OPEN ACCESS

Saudi Journal of Civil Engineering
Abbreviated Key Title: Saudi J Civ Eng
ISSN 2523-2657 (Print) |ISSN 2523-2231 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com/journal/sjce/home

Original Research Article

Rainfall-River Discharge Modelling Using Artificial Neural Network – A Case Study of Oramiriukwa River in Owerri, Imo State Nigeria

Alerechi, Kelechi [1]*, Dike B. U.[1], Nwoke H. U.[1]

[1]Department of Civil Engineering, Federal University of Technology Owerri, Imo State Nigeria

DOI: https://doi.org/10.36348/sjce.2025.v09i05.002 | **Received:** 10.04.2025 | **Accepted:** 16.05.2025 | **Published:** 19.05.2025

***Corresponding author:** Alerechi, Kelechi
Department of Civil Engineering, Federal University of Technology Owerri, Imo State Nigeria

Abstract

This study investigates the application of Artificial Neural Networks (ANNs) for rainfall-river modelling in the Oramiriukwa River, in Owerri, Imo State, Nigeria. The study utilizes daily streamflow data from the Ulakwo station (1978–1988) alongside corresponding rainfall and temperature data for Imo State, obtained from the Nigerian Meteorological Agency (NiMet). A series of Feedforward Multi-Layer Perceptron (MLP) models was developed and tested using MATLAB, with performance evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2). Among the models, Model 4 ([15]) delivered the best results, achieving an R^2 of 0.9158, MSE of 0.1294, and RMSE of 0.3597, demonstrating its effectiveness for streamflow prediction in the Oramiriukwa River. Model 2 ([30, 15, 5]) also showed good performance ($R^2 = 0.9029$), but its increased complexity suggested a potential risk of overfitting. Model 1 ([20, 10]) yielded lower predictive accuracy, highlighting the need for more complex architectures or additional input features to improve ANN performance for hydrological applications. These results confirm the effectiveness of ANNs in modelling nonlinear hydrological processes and suggest their potential for improving streamflow prediction in similar river basins. This study contributes to the growing use of data-driven methods in hydrological modelling in Nigeria and offers a foundation for future work aimed at enhancing the accuracy and robustness of ANN models.

Keywords: Artificial Neural Network (ANN), Rainfall-river discharge modelling, Oramiriukwa River, Feedforward Multi-Layer Perceptron (MLP), MATLAB, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Coefficient of Determination (R^2).

I. INTRODUCTION

Developing reliable rainfall–river discharge models remains a central concern in hydrology, particularly for enhancing decision-making in water resources planning and management. These models must consider a wide range of location-specific factors, including rainfall intensity and distribution, soil infiltration capacity and porosity, initial soil moisture conditions, land use patterns, and evaporation rates. Given the complexity and interdependence of these variables, rainfall-runoff modelling approaches are generally categorized into three types: physical, conceptual, and empirical models. This classification, as outlined by Sitterson, Knightes, Parmar, Wolfe, Muche, and Avant (2017), reflects the varying degrees to which models rely on physically-based processes or observational data to simulate hydrological behaviour.

The choice of modelling approach typically depends on the extent of available data and the understanding of the hydrological processes at play. Physical and conceptual models require comprehensive information about surface water dynamics and often involve numerous physical parameters, making them more suitable in data-rich settings (Srinivasulu, 2008, as cited in Sitterson *et al.*, 2017). Empirical models, on the other hand, also referred to as data-driven models, are particularly useful when physical data are scarce. These models focus on identifying statistical relationships between historical inputs, such as rainfall, and outputs, such as river discharge (Sitterson *et al.*, 2017).

Among empirical methods, Artificial Neural Networks (ANNs) have emerged as powerful tools for modelling the nonlinear and complex nature of hydrological systems. Their flexibility and ability to learn from data make them highly effective in various applications, including rainfall-runoff modelling (Braddock, Kremmer, and Sanzogni, 1998; Tokar and Johnson, 1999), river flow prediction (Imrie, Durucan, and Korre, 2000), time series forecasting (Hu, Lam, and Ng, 2001), and flood forecasting (Wei, Xu, Fan, and Tasi, 2002). ANNs have also been applied in rainfall-runoff prediction (Dawson and Wilby, 2000; Raid, Mania, Bouchaou, and Najjar, 2004), evaporation estimation (Andersen and Jobson, 1982), and rainfall estimation (Lin and Wu, 2009). Their suitability for such tasks has been widely acknowledged by researchers, including the ASCE Task Committee (2000a), especially in scenarios where traditional physical modelling approaches are constrained by data limitations.

The foundation of ANNs lies in the biological neural networks of the human brain, which consist of intricately connected neurons capable of parallel information processing. This biological inspiration has led to the development of artificial systems that mimic the fundamental behaviour of neural communication. The Artificial Neural Network is therefore modelled to emulate certain features of the biological system in order to solve computational and predictive problems. Tanty and Desmukh (2015) defined ANN as “a mathematical structure capable of representing the arbitrary, complex and non-linear process correlating the input and output of any system.”

In hydrological contexts, ANNs are especially valuable for modelling phenomena such as rainfall-runoff transformation, which is inherently dynamic, non-linear, and influenced by spatial and temporal variability. Traditional models often fall short in capturing these complexities. However, ANNs, through their ability to model intricate patterns and relationships in data, have proven effective in generating reliable synthetic streamflow data using rainfall inputs and historical discharge records (Raid, Mania, Bouchaou, and Najjar, 2004).

II. METHODOLOGY

A. Study Area Overview

The study area is situated in Imo State, located in southeastern Nigeria, with Owerri as the capital and its most urbanized centre. The state spans roughly 5,529.17 square kilometres and lies within the geographical coordinates of latitude 5°33′N to 6°07′N and longitude 7°08′E to 7°61′E. The state takes its name from the Imo River, which flows from the Okigwe-Awka uplands and is positioned between the lower Niger River and the central reaches of the Imo River system (Amangabara, 2015).

Topographically, the region varies from flat lowlands in the south to undulating and hilly terrain in the northern parts, with elevations ranging from 350 to 790 meters above mean sea level (Obasi, Ogbu, Orakwe and Ahaneku, 2020). Imo State experiences a humid tropical climate with a clear distinction between the rainy season (April to October) and the dry season (November to March). The highest rainfall events generally occur in April and September. According to Adeleye (1974), the annual average temperatures in Owerri hover around 32°C, maintaining warm conditions year-round.

