using geometrical construction. Phys. Rev. A. Vol. 45, pp: 3403-3411.

[12] N. H. Packard., J. P. Crutchfield., J. P. Farmer., and R. S. Shaw (1980). Geometry from a time series, Phys. Rev. Lett., Vol. 45, pp: 712-716.

[13] F. Takens (1981). Detecting strange attractors in turbulence in Lecture Notes in Mathematics, edited by D.A. Rand and L.S. Young, Vol. 898, pp: 366-381, Springer-Verlag, New York.

[**14]** H. R. Maier., and G. C. Dandy (1998). The effect of internal parameters and geometry on the performance of back-propagation neural networks: An empirical study. Environmental modelling Software, 13(2), pp: 193-209

APPENDICES

Appendix I: Statistical Indices

Mean Absolute Error: $MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Q_i - \hat{Q}_i\right|$

Mean Absolute Percentage Error:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Q_i - \hat{Q}_i\right|}{Q_i}$$

Root Mean Squared Error:

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

Mean Squared Relative Error:

$$MSRE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(Q_i - \hat{Q}_i\right)^2}{\hat{Q}_i}$$

Coefficient of Efficiency:

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

Coefficient of Determination:

$$r^2 = \left[\frac{\sum_{i=1}^{n}\left(Q_i - \bar{Q}\right)\left(\hat{Q}_i - \tilde{Q}\right)}{\sqrt{\sum_{i=1}^{n}\left(Q_i - \bar{Q}\right)^2\sum_{i=1}^{n}\left(\hat{Q}_i - \tilde{Q}\right)^2}}\right]^2$$