*Jordan Journal of Civil Engineering*, Volume **15**, No. **3**, **2021**

# A Simplified Approach for Rainfall-runoff Modeling Using Advanced Soft-computing Methods

***Deepak Kumar*** [1)]***, Thendiyath Roshni*** [2)*]***, Anshuman Singh*** [3)]***, Dar Himayoun*** [4)] ***and Pijush Samui*** [5)]

[1)] Department of Civil Engineering, National Institute of Technology Patna, Ashok Raj Path, Patna 800005, India. E-Mail: deepak.ce15@nitp.ac.in
[2)] Assistant Professor, Civil Engineering Department, National Institute of Technology Patna, Ashok Raj Path, Patna, 800005, India. * Corresponding Author. E-Mail: roshni@nitp.ac.in
[3)] Associate Professor, Civil Engineering Department, National Institute of Technology Patna, Ashok Raj Path, Patna, 800005, India. E-Mail: asingh@nitp.ac.in
[4)] Department of Civil Engineering, National Institute of Technology Patna, Ashok Raj Path, Patna 800005, India. E-Mail: himayoundar@gmail.com
[5)] Associate Professor, Civil Engineering Department, National Institute of Technology Patna, Ashok Raj Path, Patna, 800005, India. E-Mail: pijush@nitp.ac.in

## ABSTRACT

This study investigates three modeling approaches based on Xtreme Boosting Machine (XGBoost), Genetic Algorithm-optimized Emotional Neural Network (GA-EmNN) and Group Method of Data Handling-Neural Network (GMDH-NN) for rainfall-runoff modeling. The redundancy capability of Principal Component Analysis (PCA) was applied to solve the problems of input selection in these machine learning models. Hence, the objective of this study is to develop an efficient approach to pre-process the data structure and ensemble it with machine learning models that produce a higher predictive accuracy. The three XGBoost, GA-EmNN and GMDH-NN models are trained and validated for streamflow forecasting using monthly rainfall and monthly runoff data of the Jhelum basin, India. Statistical fitness indices, like correlation (r), Root Mean Square Error (RMSE), relative Nash–Sutcliffe Efficiency coefficient (r NSE), Percent Bias (PBias) and Kling-Gupta Efficiency (KGE), were used for model performance assessment. The analysis of results revealed that GA-EmNN model was well capable (Training: R=0.999, RMSE = 345.48 cumec, Testing: R=0.997, RMSE = 428.35 cumec) of predicting the streamflow, followed by GMDH-NN and XGBoost. The results of this approach would benefit future modeling efforts, by underlining the application of these methods in hydrological modeling.

**KEYWORDS:** Rainfall-runoff, PCA, XGBoost, Emotional neural network, GMDH-type neural network.

## INTRODUCTION

Streamflow modeling is very important for a number of activities that are associated with the planning, management and operation of water resource systems across the world (Mulvaney, 1851; Joshi and Yadav, 2021). Streamflow prediction techniques are broadly categorized into two groups: (1) non-physically-based and (2) physically-based. The first group is processing-based and the second group is based on pattern recognition (Bourdin et al., 2012). Physical process-based streamflow prediction involves various hydrological attributes, such as rainfall, runoff, evaporation, temperature and interception (Avdullahi et al., 2012). Moreover, the development of such physically-based models requires a great deal of understanding of the processes and mechanisms for hydrological system modeling. Additionally, it requires a large quantity of high-quality data covering the hydrological system with high pre-processing computational power which is usually a cumbersome

Received on 2/2/2021.
Accepted for Publication on 17/5/2021.

process, especially for large catchments (Kokkonen and Jakeman, 2001). Owing to these drawbacks of physically-based methods, the pattern recognition methods (data-driven models) have gained tremendous attention in recent times and now, data-driven approaches have matured due to their sound mathematical background and availability of cheap computation power (Bobba et al., 2000; Bowden et al., 2005; Brunner, 2010; Chow, 1964; Douinot et al., 2017; Li et al., 2015; McIntyre and Wheater, 2004; Moazenzadeh et al., 2018; Mosavi et al., 2018; Taormina et al., 2015; Voinov et al., 2004; Wu et al., 2014). The advantage of such models is that they do not require complex physical equations and parametric assumptions, but rather directly relate streamflow to its hydrological attributes using the pattern recognition approach.

Despite the capability of these Artificial Intelligence (AI)-based models in streamflow forecasting, they suffer from serious drawbacks when the input data series involve seasonal variations and non-stationary behavior. Moreover, their predictive performance largely depends upon the quantity and quality of data. Abrahart et al., (2007) showed that multi-collinearity and noise persistence can influence the performance of forecasting models. In order to overcome these short comings and achieve reliable accuracy of prediction, several pre-processing techniques have been applied in the past, such as Wavelet Transformation (WT) (Kalteh, 2016; Shoaib et al., 2018), Singular Spectrum Analysis (SSA) (Wang et al., 2014; Wu and Chau, 2011), Moving Average Filtering (MAF) (Mehr and Kahya, 2017) and ensemble Empirical Mode Decomposition (eEMD) (Wang et al., 2015). The critical appraisal of these techniques shows that such studies focus more on quality and redundancy of the dataset. Therefore, in this study, Principal Component Analysis (PCA) is adopted to extract significant input features based on the total variance explained by each individual rainfall data (Bartoletti et al., 2018; Remesan et al., 2018). Further, PCA significantly reduces the complexity of the inputs through dimensionality based on contribution by individual attributes.

