ECWS-7 2023

The 7th International Electronic Conference on Water Sciences

15–30 MARCH 2023 | ONLINE

Rainfall-Runoff Modelling Using Artificial Neural Network-a Case Study of Purna Sub-catchment of Upper Tapi Basin, India

Usman Mohseni1, Sai Bargav Muskula2

1,2Research Scholar, Department of Civil Engineering, IIT Roorkee, Roorkee, INDIA

INTRODUCTION

- Rainfall-runoff modelling is one of the most prominent hydrological models used to examine the relation between rainfall and runoff (Namara et al., 2020).
- These methods can be divided into two categories: (i) conceptual models that take into account the physics of the underlying process, and (ii) data-driven models that learn and behave based on the information in the data without taking into account the physics of the system (J. Chen & Adams, 2006).
- The process of rainfall-runoff is highly nonlinear and incredibly complex, and because it is interconnected with various subprocesses involved in the hydrologic cycle (Zhang & Govindaraju, 2000).

INTRODUCTION TO ANN

- Artificial Neural Networks (ANNs) are data-processing systems that mimic the human brain's capabilities(Kisi et al., 2013).
- ANN architecture consists of three layers i.e. input layer, hidden layer and output layer
- Artificial neural network (ANN) models, sometimes known as black-box models, have been effectively employed for simulating complicated hydrological phenomena (Kumar et al., 2011).
- ANN models have been extremely prevalent in the domains of hydrology, water resources and watershed (Orimi et al., 2015).

OBJECTIVES OF THE STUDY

- To develop a rainfall-runoff model for Upper Tapi using Artificial Neural Networks Technique.
- To compare ANN rainfall-runoff models developed using nntool with different neural network types i.e., FFBPNN and CFBPNN.
- To compare ANN rainfall-runoff models trained using LM, BR and SCG algorithms.
- This study attempts to improve hydrological forecasting and to determine which models' best suit rainfall-runoff modelling.

LITERATURE REVIEW

(Mitra & Nigam, 2021)

(Dumka & Kumar, 2021)

(Mahsa et al., 2021)

(Poonia & Tiwari, 2020)

(Dey, 2020)

(Obasi et al., 2020)

(Vidyarthi et al., 2020)

(Yadav et al., 2020)

(Samantaray & Sahoo, 2020)

(Samantaray et al., 2019)

(Romlay et al., 2019)

(Fatih et al., 2019)

(Hussain et al., 2019)

(Asadi et al., 2019)

(Patel & Joshi, 2017)

(Hussain et al., 2017)

(Tayebiyan et al., 2016)

(Sinha et al., 2015)

(Chen et al., 2013)

(Bakshi & Bhar, 2012)

(Nayak et al., 2013)

(Sarkar & Kumar, 2012)

(Nemade et al., 2012)

(Kalteh, 2008)

(Senthil Kumar et al., 2005)

(Rajurkar et al., 2004)

(Joshi, 2001)

(Mittal, 1943)

❑CLOSURE OF REVIEW:

- Various rainfall-runoff modelling studies using ANN technique have been conducted in various parts of the world. The above literature review helped to provide an understanding of the Methodology adopted for rainfall-runoff modelling. The literature review shows, the ANN as a widely used for R-R modelling and hence the current study adopted this model.

STUDY AREA & DATA COLLECTION

- The current study area comprises a portion of the Upper Tapi Basin known as the Purna sub-catchment.
- The area lies between Maharashtra and Madhya Pradesh, between latitudes of 20°09'N to 22°03'N and longitudes of 75°56'E to 78°17'E.
- The mean annual precipitation in the chosen area varies from 833 to 990 mm.