The hydrology of Owerri is shaped by several rivers, notably the Oramiriukwa River, which originates in the uplands of Okigwe and runs southward for about 5.8 kilometres before converging with the Otamiri and Nworie Rivers. These tributaries form part of the broader Imo River basin that ultimately discharges into the Atlantic Ocean (Amadi, Olasehinde, Okosun and Yisa, 2010). The Oramiriukwa River is flanked by forested swamp areas, predominantly covered with raffia palms (Adaka, Ndukwe and Nlewadim, 2015). A map of the study area is presented in Figure 1.

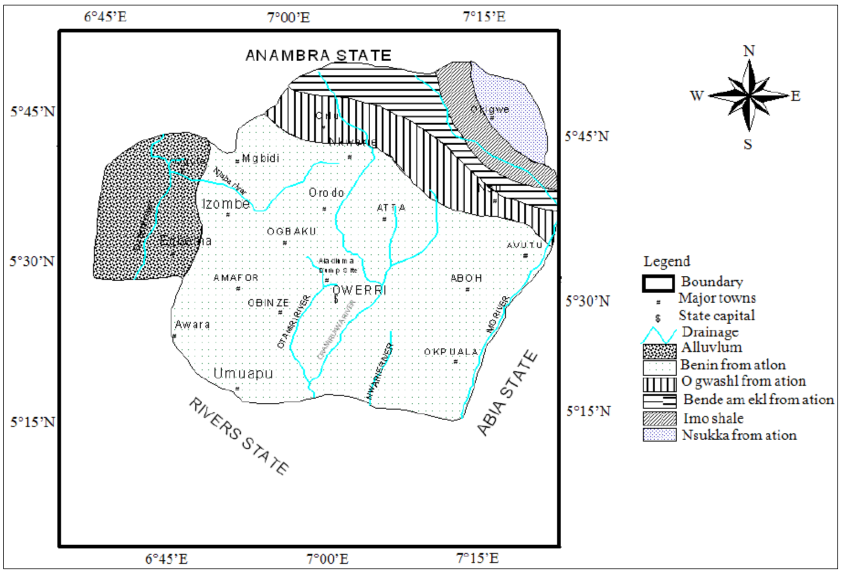


Figure 1: Map of Imo State showing the study area (Source: Amadi *et al.*, 2010)

B. Data Collection and Sources

This study utilized daily streamflow records for the Oramiriukwa River, sourced from the Ulakwo hydrological gauging station. The dataset spans a ten-year timeframe from 1978 to 1988. The Oramiriukwa River, which drains an estimated area of 795 square kilometres, is geographically located around latitude 5°25′N and longitude 7°07′E.

Complementary to the streamflow data, daily meteorological variables (specifically rainfall and temperature) for the same duration (1978–1988) were retrieved from the Nigerian Meteorological Agency (NiMet) for the Imo State region. These datasets constituted the primary inputs used in conducting the rainfall-river discharge modelling and hydrological evaluation presented in this research.

C. Data Processing

1. Data Cleaning

The first step in the preprocessing phase involved improving the dataset's quality to ensure accurate model development. Outliers were identified through scatter plots in Excel, followed by the application of Grubbs' Test (Grubbs, 1969). Any inconsistent values were either corrected or replaced to maintain the temporal integrity of the data. To address missing values in rainfall and streamflow, linear interpolation was employed, ensuring that the dataset remained continuous and suitable for training the model.

2. Lag Selection and Input Configuration

To enhance model prediction, cross-correlation analysis was employed to determine the optimal time lags between rainfall and streamflow (Sudheer & Jain, 2004; Dawson & Wilby, 1998). This technique assesses the similarity between two time series at various lags, selecting the lags that exhibit the highest correlation. The analysis was carried out using a custom MATLAB program.

The configuration of inputs and outputs was based on insights from previous studies (Behmanesh and Ayashm, 2015; Machado, Mine, Kaviski and Fill, 2011; Raid et al., 2004). This led to the development of five distinct dataset combinations, which included lagged variables such as rainfall, temperature, and streamflow. These combinations are summarized in Table 1. The resulting model structure can be represented by the following equation:

$$\mathbf{Q_t} = f(\mathbf{Q_{t\text{-}n}}, \mathbf{R_{t\text{-}n}}, \mathbf{X_{t\text{-}n}}) \quad (1)$$

where $\mathbf{Q_t}$ represents the current flow, $\mathbf{Q_{t-n}}$ is the antecedent flow at time steps t–1, t–2, … t–n, $\mathbf{R_{t-n}}$ denotes antecedent rainfall, and $\mathbf{X_{t-n}}$ encompasses other influencing factors like temperature (Kalteh, 2008; Furundzic, 1998).

3. Data Normalization

To standardize the input variables and minimize the bias introduced by varying units and scales, Z-score normalization was applied (Raid et al., 2004). The mean and standard deviation were calculated from the training dataset and used to normalize all three data subsets (training, validation, and testing). This process is described by the following equation:

$$\overline{x}_i = \frac{x_i - \mu_{(x_i)}}{\sigma_{(x_i)}} \quad (2)$$

where, $\boldsymbol{\overline{x}_i}$ represents the standardized variable, X_i is the original variable, $\mu_{(\boldsymbol{x_i})}$ is the mean of the variable, and $\boldsymbol{\sigma_{(x_i)}}$ is the standard deviation of the variable.

D. Artificial Neural Network (ANN)

1. ANN Structure

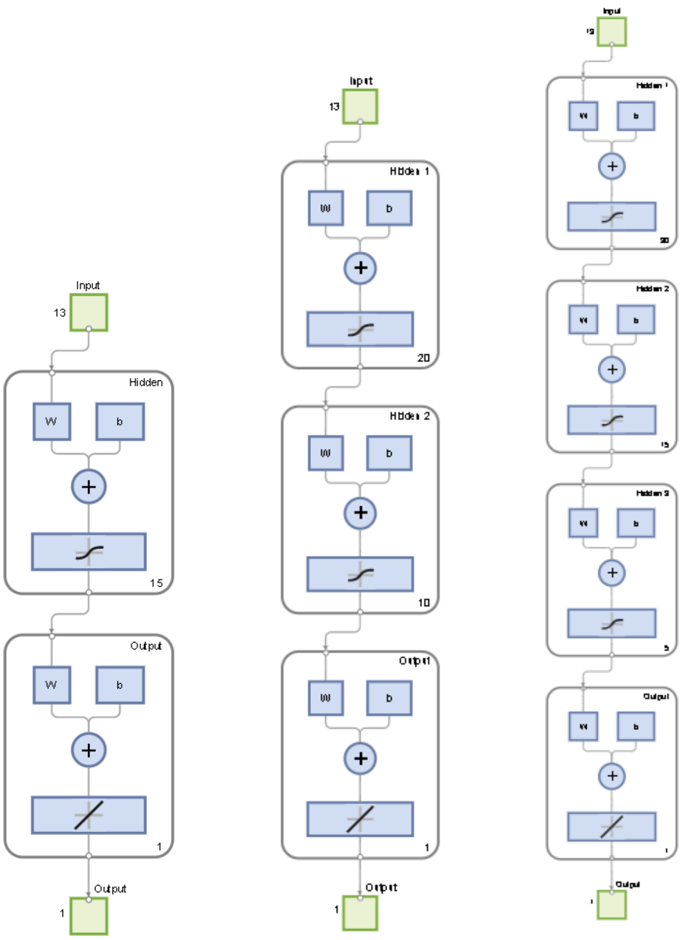
For this study, a Multi-Layer Perceptron (MLP) feedforward neural network was chosen due to its effectiveness in modelling nonlinear hydrological relationships (Gallant, 1993; Napolitano, 2011). During the modelling process, three different network architectures were tested iteratively:

- One hidden layer with 15 neurons
- Two hidden layers with 20 and 10 neurons
- Three hidden layers with 30, 15, and 5 neurons

The optimal architecture was selected based on the model's performance across these different configurations. A schematic representation of the implemented MLP network is shown in Figure 2. The general structure of a feedforward MLP with a single hidden layer is represented by the following ANN equation:

$$\mathrm{y_j} = \sum_{\mathrm{j=1}}^{\mathrm{q}} \mathrm{v_j}\, g\left(\sum_{\mathrm{i=1}}^{\mathrm{p}} \mathrm{w_{ij}x_i} + \mathrm{b_j}\right) \quad (3)$$

where $\mathbf{y_j}$ is the target output, **p** represents the number of input features, **q** is the number of nodes in the hidden layer, $\boldsymbol{g}$ is the activation function of the hidden layer nodes, $\mathbf{x_i}$ are the input features, $\mathbf{w_{ij}}$ are the weights between the input **i** and the hidden node **j**, $\mathbf{v_i}$ are the weights connecting the hidden layer to the output, and $\mathbf{b_j}$ is the bias for the hidden node **j**.



(a) (b) (c)

Figure 2: Structure of the ANN Architecture in MATLAB with (a) one hidden layer (b) two hidden layer (c) three hidden layer

2. Training Algorithm

The network training employed the Levenberg-Marquardt backpropagation algorithm, known for its efficiency and high convergence speed (Vos, 2003; Napolitano, 2011). This hybrid optimization technique combines features of the gradient descent and Newton's method, enabling it to minimize the mean squared error more effectively during network training. The algorithm computes the output of the network, evaluates the discrepancy from the observed values, and updates the weights and biases through a backpropagation process aimed at reducing the cumulative error.

The weight update rule is given by:

$$\mathrm{x}_{\mathrm{k+1}} = \mathrm{x}_{\mathrm{k}} - [J^T J + \mu I]^{-1} J^T e \qquad (4)$$

where $\mathbf{x_{k+1}}$ represents the weights at the (k+1) th epoch, $\boldsymbol{J}$ denotes the Jacobian matrix containing the first derivatives of network errors with respect to weights and biases, $\boldsymbol{\mu}$ is learning rate, and $\boldsymbol{e}$ is a vector of network errors.

To avoid overfitting and enhance model generalization, the following strategies were implemented:

i. **Early stopping**: Training was automatically halted if the validation error did not improve for 20 consecutive epochs, with the total training capped at 1000 epochs.

ii. **Activation function**: The hidden layers utilized the 'tansig' (hyperbolic tangent sigmoid) function, which is a bounded, differentiable, and symmetric activation function. It maps inputs to an output range between –1 and +1 and is defined as:

$$f(t) = \frac{2}{1+e^{-2x_i}} - 1 \qquad (5)$$

where, $\boldsymbol{f(t)}$ is the output of the activation function, and $\mathbf{x_i}$ is the input to the neuron.

3. Data Split

To facilitate model development and evaluation, the available dataset was partitioned into three subsets:

i. **Training set**: used to iteratively update the model's internal weights during learning.

ii. **Validation set:** used to monitor generalization capability and prevent overfitting by fine-tuning hyperparameters.

iii. **Testing set**: reserved for assessing the predictive performance of the trained model on previously unseen data.

The proportion of data assigned to each subset was determined empirically through trial-and-error to achieve optimal predictive accuracy. Details on the chosen ratio and its effect on model performance are presented in the 'Results' section.

4. Evaluation Metrics

The performance of the ANN model was quantitatively evaluated using three widely adopted statistical metrics:

i. **Mean Squared Error (MSE):** This measures the average squared difference between observed and predicted streamflow values. Lower MSE values indicate improved model accuracy.

$$\text{MSE} = \frac{\sum_{i=1}^{n}(Q_i - \widehat{Q_i})^2}{n} \qquad (6)$$

ii. **Rooted Mean Squared Error (RMSE):** This provides an interpretable measure of error magnitude in the same units as the streamflow data. Like MSE, lower RMSE signifies better performance.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(Q_i - \widehat{Q_i})^2}{n}} \qquad (7)$$

iii. **Coefficient of Efficiency (R^2):** This reflects how well the predicted values replicate observed streamflow variability. A value closer to 1 indicates a higher level of model explanatory power.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Q_i - \widehat{Q_i})^2}{\sum_{i=1}^{n}(Q_i - \overline{Q})^2} \qquad (8)$$

where, Q_i is observed discharge, $\widehat{Q_i}$ is predicted discharge, $\overline{Q}$ is mean observed discharge, and n is number of observations.

5. MATLAB Process

The process was executed using MATLAB, with the following steps:

i. **Data Loading**: Input and output datasets were imported from a CSV file.
ii. **Network Configuration**: The number of neurons and layers were defined using the 'feedforwardnet' function.
iii. **Parameter Initialization**: Initial weights and biases were assigned automatically.
iv. **Hyperparameter Setup**: Learning rate and early stopping conditions were configured.
v. **Training Execution**: The network was trained iteratively using input-output pairs.
vi. **Backpropagation**: Errors were propagated and used to update weights and biases.
vii. **Termination**: The process stopped when accuracy was achieved, or early stopping was triggered.

This sequence ensured that the ANN-based rainfall–river discharge model learned effectively while minimizing overfitting. This process repeats until the model reaches a desired accuracy or the early stopping criteria are met, ensuring it performs well on unseen data.

III. RESULTS

A. Overview of the Cleaned Data

The cleaned daily streamflow records for the Oramiriukwa River (1978–1988), obtained from the Ulakwo gauging station, are presented in Figure 3. Complementary daily rainfall and temperature datasets, acquired from the Nigerian Meteorological Agency (NiMet) for Imo State, are shown in Figures 4 and 5, respectively.