In addition to this pre-processing technique, several researchers have also investigated the advanced AI techniques for R-R modeling to attain better forecasting accuracy. Moosavi et al. (2017) investigated the discrete wavelet and Wavelet-based Group Methods of Data Handling (W-GMDH) model for runoff forecasting and concluded that pre-processing techniques can improve the performance of runoff forecasting. Nourani (2017), Sharghi et al. (2019) and Sharghi et al. (2018) have used emotional neural network and its variant with wavelet to investigate the feasibility of AI technique to mitigate the non-stationarity in the dataset. Solomatine and Dulal (2003) have used tree-based models and ANN for R-R modeling and concluded that model trees can be an alternative to ANNs. Furthermore, Galelli and Castelletti (2013) have for the first time investigated tree-based ensemble method (CART and M5) for streamflow forecasting and compared it with data-driven approach (ANNs and multiple linear regression) and their results showed that ensemble method performs comparatively better and requires less computations on large datasets. Yaseen et al. (2018) investigated extended version of extreme learning machine (EELM) model in river-flow modeling and compared its performance with support vector machine (SVM) and ELM. They found that EELM has a better prediction accuracy than SVM and ELM. Chau (2017) emphasized and demonstrated the use of meta-heuristic techniques based on the data-driven approach for R-R modeling.

Based on the above critical appraisal, we are motivated to use the advanced machine learning algorithms in conjunction with the appropriate data pre-processing technique. In this article, we applied PCA as a pre-processing method for the selection of input variables considering different rain gauge stations. In addition, we employed relatively new machine learning algorithms, such as Xtreme Boosting Machine (XGBoost), Genetic Algorithm-optimized Emotional Neural Network (GA-EmNN) and Group Method of Data Handling-Neural Network (GMDH-NN) to establish a relationship between the input (rainfall) and output flow in a compact and flexible way. Comparative study has been performed among the selected models to test the adaptability of GA-EmNN model for rainfall-runoff (R-R) modeling.

## Theoretical Background

### *Principal Component Analysis (PCA)*

PCA is a standard approach for dimension reduction and input preparation (Jolliffe, 2011). It reduces the

dimension of input variables in terms of new linear combinations called principal component (PC) with the objective of losing very little information throughout the process (Chatfield, 2018). Moreover, where inputs don't have enough higher dimensional space, PCA can successfully be applied for optimal selection of input data (Hotelling, 1933). The basic advantage of adopting this method is that it transforms input space into a set of variables that are uncorrelated with each other and minimizes the problem of multicollinearity. Moreover, if input variables followed the normal distribution, the orthogonally transformed variables are not only uncorrelated, but also independent (Chatfield, 2018).

Suppose that an event *X* consists of $p$ attributes as $X = (x_1, x_2, x_3, \dots x_p)$ for runoff-affecting factors to obtain new variables $\xi = (\xi_1, \xi_2, \xi_3, \dots \xi_p)$ that form a linear function of input space, but are uncorrelated with decrease variance from $PC_1$ to $PC_p$. The variance contribution by each PC is calculated as:

$$\xi_i = \alpha_{i1}x_1 + \alpha_{i2}x_2 + +\alpha_{ij}x_i + \cdots + \alpha_{ip}x_p \quad (1)$$

For self- orthogonal transformation, the following conditions are required:

$$\sum_{i=1}^{p} \alpha_{ij}\alpha_{ik} = 0 \quad j \neq k \quad (2)$$

$$\sum_{i=1}^{p} \alpha_{ij}\alpha_{ik} = 0 \quad j = k \quad (3)$$

The PCs depend upon the degree of importance, meaning that the first PC gives the largest variance which decreases subsequently. Based on the above facts, this investigation used PCA as a dimension-reduction technique to reduce the number of inputs of the model considering different rain gauge stations.

***Extreme Gradient Boosting Machine (XGBoost)***

XGBoost is a scalable tree-boosting approach proposed by Chen and Guestrin (2016). The basic working framework XGBoost is the gradient-boosting algorithm. Boosting approach fits the base model sequentially through coalescing a series of weak learners to strong learners in the form of regression trees. Furthermore, the additive nature of base learners helps minimize the loss function till its reduction becomes limited or terminates. The rapid and scalable nature of learning makes XGBoost a superior model in the field of machine learning. The basic work frame of XGBoost is presented as follows:

Suppose a given dataset $\mathrm{D}$ having n samples with m features as $\mathrm{D} = \{x_i, y_i\}$, where *x* and *y* are predictor and predictand, respectively. XGBoost estimates the approximate function $F_k(x)$ through $K$ additive functions $f_K(x)$ as $F_K(x) = \sum_{K=1}^{K} f_k(x)$ through base learner. The $f_K(x)$ decision-tree learning starts from top to down as a decision tree, which is calculated as $\omega_{q(x)}, \mathrm{q} \in \{1,2, \dots, T\}$, where $q$ and $T$ denote the decision rule and leaves of the tree, whereas ω is the vector which represents sample initial weight at each node of the leaves (i.e., the score at each leaf). The objective function $(\ell_k)$ is calculated through the loss function by adding a regularization term as in Equation 4.

$$\ell_k() = \sum_{i=1}^{n} \psi\{y_i, \mathrm{F}(x_i)\} + \sum_{K=1}^{K} \Omega(f_k) \quad (4)$$

$\mathrm{F_K}(x_i)$ shows as $i^{th}$ sample at the *K-th* boost and $\Omega(f) = \gamma T + 0.5\lambda \parallel w \parallel^2$. γ and λ denote the complexity parameters and fixed coefficient, whereas $\parallel w \parallel^2$ is *L2* leaf weight. The term $\psi(*)$ is a specific loss function to minimize the difference between predicted and target values. The regularization term $(\Omega(*))$ supports in minimizing the model complexity. The basic difference between XGBoost and gradient boost is the regularization term as shown in Equation (4). If the regularization term is removed from the objective function, it acts as a simple gradient boosting loss function. The primary goal of the XGBoost is to search best suited $f_K$ to minimize the objective function. This minimization of the objective function is achieved through the Taylor expansion. The XGBoost algorithm uses greedy search algorithm to search optimal tree structure (Chen and Guestrin, 2016). This algorithm first decides the splitting point based on the percentile of feature distribution and then maps the variables into the buckets through the splitting process. Henceforth, first- and second-order gradient statistics are calculated for the estimation of the loss function, which results in the best candidate points for the assigned proposal. Further details are available in Chen and Guestrin (2016), Chen et al. (2016) and Nielsen (2016).