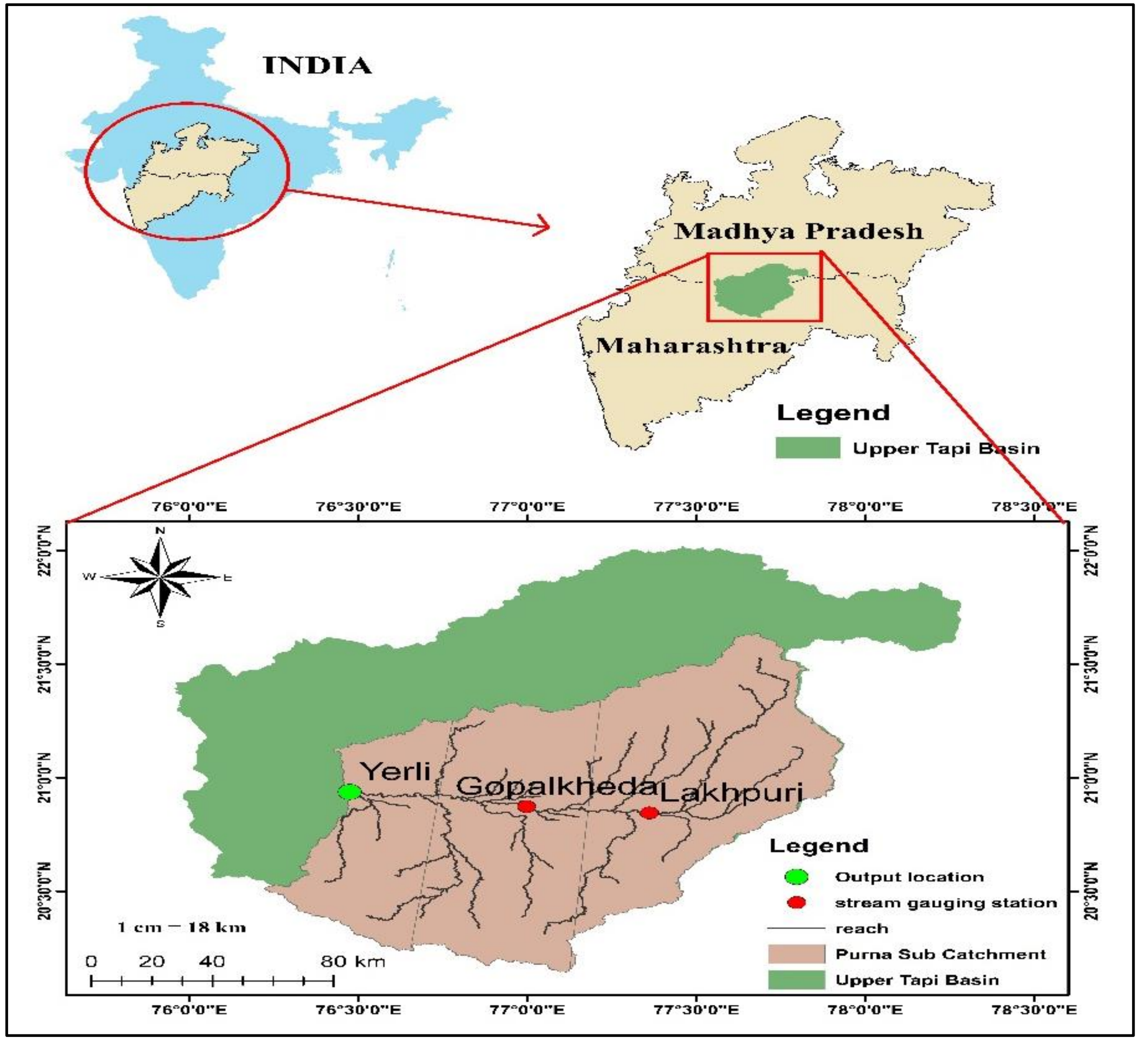

Figure 1. Index Map of Study Area.

Table 1. Source of Data.

Data Type	Data Source
Digital Elevation Model (DEM)	USGS Earth Explorer
Rainfall Data	Central Water Commission
Discharge Data	Central Water Commission
Meteorological data	www.power.larc.nasa.gov

METHODOLOGY

- **Data Collection:**
- **Import Data:**
- **Creating Network:**
- **Number of Neurons:**
- **Network Training:**
- **Result:**
- **Retraining:**
- **Model Evaluation:**

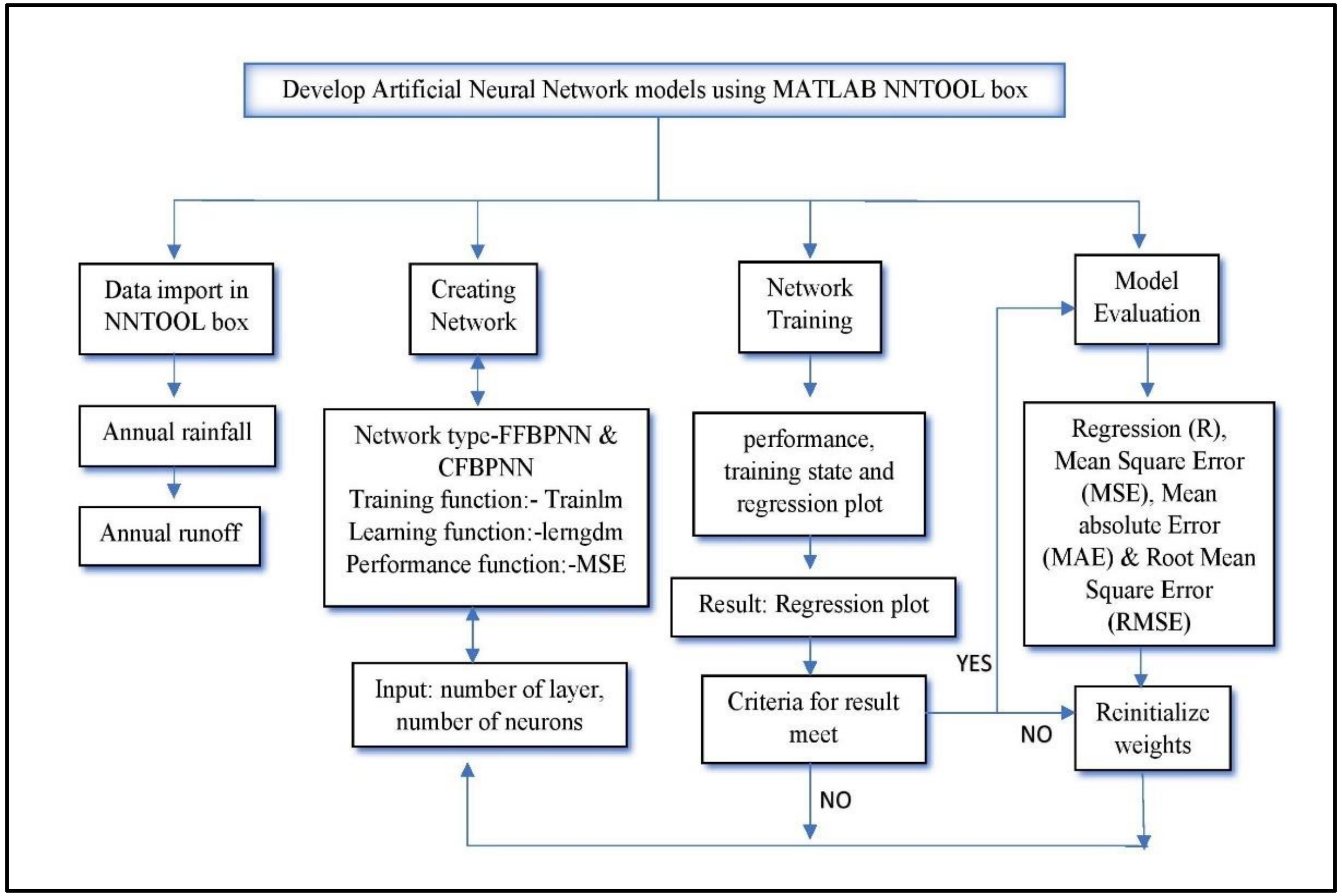

Figure 2. Flow chart of NNTOOL

- **Data Collection**
- **Neural Fitting App**
- **Data Selection:**
- **Validation and Test:**
- **Network Architecture:**
- **Select Algorithm:**
- **Train Network:**
- **Retrain:**
- **Output:**