For improved readability and to highlight seasonal trends, the data displayed in these figures correspond to a representative hydrological year spanning April 1978 to March 1979. This one-year window enables focused visualization of variability in streamflow response to climatic inputs.

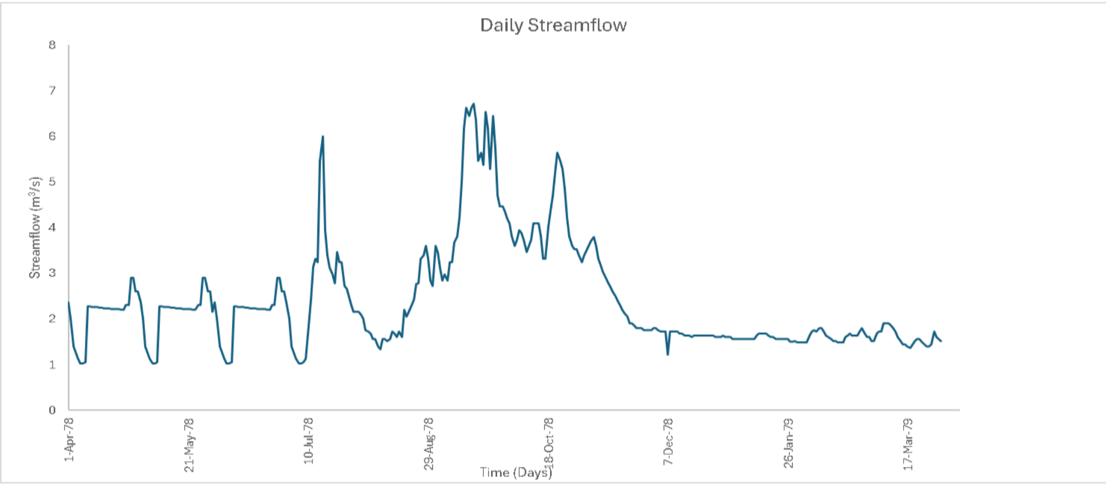


Figure 3: Time Series Plots of Daily Streamflow the Oramiriukwa River

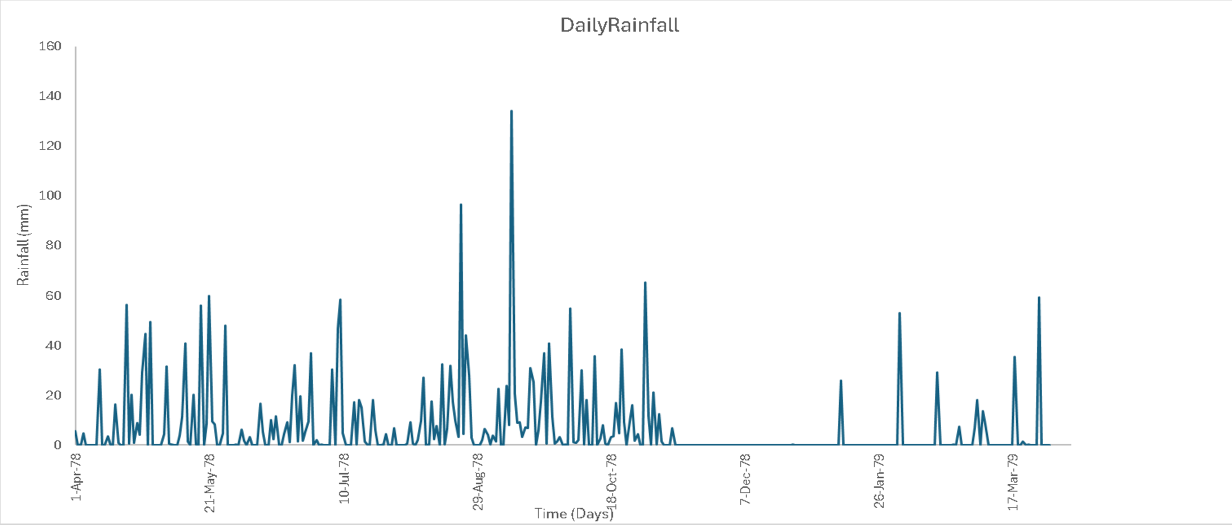


Figure 4: Time Series Plots of Daily Rainfall Data for Imo State

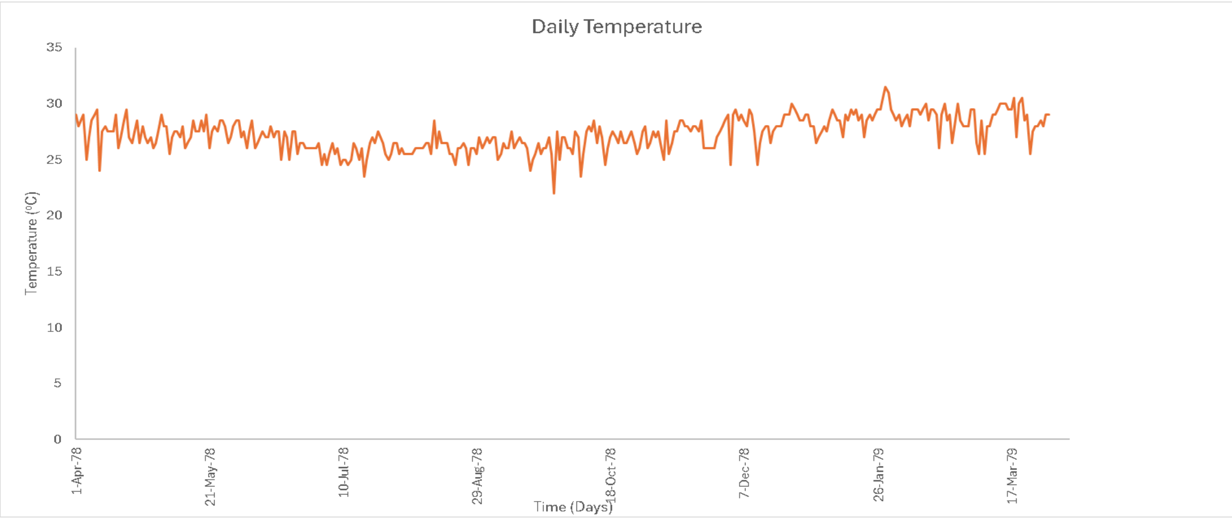


Figure 5: Time Series Plots of Daily Temperature Data for Imo State

B. ANN Model Results

The architecture and structure of the five (5) input scenarios used to develop the rainfall-river discharge ANN-based models are summarized in Table 1. Each model was constructed with varying combinations of rainfall, temperature, and antecedent discharge values to examine their predictive influence on streamflow. Three ANN architectures; single-layer ([15]), two-layer ([20, 10]), and three-layer ([30, 15, 5]) feedforward MLPs, were tested for each input configuration.