***Emotional Neural Network (EmNN)***

Emotion is a complex term and has no single universally accepted definition. Discussion about the necessity of emotions in artificial intelligence was initiated by Fellous (1999). In nature, emotions arise spontaneously rather than through conscious efforts. This concept is captured and utilized in neuro-modulation or emotional neural network (EmNN) models. In hydrological applications, any response simulation models can be performed by incorporating these emotions to improve the learning capability of artificial intelligence. This ensembling emotion concept differentiates EmNN model from other conventional ANN models or hybrid models. Figure 1 represents the single cell of the $i^{th}$ neuron and its flow of information. In EmNN model, the network is modulated through artificially emitted hormones which act as emotion bias in terms of total potential ($H_h$) to each neuron in the network. During operation, along with the changes in each neuron, the total potential parameters are automatically adjusted by inputs and outputs of the network (Roshni et al., 2020).

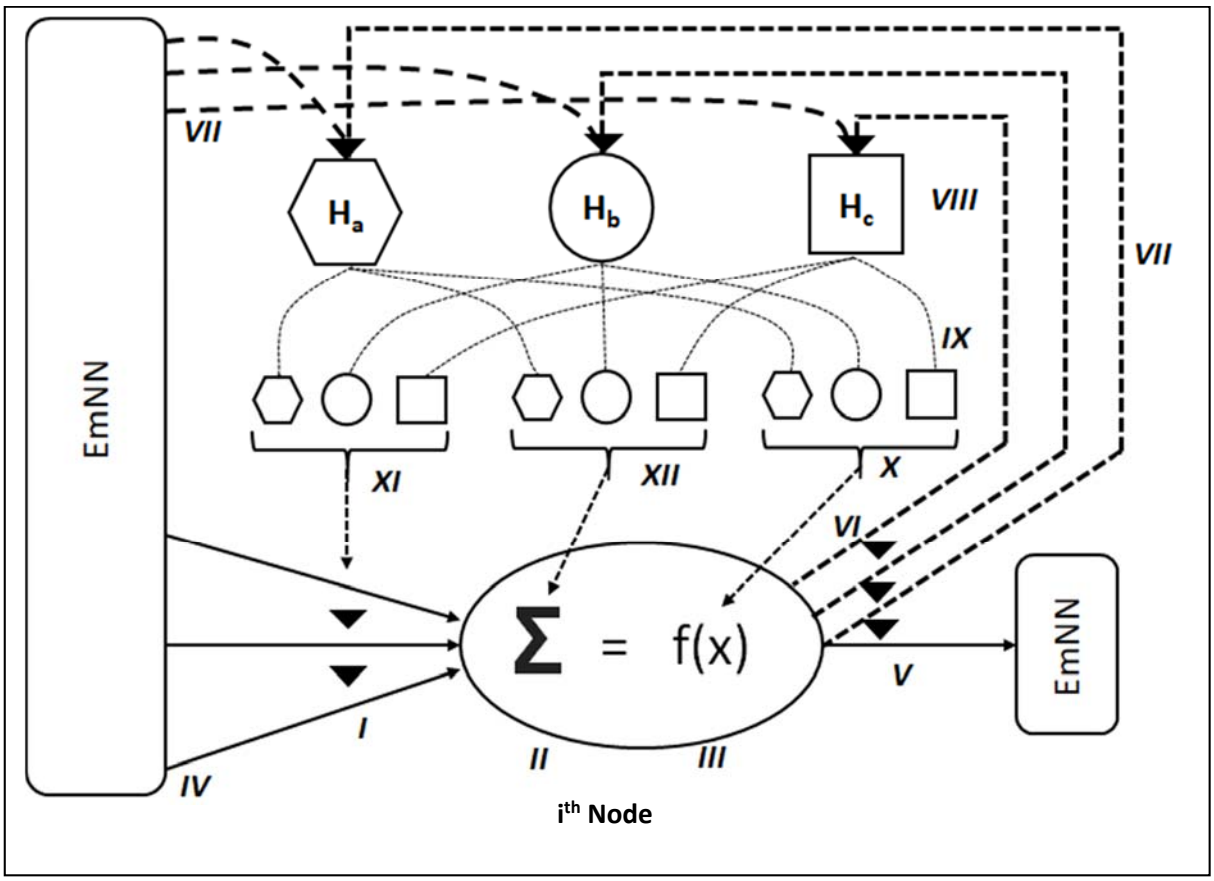


**Figure (1): A systematic flow diagram of EmNN for a single unit (Nourani, 2017)**

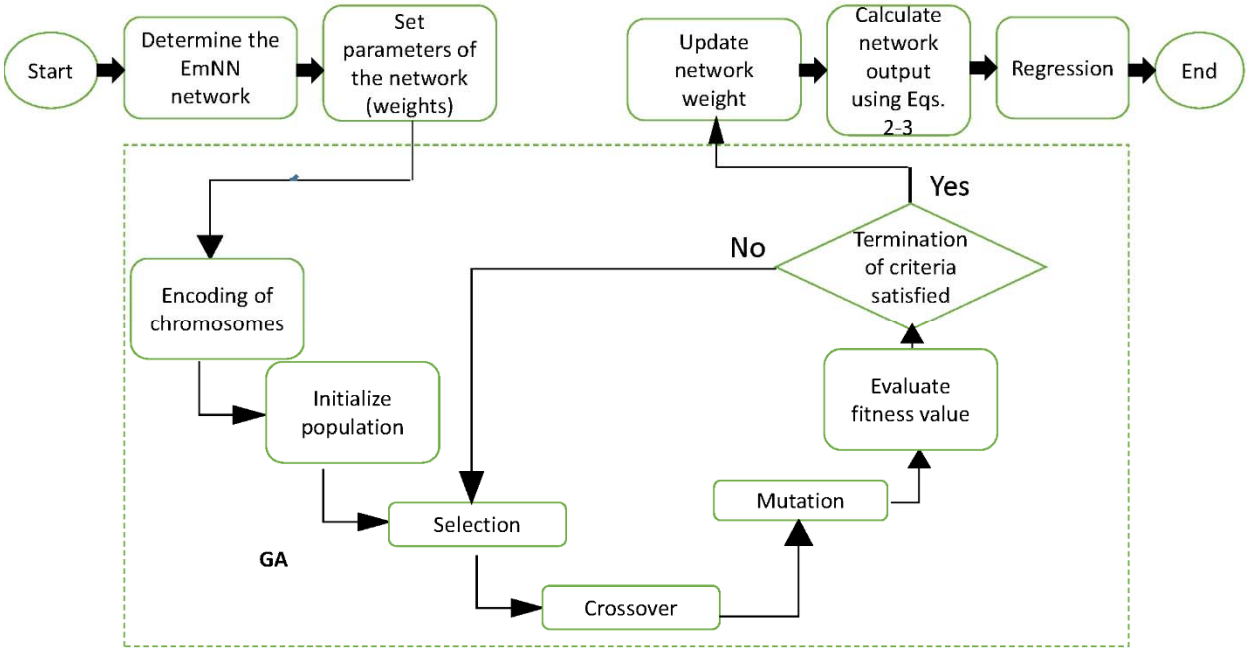


**Figure (2): Flowchart of emotional artificial neural network with GA**

The schematic work flow of an inner neuron from EmNN-GA is shown in Fig. 2. In EmNN model, there is reversible flow of information from inputs to outputs and *vice versa*. The cells of the EmNN (I-III) send information to EmNN (IV) and produce hormones (VIII). These hormones add the hormones output of the EmNN (VII) to one-time step of all cells. The total hormone value as total potential of the EmNN is calculated using Equation 5.

$$H_h = \sum_i H_{i,h}; \qquad H = (a, b, c) \tag{5}$$

The hormones in the system (H) differed with inputs and target data samples and hence are considered as dynamic coefficients. In the training phase, the hormone signals act on all units of the network (i.e., weights (I), net function (II) and activation function (III)). The net response of i$^{th}$ neuron ($Y_i$) is the total of all the weighted hormonal functions on activation functions. The output of the i$^{th}$ neuron with three hormonal glands of $H_a$, $H_b$ and $H_c$ is computed as shown in Eqn. (5) (Nourani, 2017).

$$Y_i = \underbrace{\left(\lambda_i + \sum_h \sigma_{i,h} H_h\right)}_{1} \times f\Big(\sum_j \underbrace{\left(\beta_i + \sum_h \zeta_{ih} H_h\right)}_{2} \times \underbrace{\left(\theta_{i,j} + \sum_h \phi_{i,j,k} H_h\right)}_{3} \mathrm{X}_{ij}\Big) + \underbrace{\left(a_i + \sum_h x_{i,h} H_h\right)}_{4} + \underbrace{\left(\delta_i + \sum_h \rho_{i,h} H_h\right)}_{5} \tag{6}$$

The weight applied to the activation function ($f$ ()) having both statistic neural weight of $\lambda_i$ and hormonal weight of $\sum_h \sigma_{i,h} H_h$ is denoted by term (1), whereas term (2) stands for applied net weight. Term (3) denotes the assigned weight to the input value of $\mathrm{X}_{i,j}$ coming from $j^{th}$ neuron of previous layer and term (4) indicates the bias of the net function, again containing neural and hormonal weights of $\alpha_i$ and $\sum_h \chi_{i,h} H_h$. The last term (5) denotes the contribution to the activation function, where neural and hormonal weights contribute as $\delta_i$ and $\sum_h \rho_{i,h} H_h$, respectively. The output of the cell is then calculated as a factor of the output of the response function as:

$$output_i = neurality_i \times Y_i \tag{7}$$

All the hormones of Eqn. (5) are produced by:

$$H_{i,h} = glandity_{i,h} \times Y_i \quad (8)$$

where $glandity_{i,h}$ is a parameter which generates hormonal level in the cell. Each hormone ($H_h$) is initialized with maximum value of input parameters of each sample. Then, the hormone value is upgraded in the training phase by considering the reliable agreement between observed and predicted values of target.

The degree to which a cell function is considered as a neural cell for a given hormone is determined by weights (V, VI). The weights of both neural and hormonal routes have an important role in the emotional neural network and are shown in Eqn. (5). In order to optimize the conventional weights, biases and hormonal weights and biases in each cell, a genetic algorithm (GA) is ensembled with the emotional neural network. By genetic algorithm (GA) optimization technique, all the flow component weights will be trained to get the best possible global values for calibration processes (Fig. 2). Studies proved that GA ensemble networks show a reliable agreement between observed and target values. Further details of EmNN and GA are furnished in Nourani (2017).

### *Group Method of Data Handling (GMDH)-type Neural Network*

GMDH type neural network is a machine learning technique developed by Ivakhnenko (1968) inspired from Kolmogorov-Gabor polynomial (KGP). The KGP approximates the random sequence of inputs and can be calculated by either system of Gaussian normal equation or adaptive methods. The connection between predictors and target variables can be estimated by Volterra functional series and the analogue is called as KGP. Equation (9) is known as Ivakhnenko polynomial.

$$y = a + \sum_{i=1}^{m} b_i \cdot x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} \cdot x_i \cdot x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} d_{ijk} \cdot x_i \cdot x_j \cdot x_k + \cdots \quad (9)$$

where m represents the number of predictors and constants (a, b, c and d) are the coefficients (or weights) of predictors in a polynomial. Here, $y$ is the target and $x_i$ and $x_j$ are the predictors to be regressed. A GMDH-type neural network derived its structure through investigating the relation between predictors and target considering individual effects. The target value of GMDH-type neural network is estimated using Equation 10. The GMDH algorithm consists of neurons in the layers and the number of which is defined by the number of input variables. Suppose $p$ is the number of input variables after considering all the pairwise combinations of variables, the number of neurons is equal to $h = \binom{p}{2}$ and the coefficient is calculated using Equation (10) for each neuron. Interested readers can find more details in (Anastasakis and Mort, 2001; Dag and Yozgatligil, 2016).