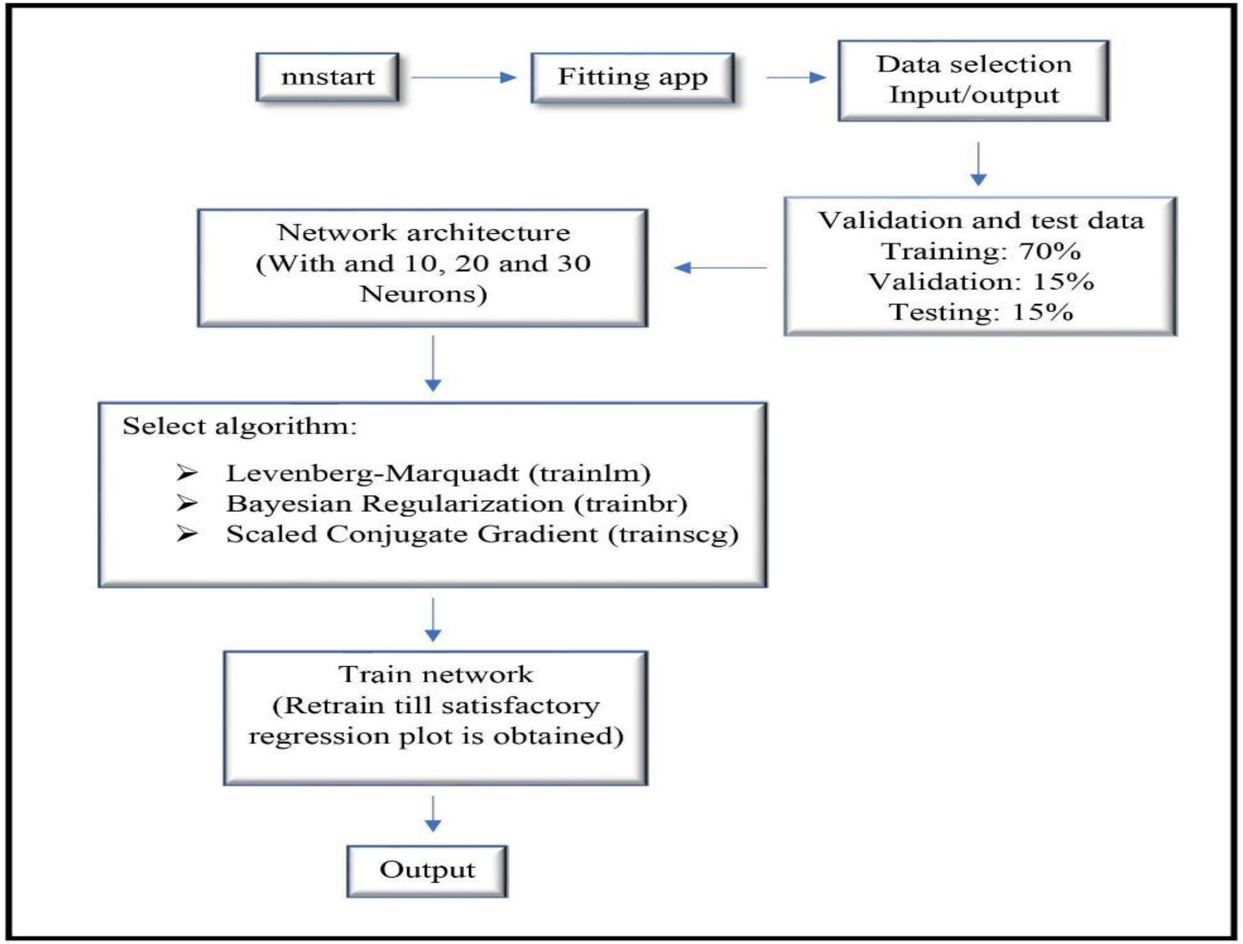

Figure 3. Flow chart of nnstart

MODEL DEVELOPMENT

- The ANN contains three layers such as an input layer, a hidden layer and an output layer.

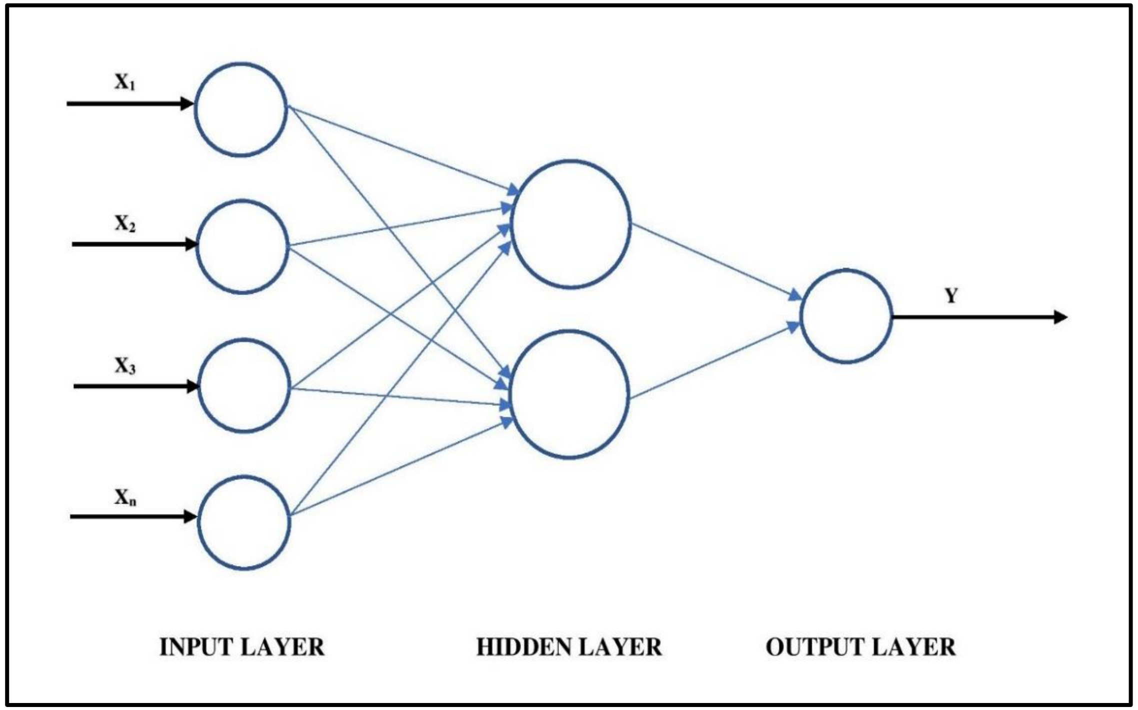

Figure 4. ANN Model

MODEL USING NNTOOL

- Two different models were developed, i.e., Feed Forward Back Propagation Neural Network (FFBPNN) Cascade Forward Back Propagation neural network (CFBPNN) with three different architecture (6-2-1, 6-3-1 and 6-4-1) using several combinations of transfer functions i.e. (transig, logsig and purelin) along with two sets of neurons 10 and 20 and then compared their capability for estimation of flow for the period 1981-2016.
- **(FFBPNN):** FFBPNN as the training method is a popular training technique that is commonly used to model hydrologic problems.
- Figure 5 depicts the structure of the FFBPNN.