Table 1: ANN Architecture for Rainfall-River Discharge Model

ANN Model	Input	Output	Network Architecture
1.	R_t, T_t	Q_t	[15], [20, 10], [30, 15, 5]
2.	R_t, T_t, Q_{t-1}	Q_t	[15], [20, 10], [30, 15, 5]
3.	R_t, R_{t-1}, R_{t-1}, T_t, T_{t-1}, Q_{t-1}, Q_{t-2}	Q_t	[15], [20, 10], [30, 15, 5]
4.	R_t, R_{t-1}, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-5}, R_{t-5}, Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}	Q_t	[15], [20, 10], [30, 15, 5]
5.	R_t, R_{t-1}, R_{t-1}, R_{t-2}, R_{t-3}, R_{t-5}, R_{t-5}, T_t, T_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}	Q_t	[15], [20, 10], [30, 15, 5]

Note: subscripts, t, t-1, t-2… represent time steps (e.g., current day, one day ago, two days ago, etc.); R = Rainfall (mm); T = Temperature (°C); and Q = Discharge (m^3/s)

Model optimization was conducted in MATLAB through iterative tuning of network parameters, activation functions, and regularization settings. A data split ratio of 70% for training, 15% for validation, and 15% for testing was adopted as it provided the best generalization performance. The corresponding time periods for the data subsets were:

i. **Training set (70%):** April 1978 to March 1985.

ii. **Validation set (15%):** April 1985 to September 1986.
iii. **Testing set (15%):** October 1986 to March 1988.

This configuration ensured effective model learning while preserving a sufficient portion of the dataset for independent validation and final testing.

The performance metrics for each ANN model were evaluated, and the best-performing model for the Oramiriukwa River is presented in Table 2.

Table 2: Best Performing Feedforward-MLP ANN Model for Oramiriukwa River

ANN Model	Best Performing Network Architecture	Dataset	Statistical Parameters		
			MSE	RMSE	R^2
1	[20 10]	Training	0.8826	0.9395	0.1823
		Validation	1.9859	1.4092	0.1016
		Testing	1.2336	1.1107	0.1972
2	[30 15 5]	Training	0.2973	0.5453	0.7247
		Validation	0.3169	0.563	0.8566
		Testing	0.1493	0.3864	0.9029
3	[30 15 5]	Training	0.3113	0.5579	0.7117
		Validation	0.3544	0.5953	0.8402
		Testing	0.183	0.4278	0.8808
4	[15]	Training	0.2977	0.5456	0.7242
		Validation	0.3802	0.6166	0.8292
		Testing	0.1294	0.3597	0.9158
5	[15]	Training	0.289	0.5376	0.7323
		Validation	0.3333	0.5773	0.8503
		Testing	0.149	0.386	0.903

Figures 6 to 11 present scatter plots and time series plots, customized in the MATLAB program, for the best-performing models in Model 1, Model 2, and Model 4, as summarized in Table 2. These figures compare the daily predicted streamflow against the actual streamflow for the testing dataset, showcasing the model's performance. The plots visually illustrate the model's ability to replicate streamflow patterns, highlighting its predictive accuracy and identifying any discrepancies between predicted and actual streamflow. Additionally, Figures 12 to14 display sample snippets of the plots automatically generated by MATLAB after the program run, showcasing the model's results in a more condensed form.

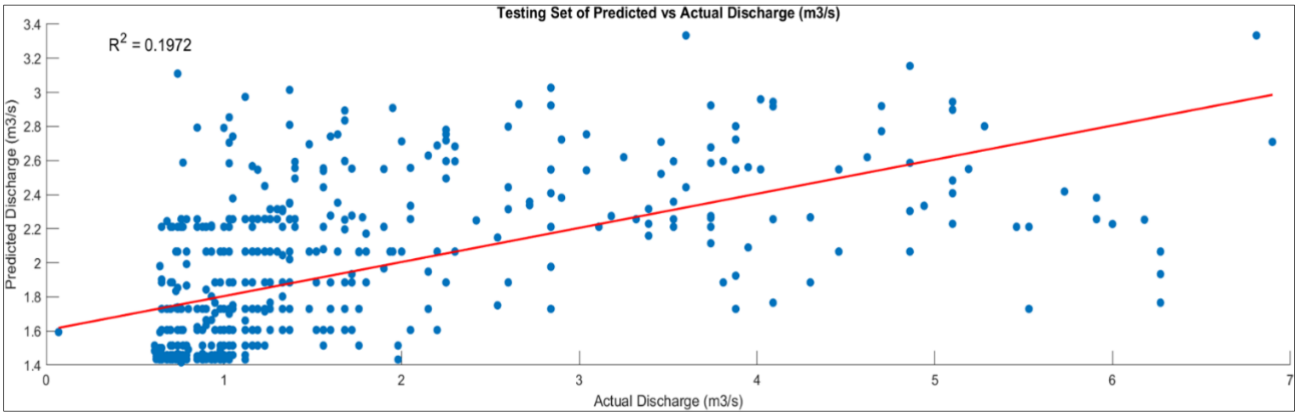


Figure 6: ANN Model 1 Prediction Performance of Best Fitted for Testing Dataset

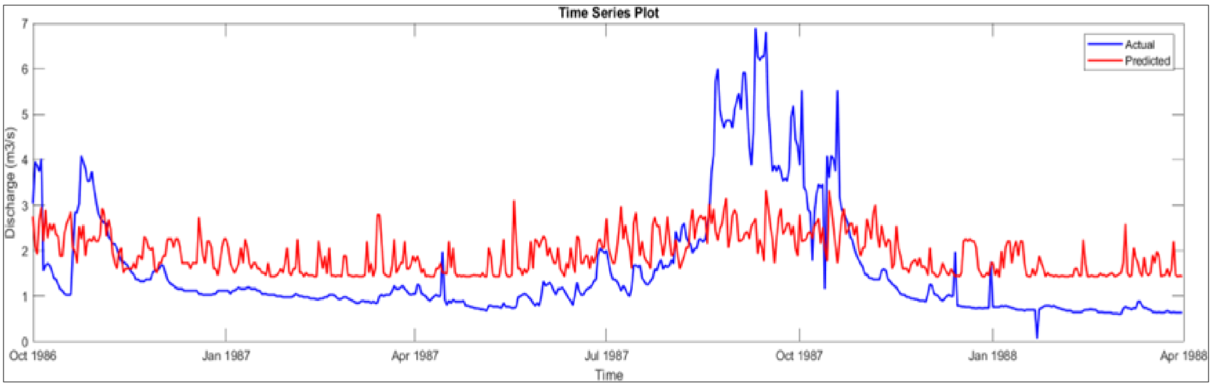


Figure 7: ANN Model 1 Time Series Plots of the Actual vs. Predicted Daily Streamflow for the Testing Dataset