$$y = a + \sum_{i=1}^{m} b_i \cdot x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} \cdot x_i \cdot x_j \quad (10)$$

## MATERIAL AND METHODS

### *Study Area and Data Description*

The Jhelum river basin is located in Kashmir valley of India and is spread in an area of 17622 km$^2$. It lies between 33$^0$-35$^0$N and 73$^0$-76$^0$E. The River Jhelum flowing through the Jhelum river basin has 24 tributaries. Figure 3 shows the location map of the study area with the locations of three rain gauge stations and one discharge gauging station. The monthly rainfall data was collected from the Indian Metrological Department (IMD) Pune for the following rain gauge stations: Quazigund (Q), Srinagar (S) and Gulmarg (G) stations for the period Jan. 1987-Deec. 2016. The river discharge data was collected from the Irrigation and Flood Control Department of Srinagar.

The rainfall data collected from the IMD has missing values. In real-world problems, missing values in a dataset represent a common phenomenon. The structural pattern of missing values in the observation data is shown in Figure 4. To handle the missing values effectively, the well-known K-nearest neighbor method has been adopted (Buuren and Groothuis-Oudshoorn, 2010). It identifies 'K' closest observations based on the Euclidean distance. The weighted average (based on distance) of these 'K' observations is used to impute the values. This study considered K=12 to compute the missing values. More detail about K- nearest neighbor imputation can be found in Van Buuren (2018). Descriptive statistics of observation data after the treatment of missing values can be found in Table 1.

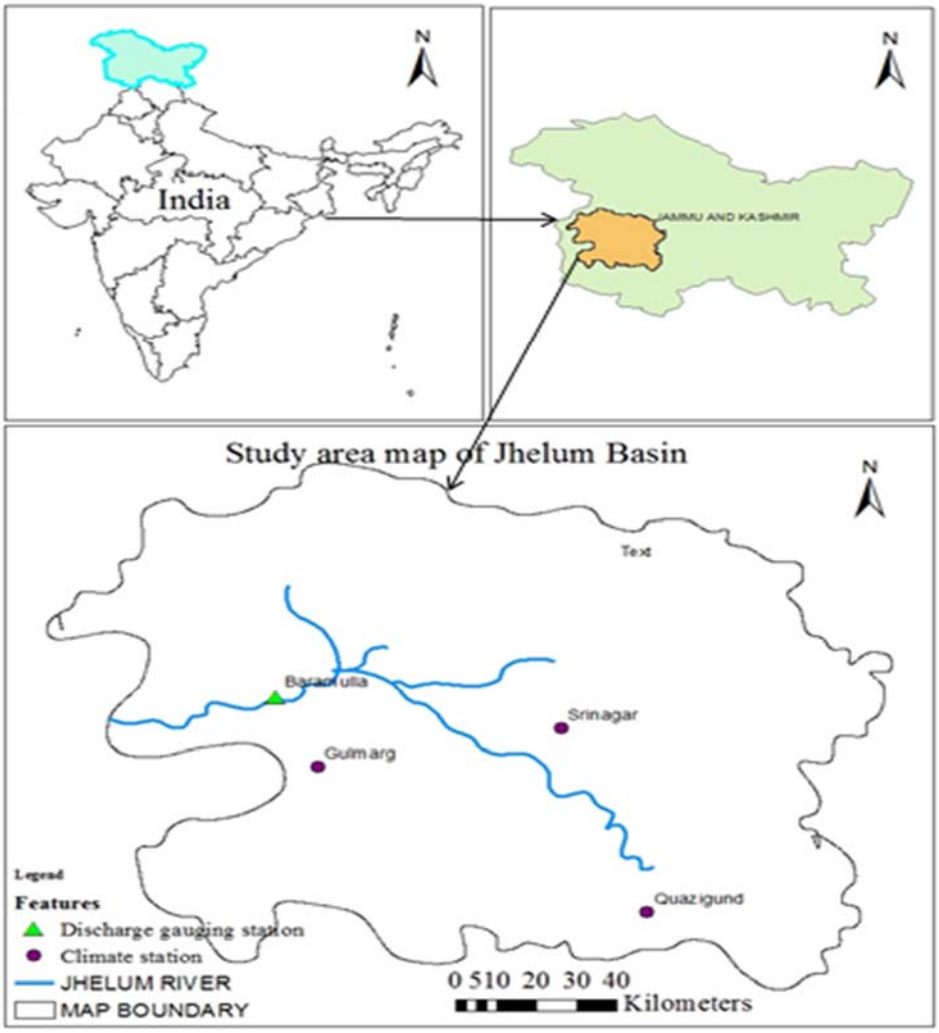


**Figure (3): Location of study area and Jhelum basin, India**

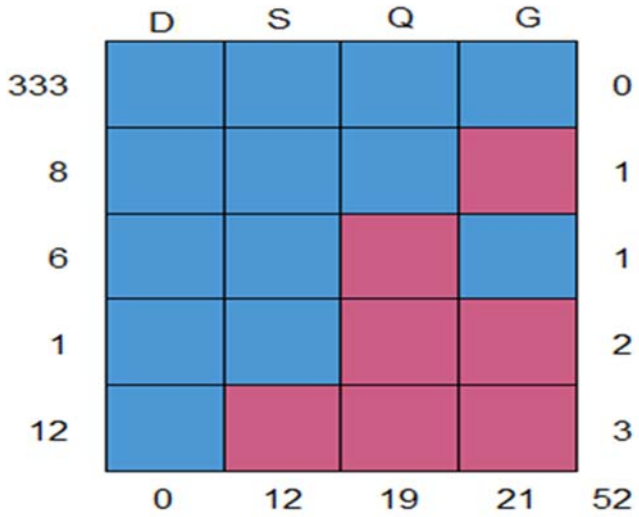


**Figure (4): Display of missing-data patterns at different rain gauge stations and discharge (S = Srinagar, Q = Quazigund, G= Gulmarg, D= Discharge)**