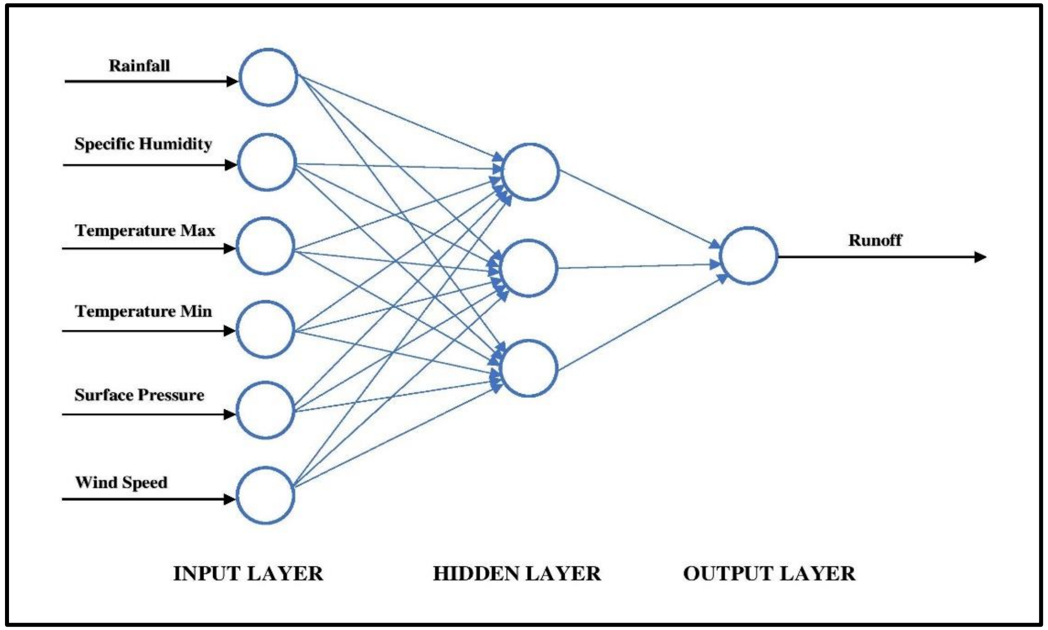

Figure 5. FFBPNN 3 Layer Model.

❑ (CFBPNN):

- Cascade Forward neural networks are similar to FFNNs, except they have a weighted link from the input to each layer, which is then connected to subsequent layers.
- In CFBPNN, neurons in one layer are responsible for computing and updating the weights of all layers in front of them.
- This model is similar to FFBPNN, except it uses the back propagation algorithm to adjust the weights.

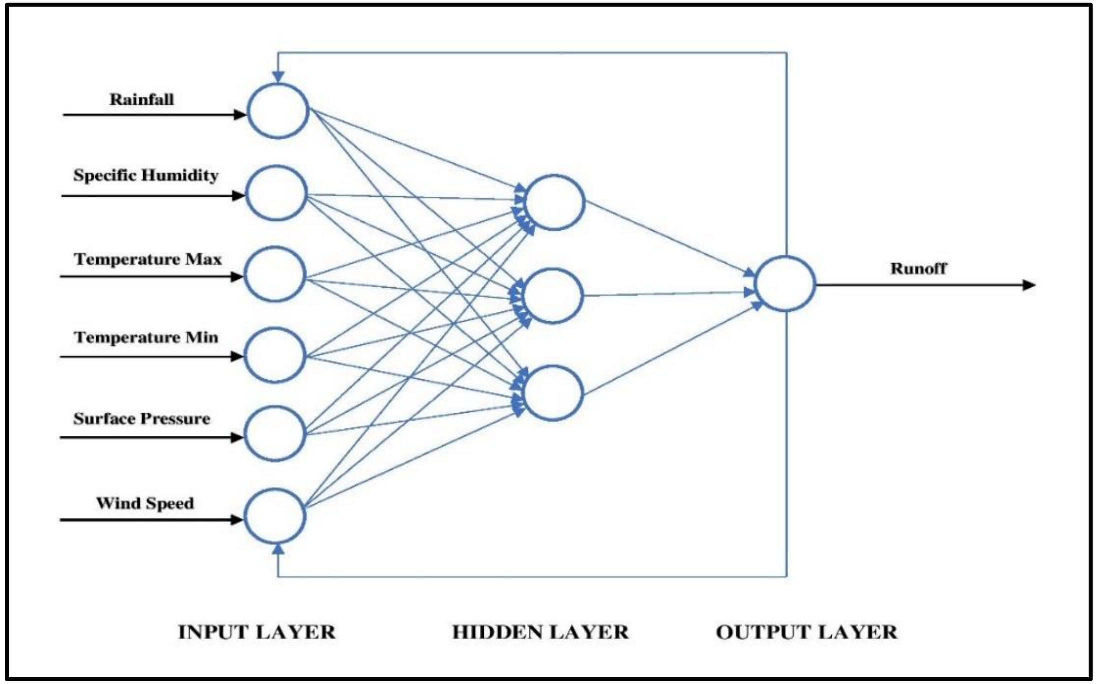

Figure 6. CFBPNN 3 Layer Model.