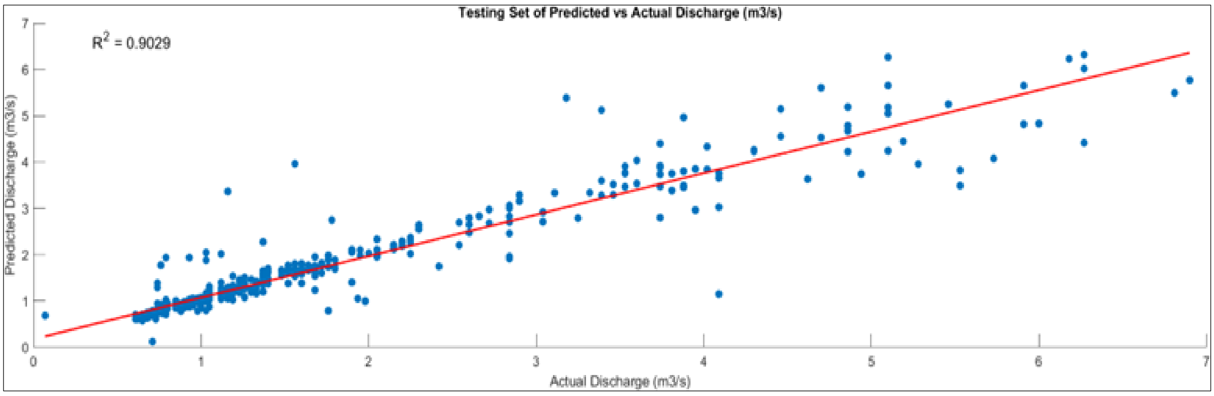


Figure 8: ANN Model 2 Prediction Performance of Best Fitted for Testing Dataset

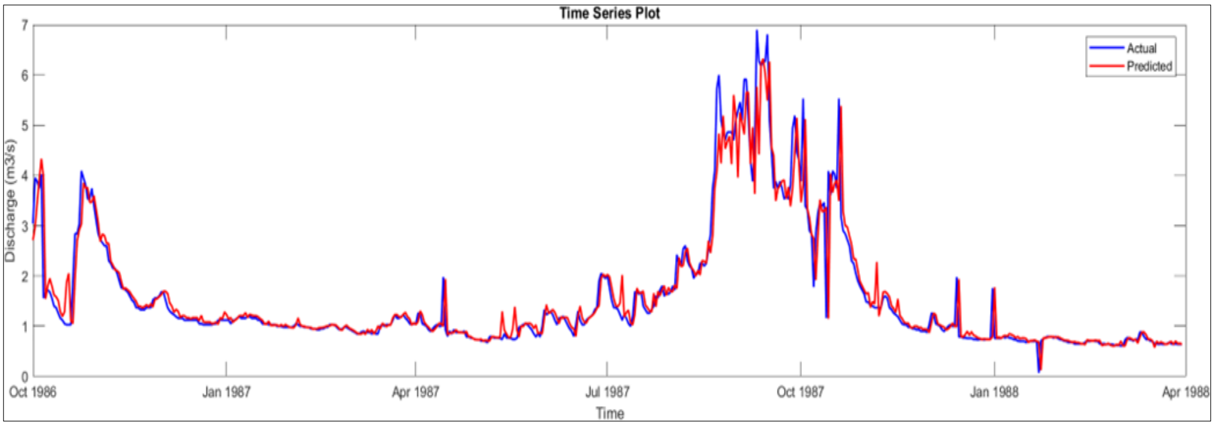


Figure 9: ANN Model 2 Time Series Plots of the Actual vs. Predicted Daily Streamflow for the Testing Dataset

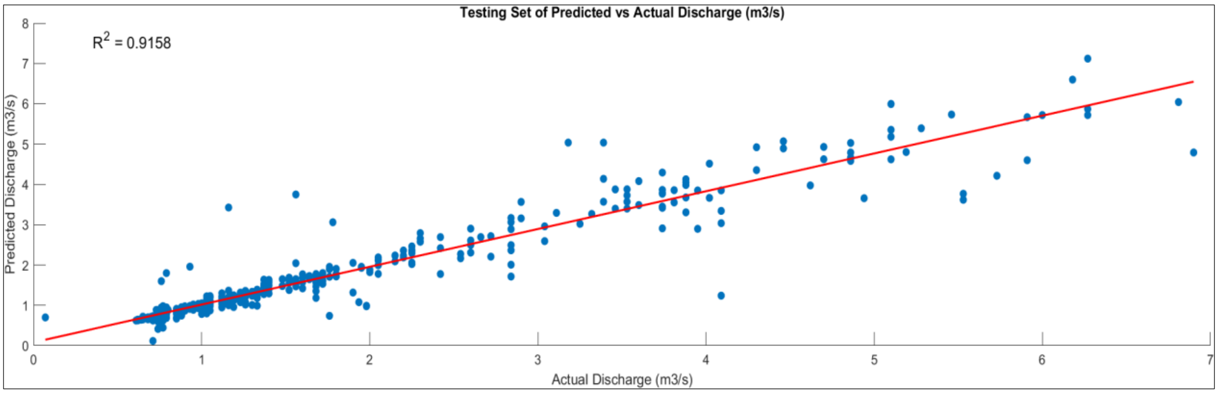


Figure 10: ANN Model 4 Prediction Performance of Best Fitted for Testing Dataset

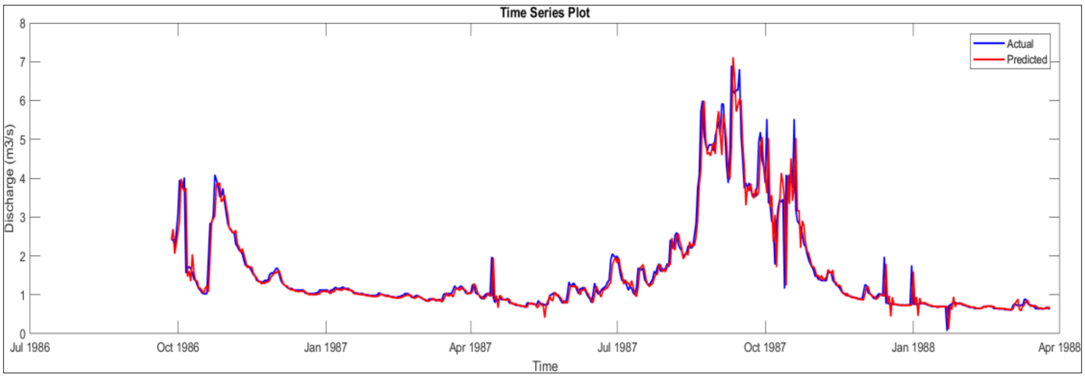


Figure 11: ANN Model 4 Time Series Plots of the Actual vs. Predicted Daily Streamflow for the Testing Dataset