**Table 1. Statistical description of the input variables (rainfall and discharge) after K-nearest neighbor imputation**

| Statistics | Rainfall stations (mm) | | | Discharge (cumec) |
|---|---|---|---|---|
| | Srinagar (S) | Quazigund (Q) | Gulmarg (G) | Baramulla (B) |
| Minimum | 0.0 | 0.0 | 0.0 | 646.7 |
| Maximum | 294.60 | 39.58 | 693.40 | 42159.2 |
| Mean | 57.22 | 99.41 | 124.08 | 9054.0 |
| Standard deviation | 104.53 | 88.16 | 46.87 | 8209.4 |

**Development of Predictive Models**

In this study, three R-R models were developed for the Jhelum catchment, India by employing XGBoost, EmNN and GMDH modeling approaches. To predict the streamflow discharge, these three advanced machine-learning regression models were considered. The selection of appropriate inputs has always been a critical and important task in model development. This study adopted PCA to aggregate the different stations' rainfall data to equivalent precipitation (Abdi and Williams, 2010; Bartoletti et al., 2018; Li et al., 2000). The advantage of PCA is that it eliminates information redundancy through orthogonal transformation.

For a given set $R \in \mathbb{R}^{wxs}$of $w$ time- indexed measurement, the value of $S$ is represented as $S = n \times k + m$. The data generated from each individual rain gauge is indexed by k columns of shifted samples, followed by m columns of shifted output flow data.

$$R = \begin{bmatrix} \overbrace{\overbrace{r_1(1)r_1(2)\dots r_1(k)}^{\text{first rain gauge}}}_{1} & \overbrace{r_n(1)r_n(2)\dots r_n(k)}^{\text{n-th rain gauge}} & \overbrace{Q(2)\dots Q(m)}^{\text{past flow samples}} & \overbrace{Q_1}^{\text{Current flow}} \\ r_1(2)r_1(3)\dots r_1(k+1) & r_n(2)r_n(3)\dots r_n(k+1) & Q(3)\dots Q(m+1) & Q_2 \\ \vdots & \vdots & \vdots & \vdots \\ r_{n1} & r_{nn} & Q_n & Q_n \end{bmatrix} \quad (11)$$

After PCA transform, its principal components are given by:

$$z = R \cdot \mathrm{P} \quad (12)$$

where $P \in \mathbb{R}^{sxs}$represents the correlation matrix of eigenvectors. The columns were sorted by descending magnitude based on respective eigenvalues. The optimal number of principal components was selected based on the scree plot shown in (Fig. 5). The scree plot shows the input variance contributed by each component after the PCA. The fist component contributed 68 percent of the total variance, whereas the second component contributed approximately 23 percent. The remaining factors contribute a very marginal proportion of the variability and are likely unimportant.

The structure of the models drawn on an input matrix (x) defined by x= (PC1, PC2) is represented as the predictor variables, while the streamflow is represented as the target variable (y). For model development, appropriate input selection is one of the crucial and cumbersome processes, especially for time-series analysis. In any modeling process, the major task is to find the appropriate training and testing dataset. There is no thumb rule for data division and it depends upon the problem and the available dataset. 70% of the dataset (period: 1987-2007) was used for training the XGBoost and EmNN models and the remaining 30% of the dataset (2008-2016) was employed for testing the models. Prior to model development, all datasets were normalized using Equation (13). These models were trained based on the trial method and tuning parameters were selected

based on the lowest RMSE yield. Figure 6 represents the methodology adopted for R-R modeling with optimized parameters.

$$z_{\text{normalized}} = \frac{(z - z_{min})}{(z_{max} - Z_{min})} \quad (13)$$

where Z= data value, $Z_{min}$=minimum value of the whole dataset, $Z_{max}$= maximum value of the whole dataset and $Z_{normalized}$= normalized dataset.