MODEL USING NNSTART

- In this study, three different algorithms namely Levenberg Marquardt (trainlm), Bayesian Regularization (trainbr) and Scaled Conjugate Gradient (trainscg) were used for model development.
- Three different models were developed based on the algorithms used. Each model was developed for 36 samples (1981-2016).
- 70% of the 36 samples are used for training, 15% for validation, and 15% for testing.
- The number of Neurons adopted by each model were 10,20,30,40,50 and 60 for better performance of the network during training. Finally, all the models were compared based on different model performance evaluation criteria and demonstrated the best model.

ALGORITHMS USED

- **Levenberg Marquardt:** This algorithm is quick but consumes more memory.

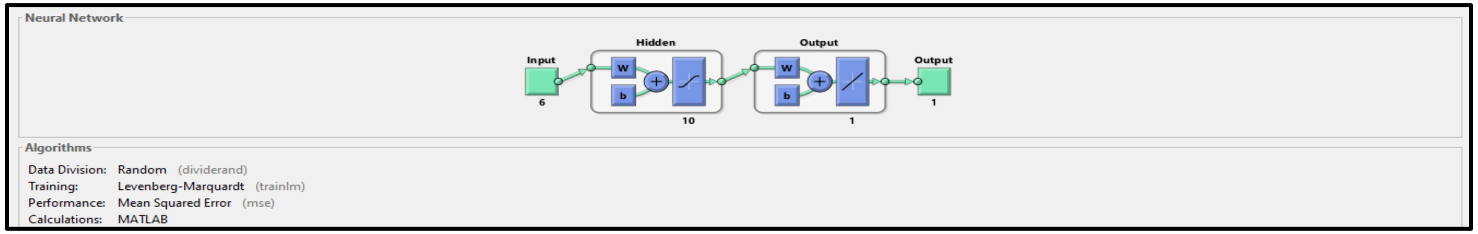

Figure 7. The Architecture of Levenberg Marquardt algorithm.

- **Bayesian Regularization:** This algorithm takes longer, it can provide strong generalisation for complex, tiny, or noisy datasets.

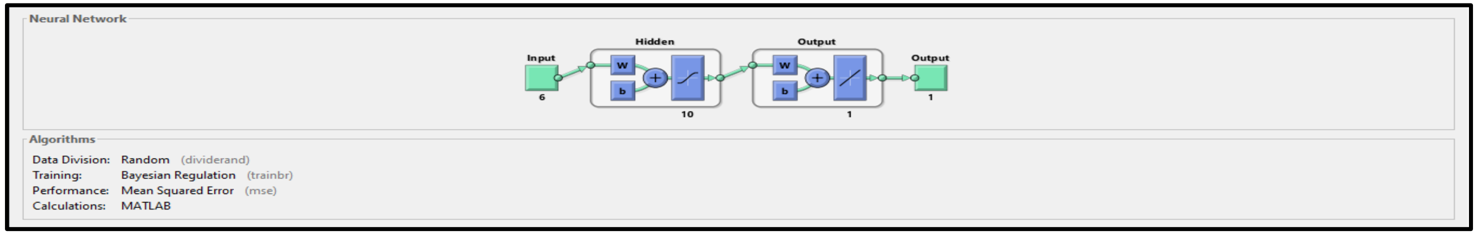

Figure 8. The architecture of Bayesian Regularization algorithm.

- **Scaled Conjugate Gradient:** This algorithm uses less memory.

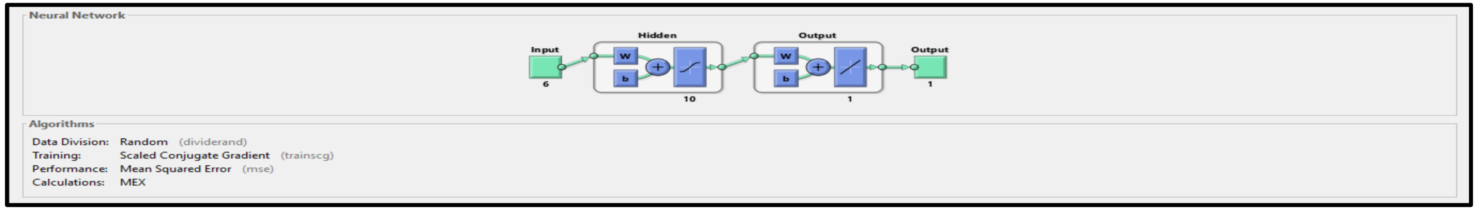

Figure 9. The Architecture of Scaled Conjugate Gradient algorithm.

MODEL EVALUATION CRITERIA

- The findings of the ANN model applied in this study were evaluated by means of:
- Mean square error (MSE):

$$\text{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{i=\mathbf{1}}^{n}(Q_p - Q_o)^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. (1)$$

- Root mean square error (RMSE):

$$\text{RMSE}= \left[\frac{\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}}(\boldsymbol{Q}(\boldsymbol{i})-\widehat{\boldsymbol{Q}}(\boldsymbol{i}))^{\mathbf{2}}}{\boldsymbol{n}}\right]^{\mathbf{0.5}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(2)$$

- Regression Coefficient (R): Using Regression Plot between predicted and observed runoff.

Where,

Q_p is the value of predicted runoff

Q_o is the value of observed runoff

$\hat{Q}(i)$ is the n estimated runoff value

Q*(i)* is the n observed runoff value

RESULTS & DISCUSSION

❑ FFBPNN:

- Table 2 contains the inclusive outcomes.

Table 2. Results of FFBPNN for Yerli station.

Transfer Function	Network Architecture	Number of Neurons	MSE	RMSE	R
Transig	6-2-1	10	0.7905	0.8891	0.94204
Transig	6-3-1	10	0.6872	0.8289	0.91321
Transig	6-4-1	10	0.6618	0.8135	0.92072
Transig	6-2-1	20	0.6028	0.7764	0.94418
Transig	6-3-1	20	0.5750	0.7582	0.93744
Transig	6-4-1	20	0.4982	0.7056	0.96213
Logsig	6-2-1	10	1.2126	1.1011	0.94575
Logsig	6-3-1	10	1.0872	1.0426	0.91362
Logsig	6-4-1	10	0.9861	0.9930	0.83050
Logsig	6-2-1	20	0.9812	0.9905	0.93820
Logsig	6-3-1	20	0.9794	0.9896	0.94116
Logsig	6-4-1	20	0.9713	0.9532	0.95182
purelin	6-2-1	10	1.3811	1.1752	0.77560
purelin	6-3-1	10	1.2972	1.1389	0.81533
purelin	6-4-1	10	1.1988	1.0948	0.88452
purelin	6-2-1	20	1.1697	1.0815	0.77055
purelin	6-3-1	20	1.0836	1.0409	0.86730
purelin	6-4-1	20	0.9891	0.9945	0.90754

- The transig transfer function with architecture 6-4-1 gives better results in the current study. Figure 10 depicts the best regression plot.