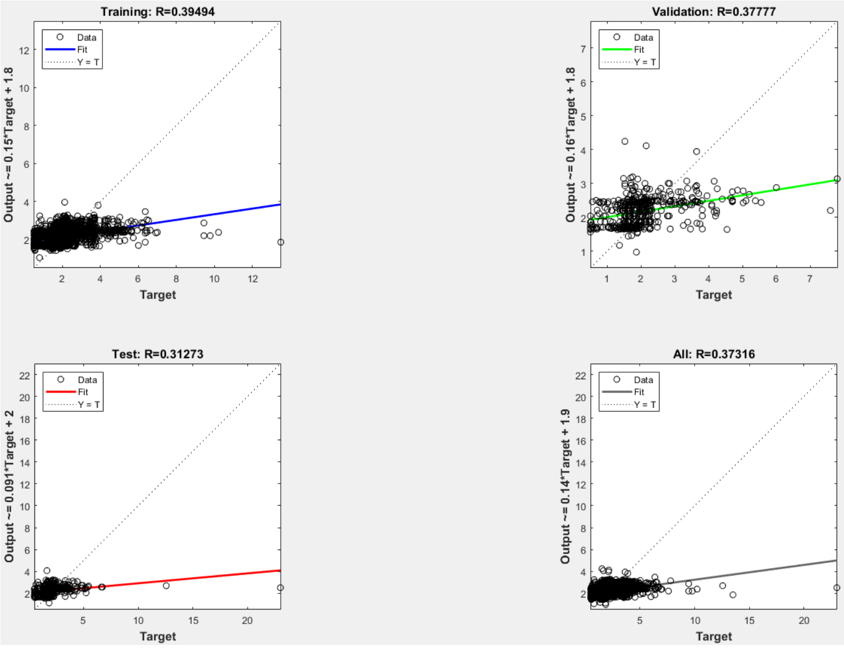


Figure 12: MATLAB-generated automatic plot for ANN Model 1

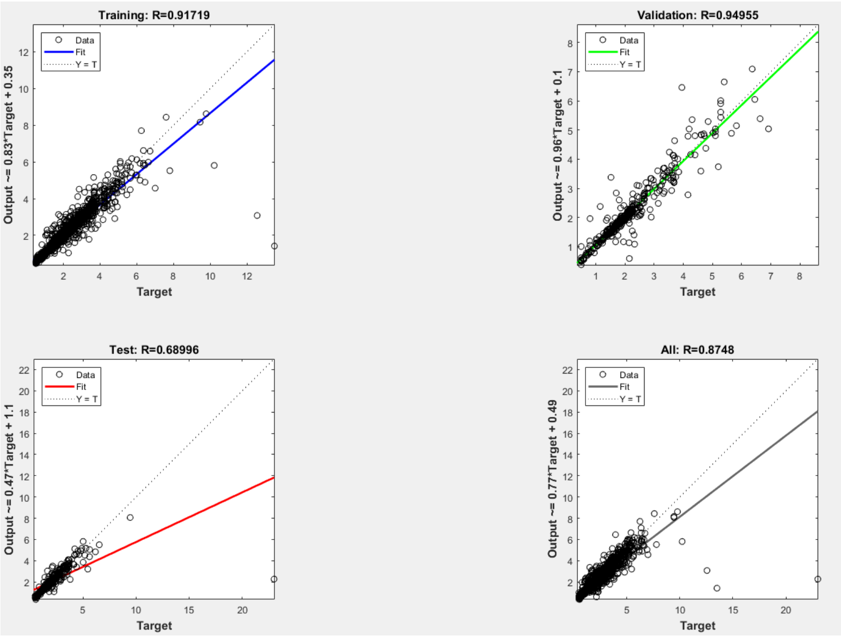


Figure 13: MATLAB-generated automatic plot for ANN Model 2

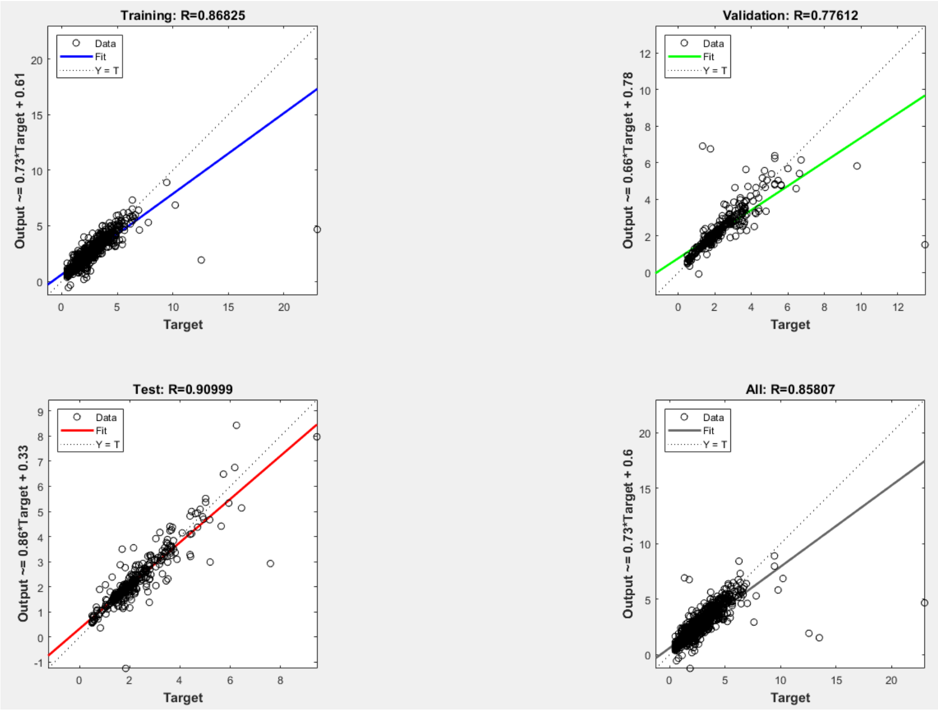


Figure 14: MATLAB-generated automatic plot for ANN Model 4

IV. CONCLUSION

This study highlights the effectiveness of Artificial Neural Networks (ANNs) in modelling the relationship between rainfall and river discharge in the Oramiriukwa River. By employing various ANN architectures and optimizing network parameters, the research aimed to identify the best configuration for accurate streamflow prediction.