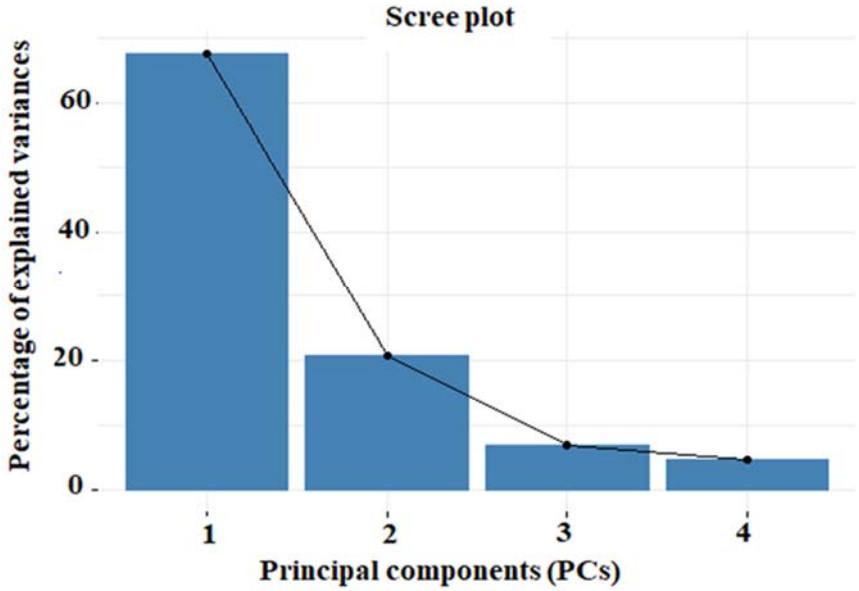


**Figure (5): Scree plot displaying the percentage of variance explained by individual principal components**

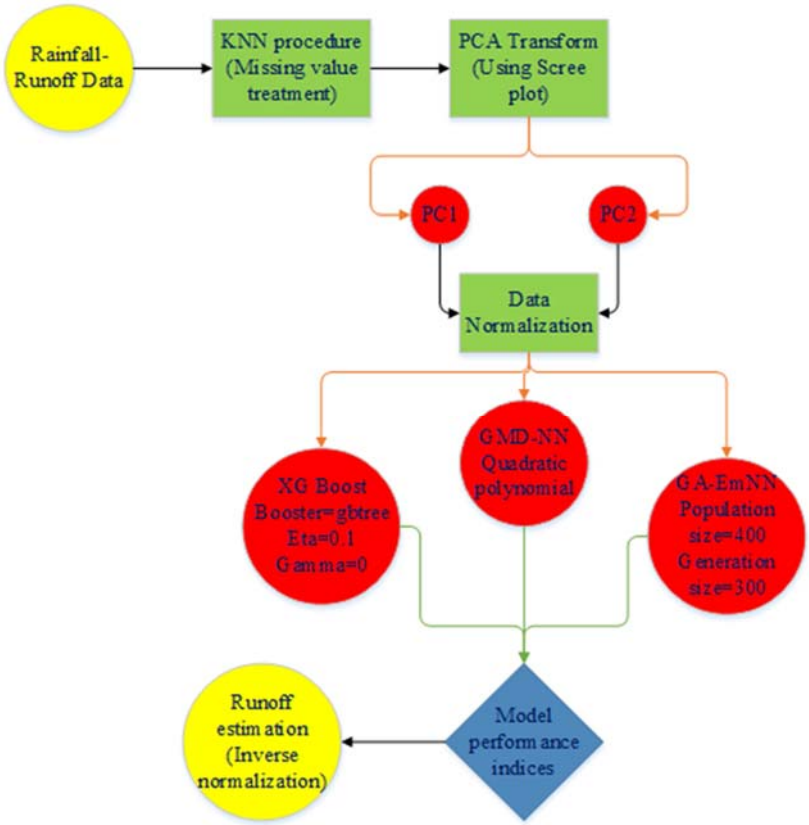


**Figure (6): Flowchart of the methodology of model development**

**Model Performance Assessment**

A number of values of control parameters were selected initially and thereafter varied in trials till the best fitness measures were obtained in the model development process. To investigate the performance of the proposed models, five statistical indicators, including Coefficient of Correlation (*r*), Root Mean Square Error (RMSE), relative Nash–Sutcliffe Efficiency coefficient (r NSE), Percent Bias (*PBIAS* %) and Kling-Gupta Efficiency (KGE), were used. All statistical fitness parameters do not indicate one-to-one relationship with each other, but rather show the different information based on certain correlation and variance. These criteria are basically classified into two classes: (Type 1 and Type 2). Type 1 exhibits the closeness of the forecast value to the target value, while type 2 shows the closeness of the mean of forecast value to the mean of target value (Papacharalampous et al., 2019). Furthermore, these metrics (*r*, r NSE, PBIAS %, KGE) are dimensionless, while *RMSE* is expressed in the same unit as the data.

r, RMSE, PBIAS%, rNSE and KGE are calculated using the following formulae:

$$r = \left( \frac{\sum_{i-1}^{l} \left(D_{E_S} - D_{\overline{E}_t}\right)\left(D_{o_i} - D_{\overline{o_i}}\right)}{\sqrt{\sum_{i=1}^{l} \left(D_{E_i} - D_{\overline{E}_i}\right)^2 \sum_{i=1}^{n} \left(D_0 - D_{\overline{o_i}}\right)^2}} \right) \quad (14)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{l} \left(D_{E_i} - D_{0_i}\right)^2}{l}} \quad (15)$$

$$PBIAS\% = \left(\frac{\sum(D_{oi} - D_{Ei})}{\sum D_{oi}}\right) \times 100 \quad (16)$$

$$rNSE = \left( 1 - \frac{\sum_{i=1}^{l} \left(\frac{D_{Ei} - D_{o_i}}{D_{o_i}}\right)^2}{\sum_{i=1}^{l} \left(\frac{D_{o_i} - D_{\bar{o}_i}}{D_{\overline{o_i}}}\right)^2} \right) \quad (17)$$

$$KGE = 1 - \sqrt{(s[1] * (r-1)^2 + (s[2] * (\alpha - 1))^2 + (s[3] * (\beta - 1))^2)} \quad (18)$$

*r* = Pearson product moment correlation coefficient.

$\beta = \frac{\mu_E}{\mu 0}$

$\alpha = \frac{\sigma_E}{\sigma 0}$

*s* =the scaling factors to be used for re-scaling the criteria space before computing the Euclidean distance.

$D_{E_i}$ is the $i^{th}$ estimated monthly streamflow using the models; $D_{o_i}$ is the $i^{th}$ observed monthly streamflow; $D_{\overline{E_i}}$ is the average estimated monthly streamflow; $D_{\overline{o_i}}$ is the average observed monthly streamflow and $l$ is the number of observations.

## RESULTS AND DISCUSSION

This section discusses the results of the proposed approach with respect to the following points; firstly, to select the optimal inputs for the models by using principal component analysis considering different rain gauge stations and discharge in a watershed for R-R modeling. It also shows how the PCA can be used to derive aggregate inputs (in terms of PCs) from various rain gauge stations situated and discharge value in the catchment for the modeling process. In addition, prior to model development, missing values are imputed using nearest neighbor method, which is an effective method for data imputation (Buuren and Groothuis-Oudshoorn, 2010). Subsequently, the proposed GA-EmNN method was tested for R-R modeling. Lastly, the performances of GA-EmNN, XGBoost and GMDH-neural network models were compared to find their adaptability.