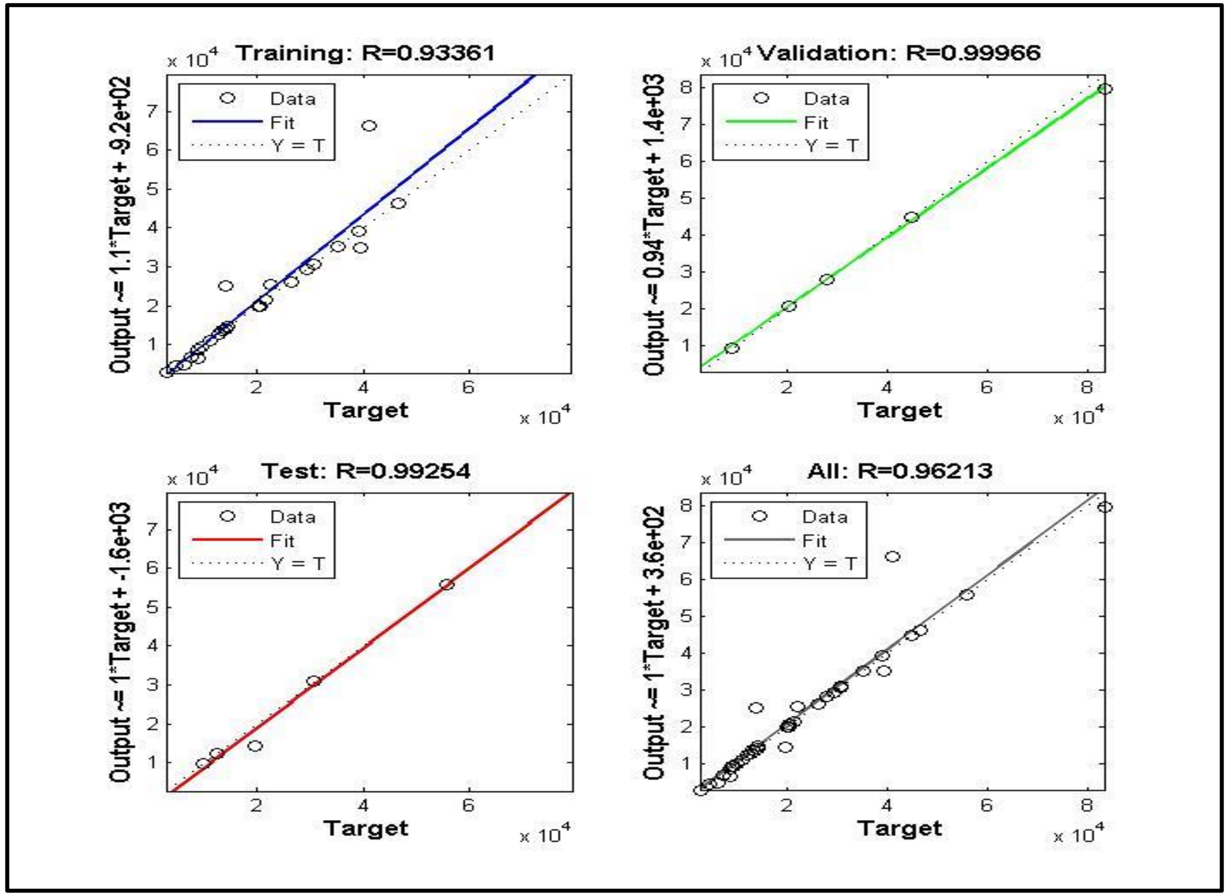

Figure 10. Regression plot for FFBPNN Transig 6-4-1 model.

- **CFBPNN**

While considering 6-2-1, 6-3-1, and 6-4-1 architectures, the transig function provides the best value for performance. The most effective model architecture for the Transig function is 6-4-1, which has MSE values of 0.8813, the value of RMSE 0.9387, and the value of R 0.96096.

Table 3. Results of CFBPNN for yerli station

Transfer Function	Network Architecture	Number of Neurons	MSE	RMSE	R
Transig	6-2-1	10	1.6720	1.2930	0.91317
Transig	6-3-1	10	1.5852	1.2590	0.91972
Transig	6-4-1	10	1.5169	1.2316	0.92880
Transig	6-2-1	20	1.4723	1.2134	0.92438
Transig	6-3-1	20	1.0288	1.0142	0.94060
Transig	6-4-1	20	0.8813	0.9387	0.96096
Logsig	6-2-1	10	1.2124	1.1010	0.92847
Logsig	6-3-1	10	1.0696	1.0342	0.90271
Logsig	6-4-1	10	0.9713	0.9855	0.89329
Logsig	6-2-1	20	0.9644	0.9829	0.90213
Logsig	6-3-1	20	0.9277	0.9631	0.93750
Logsig	6-4-1	20	0.8904	0.9436	0.94575
Purelin	6-2-1	10	1.5135	1.2302	0.76575
Purelin	6-3-1	10	1.3866	1.1775	0.82150
Purelin	6-4-1	10	1.2052	1.0978	0.89772
Purelin	6-2-1	20	1.1745	1.0837	0.76613
Purelin	6-3-1	20	1.0289	1.0143	0.86520
Purelin	6-4-1	20	0.9916	0.9957	0.98270

- The transig transfer function with architecture 6-4-1 gives better results in the current study. Figure 11 depicts the best regression plot.