Among the different ANN models tested, Model 4, with [15] neurons, demonstrated the best performance, achieving an R^2 value of 0.9158 on the testing dataset. This model also showed low Mean Squared Error (MSE) of 0.1294 and a Root Mean Squared Error (RMSE) of 0.3597, emphasizing its predictive accuracy and strong potential for real-world applications in hydrological modelling.

While Model 2, which utilized a more complex structure of [30, 15, 5] neurons, also performed well with an R^2 value of 0.9029, it came with increased computational demands and a higher risk of overfitting, suggesting that simpler architectures may provide a more balanced approach for this particular task.

In contrast, Model 1, designed with [20, 10] neurons, showed weaker performance, achieving an R^2 of only 0.1823. This indicates that more advanced architectures or additional features, such as historical streamflow data, are needed to improve model accuracy.

In general, the results confirm that ANN-based models, particularly Models 2 to 5, are highly effective for streamflow prediction in the Oramiriukwa River. A key factor in enhancing the performance of these models was the inclusion of lagged streamflow values, which significantly improved the accuracy of the predictions. This approach proved to be highly effective in refining hydrological forecasts for the river. Hence, future studies in the region could benefit from incorporating lagged inputs, further boosting the accuracy and reliability of hydrological models. These models offer a strong foundation for similar applications in other regions. To enhance prediction accuracy and better support decision-making in water resources management, future research should focus on refining model architectures and integrating additional data sources.

REFERENCES

1. Adaka G., Ndukwe E., and Nlewadim A. (2015). Length-Weight Relationship of Some Fish Species in a Tropical Rainforest River in South-East Nigeria. *Transylv. Rev. Syst. Ecol. Res.* 17, 2, 73-78.
2. Adeleye, D.R. (1974). Sedimentology of the fluvial bida sandstone (cretaceous), Nigeria. *Sedimentary Geol.*, 12: 1-24. DOI: 10.1016/0037-0738(74)90013-X
3. Amadi, A.N., Olasehinde, P.I., Okosun, E.A. and Yisa, J. (2010). Assessment of the Water Quality Index of Otamiri and Oramiriukwa Rivers. *Physics International* 1 (2): 102-109.
4. Amangabara, G. (2015). Drainage Morphology of Imo Basin in the Anambra – Imo River Basin Area, of Imo State, Southern Nigeria. *Journal of Geography, Environment and Earth Science International*, 3, 1-11.
5. American Society of Civil Engineers (2000a). Artificial neural networks in hydrology. I:

Preliminary concepts, *Journal of Hydrologic Engineering*, 5(2), 115-123.
6. Behmanesh J., Ayashm S. (2015). Rainfall-runoff Modeling in the Turkey River Using Numerical and Regression Methods. *Journal of Applied Fundamental Sciences*, 7(1), 91-102
7. Dawson, C.W. and Wilby, R.L. (1998). An artificial neural network approach to rainfall runoff modelling. *Hydrological Sciences Journal*, 43, 47-66.
8. Dawson, C.W. and Wilby, R.L. (2000) A comparison of artificial neural networks used for rainfall-runoff modelling. Hydrology and Earth Systems Sciences 3, 529-540.
9. Furundzic, D. (1998). Application example of neural networks for time series analysis: rainfall runoff modelling. *Signal Processing*, 64, 383-396.
10. Gallant, S. (1993). *Neural Network Learning and Expert Systems*. MIT, Massachusetts.
11. Grubbs, (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*. 11(1). 1-21.
12. Hu, T. S., Lam, K.C. and Ng, S.T. (2001). River flow time series prediction with a range-dependent neural network. *Hydrological Science Journal* 46 (5), 729-745.
13. Imrie, C. E., Durucan, S. and Korre, A. (2000). River flow prediction using artificial neural networks: Generalization beyond the calibration range. *Journal of Hydrology*, 233, 138-153.
14. Kalteh, A.M. (2008) *Rainfall-Runoff Modelling Using Artificial Neural Networks (ANNs): Modelling and Understanding*. Caspian Journal of Environmental Sciences, 6, 53-58.
15. Karim, M. F. and Kennedy, J. F. (1990). Menu of Coupled Velocity and Sediment Discharge Relations for Rivers. *Journal Hydrology Engineering*.116(8). 978- 996.
16. Kitamura, Y. and Nakayama, H. (1985). *Rainfall-runoff in the Catchment Area of Muda and Pedu Dams*. Quarterly report no. 18, TARC, Alor Setar.
17. Machado, F., Mine, M., Kaviski, E. & Fill, H. (2011) Monthly rainfall–runoff modelling using artificial neural networks. *Hydrol. Sci*. J. 56(3), 349–361.
18. Napolitano, G. (2011). *An Exploration of Neural Networks for Real-time Flood Forecasting*, [PhD Thesis]. The University of Leeds School of Geography.
19. Riad, S., Mania, J., Bouchaou, L. and Najjar, Y. (2004). Rainfall-Runoff Model Using an Artificial Neural Network Approach. *Mathematical and Computer Modelling* 40, 839-846 www.sciencedirect.com
20. Sitterson, J., Knightes C., Parmar, R., Wolfe, K., Muche, M., and Avant, B. (2017). *An Overview of Rainfall-Runoff Model Types*. U.S. Environmental Protection Agency, Washington DC, EPA/600/R-17/482.
21. Sudheer, K.P. and Jain, A. (2004). Explaining the internal behaviour of artificial neural network river flow models. *Hydrological Process*, 118(4), 833-844.
22. Tanty, R. and Desmukh, T. S. (2015). Application of Artificial Neural Network in Hydrology- A Review. *International Journal of Engineering Research & Technology*, 4(06), 184-188
23. Tokar, A. S. and Johnson, P. A. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4 (3), pp. 232-239, ASCE
24. Vos, N.J. (2003). *Rainfall-Runoff Modelling Using Artificial Neural Networks*. [Master's Thesis] Delft University of Technology
25. Wei, Y., Xu, W., Fan, Y., and Tasi, H. T. (2002). Artificial neural network based predictive method for flood disaster. *Computers ~ Industrial Engineering* 42, 383-390.