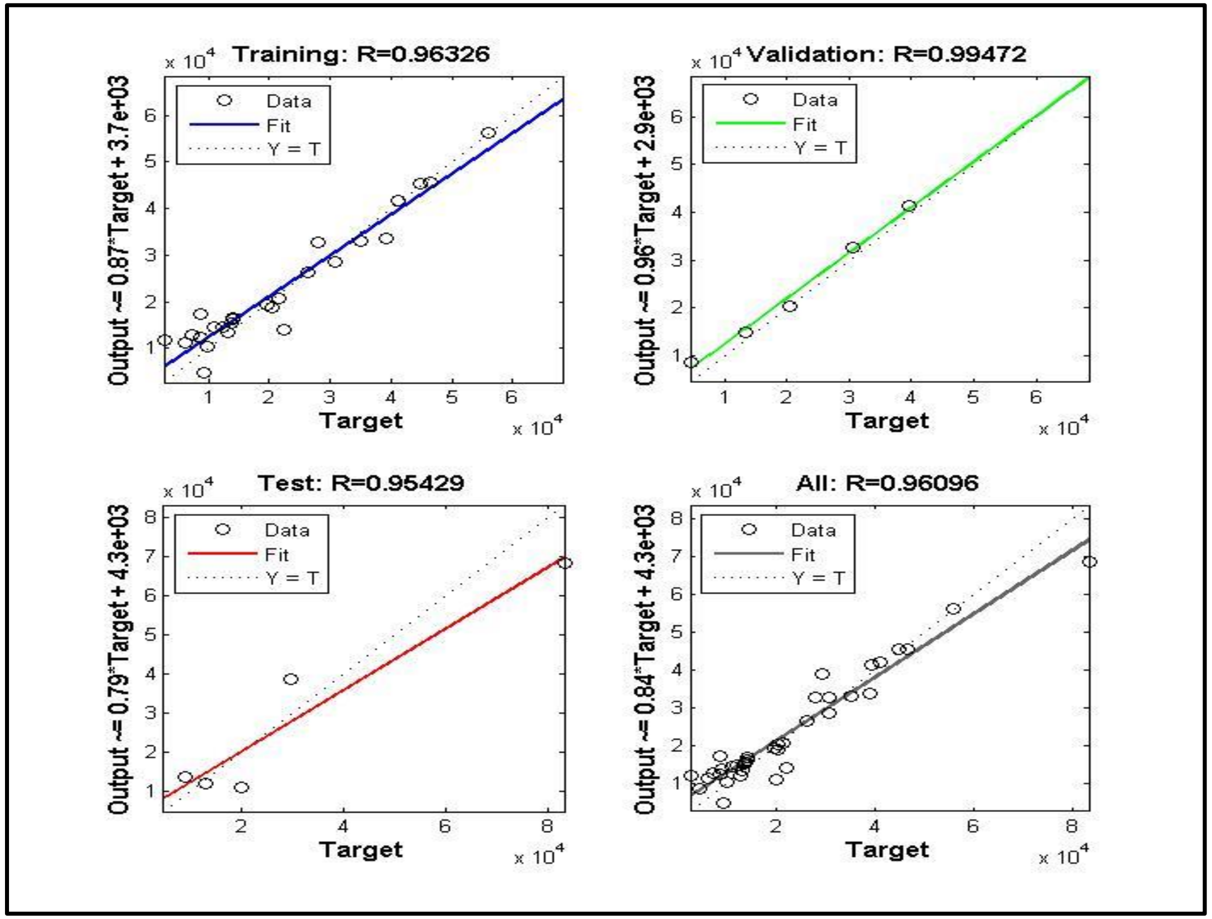

Figure 11. Regression plot for CFBPNN Transig 6-4-1 model.

NNSTART

- Table 3 shows the yerli station results for ANN trained by LM, BR, and CGS. The values based on R, MSE, and RMSE are used to characterize the optimum network architecture. The best number of neurons in the hidden layer and the best type of model are displayed in the best configuration column for each unique learning algorithm on the yerli station.

Table 4. Results of nnstart for yerli station.

Algorithm	Number of Neurons	MSE	RMSE	R
LM	10	0.9774	0.9886	0.89887
LM	20	0.8557	0.9250	0.90763
LM	30	0.7279	0.8531	0.95057
BR	10	1.1745	1.0837	0.85428
BR	20	0.9669	0.9833	0.91349
BR	30	0.8133	0.9018	0.92790
CGS	10	1.7262	1.3138	0.91291
CGS	20	1.1680	1.0807	0.92185
CGS	30	0.9086	0.9532	0.94573

- According to the findings, the ANN trained using LM attained the lowest value of MSE with the fewest number of iterations. However, ANN is trained by BR achieved a good MSE value, but it required the most iterations and had a propensity to overfit. ANN trained using CGS, on the other hand, achieves the maximum number of MSE values with the fewest iterations. Figure 12 Shows the best regression plot for the LM algorithm with 30 neurons.

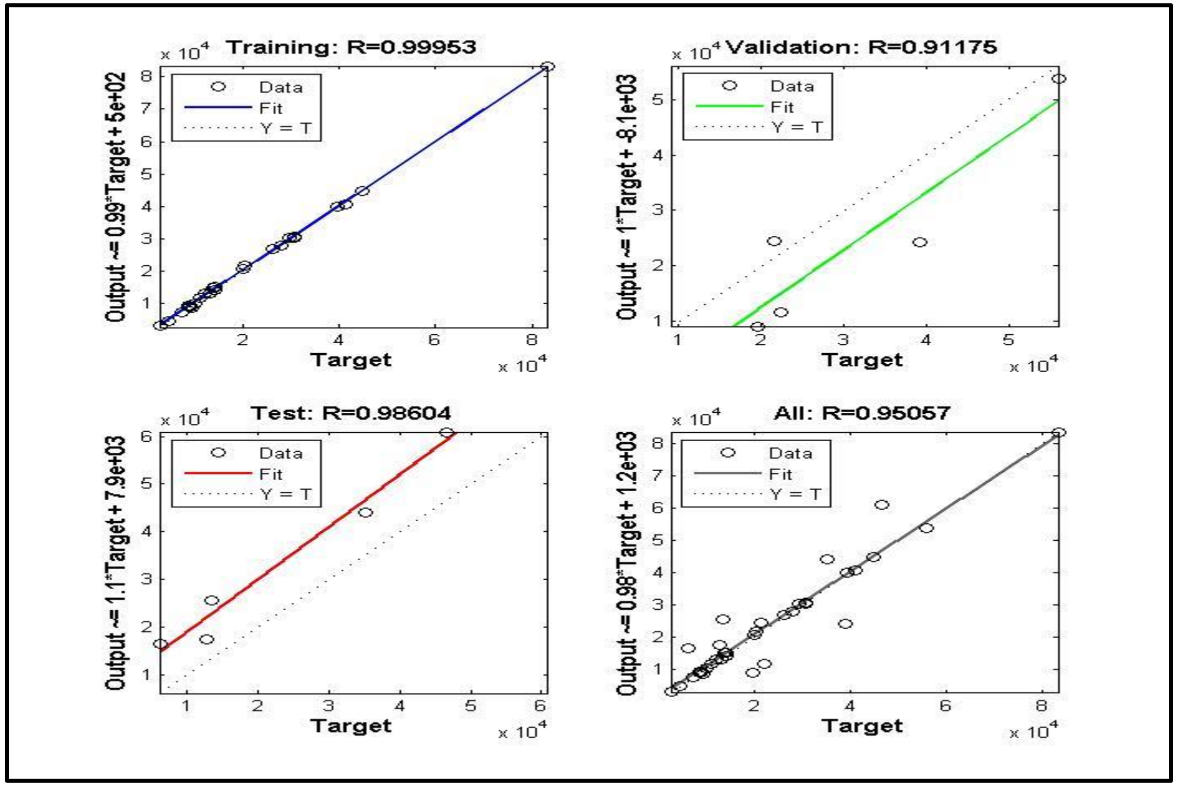

Figure 12. Regression plot for LM algorithm with 30 Neurons.

CONCLUSION

- Using the NNTOOL for the Transig function in FFBPNN, the most prominent model architecture is 6-4-1, which has MSE values of 0.4982, a value of RMSE 0.7056, and the value of R 0.96108.
- The 6-4-1 model architecture for the Transig function is the most effective for CFBPNN, with MSE values of 0.8813, RMSE values of 0.9387, and R values of 0.96096.
- Using nnstart three different algorithms LM, BR and CGS were used to predict yearly runoff. . Among the three, LM trained the algorithm with 30 neurons is the best model with MSE values of 0.7279, RMSE values of 0.8531, and R values of 0.95057.
- According to the findings, FFBPNN predicts better results than CFBPNN, and the LM algorithm stands out among the other algorithms.
- The current work contributes to policymakers' ability to design successful flood management measures in the near future by providing vital information on rainfall-runoff modeling at a regional scale.

REFERENCES

- A.M.Kalteh. (2008). Caspian Journal of Environmental Sciences. Caspian Journal of Environmental Sciences, 6(1), 53–58.
- Asadi, H., Shahedi, K., Jarihani, B., & Sidle, R. C. (2019). Rainfall-runoff modelling using hydrological connectivity index and artificial neural network approach. Water (Switzerland), 11(2). https://doi.org/10.3390/w11020212
- Bakshi, S., & Bhar, K. K. (2012). Estimation of Discharge in Rivers using ANN Modeled Rating Curves. Journal of The Institution of Engineers (India): Series A, 93(3), 181–186. https://doi.org/10.1007/s40030-013-0020-4
- Chang, T. K., Talei, A., Quek, C., & Pauwels, V. R. N. (2018). Rainfall-runoff modelling using a self-reliant fuzzy inference network with flexible structure. Journal of Hydrology, 564(July), 1179–1193. https://doi.org/10.1016/j.jhydrol.2018.07.074
- Chen, J., & Adams, B. J. (2006). Integration of artificial neural networks with conceptual models in rainfall-runoff modeling. Journal of Hydrology, 318(1–4), 232–249. https://doi.org/10.1016/j.jhydrol.2005.06.017
- Chen, S. M., Wang, Y. M., & Tsou, I. (2013). Using artificial neural network approach for modelling rainfall-runoff due to typhoon. Journal of Earth System Science, 122(2), 399–405. https://doi.org/10.1007/s12040-013-0289-8
- Dey, K. K. (2020). A rainfall-runoff model using artificial neural networks for the district of Bankura in a time of climate change. Indian Journal of Science and Technology, 13(33), 3364–3376. https://doi.org/10.17485/ijst/v13i33.816

REFERENCES

- Nayak, D. R., Mahapatra, A., & Mishra, P. (2013). A Survey on Rainfall Prediction using Artificial Neural Network. Int. J. Comput. Appl, 72(16), 32–40.
- Nemade, P. S., Gupta, P. V., & Bhoite, P. A. (2012). Comparative Rainfall Prediction Analysis Year 2012 for Indore (MP) Region , India by Using Artificial Neural Network. 3(5), 3–6.
- Orimi, M. G., Farid, A., Amiri, R., & Imani, K. (2015). Cprecip parameter for checking snow entry for forecasting weekly discharge of the Haraz River flow by artificial neural network. Water Resources, 42(5), 607–615. https://doi.org/10.1134/S0097807815050073
- Patel, A. B., & Joshi, G. S. (2017). Modeling of Rainfall-Runoff Correlations Using Artificial Neural Network-A Case Study of Dharoi Watershed of a Sabarmati River Basin, India. Civil Engineering Journal, 3(2), 78–87. https://doi.org/10.28991/cej-2017-00000074
- Rajurkar, M. P., Kothyari, U. C., & Chaube, U. C. (2004). Modeling of the daily rainfall-runoff relationship with artificial neural network. Journal of Hydrology, 285(1–4), 96–113. https://doi.org/10.1016/j.jhydrol.2003.08.011
- Samantaray, S., & Sahoo, A. (2020). Prediction of runoff using BPNN, FFBPNN, CFBPNN algorithm in arid watershed: A case study. International Journal of Knowledge-Based and Intelligent Engineering Systems, 24(3), 243–251. https://doi.org/10.3233/KES-200046
- Tayebiyan, A., Mohammad, T. A., Ghazali, A. H., & Mashohor, S. (2016). Artificial neural network for modelling rainfall-runoff. Pertanika Journal of Science and Technology, 24(2), 319–330

Thank You for your Attention.