To accomplish the above stated objective, PCA has been performed on the whole datasets for the selection of appropriate inputs for the modeling process. This technique removes the redundancy in input variables by determining the variations in input variables and how these variations are linked with each other. The resultant decomposed PCs (i.e., covariance matrix) and their variance explained by each component for rainfall runoff data are shown in Figure 5. Two PCs (PC1 and PC2) were selected as final inputs for the modeling process, which explained nearly 91% of the cumulative variance in the rainfall and discharge data. Bartoletti et al. (2018) have also emphasized the use of PCA over the GIS platform to calculate synthetic equivalent precipitation. In GIS platform, the time delay properties of different rain gauges, which are not taken into account, result in the loss of spatial characteristic of rainfall and assume the same rainfall over the catchment. This may be the approximately wrong assumption in the

case of spotty and short-duration rainfall. On the other side, PCA provides flexibility by selecting the number of PCs based on scree plots preserving space and time characteristics for individual rain gauge stations for the selected basin. In addition, methods like PCA also solve the problem of multicollinearity.

Coming to the data-driven models, two PCs (PC1 and PC2) were selected as inputs and discharge as output to train the model. The analysis of the results considering different statistical indices is presented in Table 2, which shows that all the models performed significantly well during the training phase. Based on the KGE and RMSE values, it was found that GMDH-NN (*KGE* = 0.999, *RMSE* =343.63) model had the best performance, followed by GA-EmNN (*KGE* = 0.998, *RMSE* =345.48) and XGBoost (*KGE* = 0.992, *RMSE* =784.25). Furthermore, in terms of average tendency (PBIAS %) of the forecast values, all the models are found satisfactory. The statistical indicator rNSE values show that all the models are well trained during the training phase.

**Table 2. The models' performance based on fitness parameters**

| | Training | | | Testing | | |
|---|---|---|---|---|---|---|
| | **XGBoost** | **GMDH-NN** | **GA-ENN** | **XGBoost** | **GMDH-NN** | **GA-ENN** |
| r | 0.996 | 0.999 | 0.999 | 0.973 | 0.997 | 0.997 |
| KGE | 0.992 | 0.999 | 0.998 | 0.971 | 0.990 | 0.994 |
| PBIAS % | 0 | 0 | -0.2 | 0.5 | 0.9 | 0.5 |
| rNSE | 0.977 | 0.993 | 0.993 | 0.935 | 0.988 | 0.988 |
| RMSE | 784.246 | 343.626 | 345.483 | 790.892 | 438.387 | 428.352 |

During the testing phase, GA-EmNN (*r* = 0.997, *KGE* = 0.994, *rNSE* = 0. 998) model had the best performance, followed by GMDH-NN (*r* = 0.997, *KGE* = 0.990, *rNSE* = 0. 998) and XGBoost (*r* = 0.973, *KGE* = 0.971, *rNSE* = 0.935). The lower the error, the better the model. Based on this fact, GA-EmNN is the best among the tested models for R-R modeling. The scattered plot (Fig. 7) represents the fact that during the training and testing periods, the GA-EmNN model is well capable of reproducing the low flows and high flows of discharge, as all the points are well fitted around the lines, except for few high points during testing. Figure 8 confirms the fact that GA-EmNN model is capable of handling the oscillations in both the rising and falling peaks of streamflow during the testing phase.

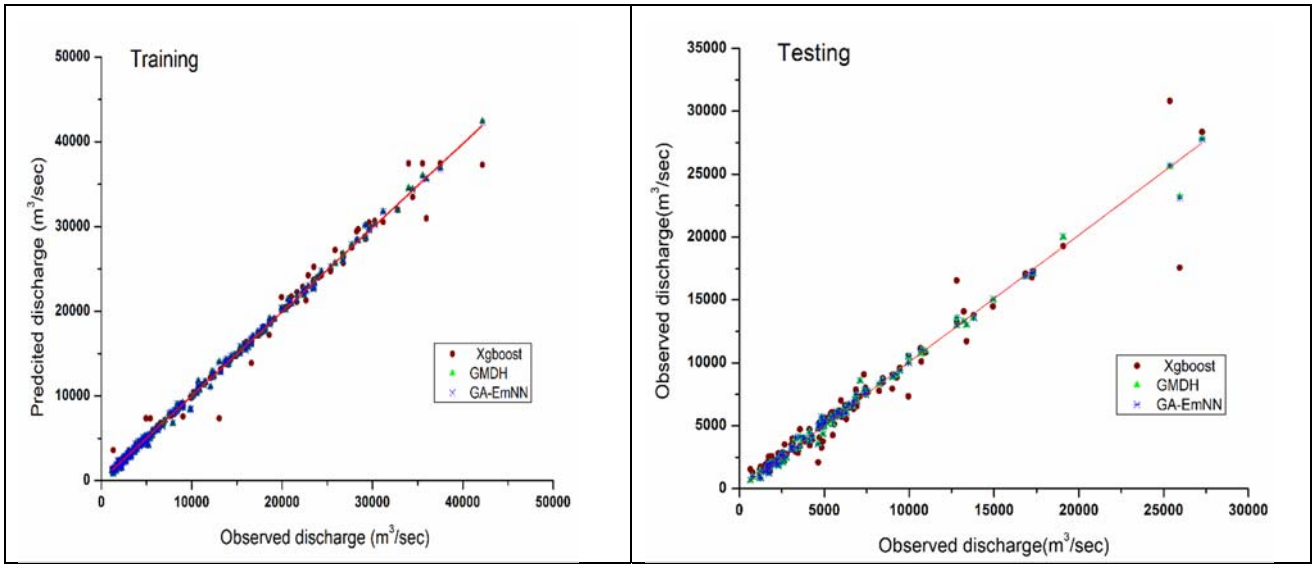


**Figure (7): Scatter plot displaying the performance of the models for training and testing**

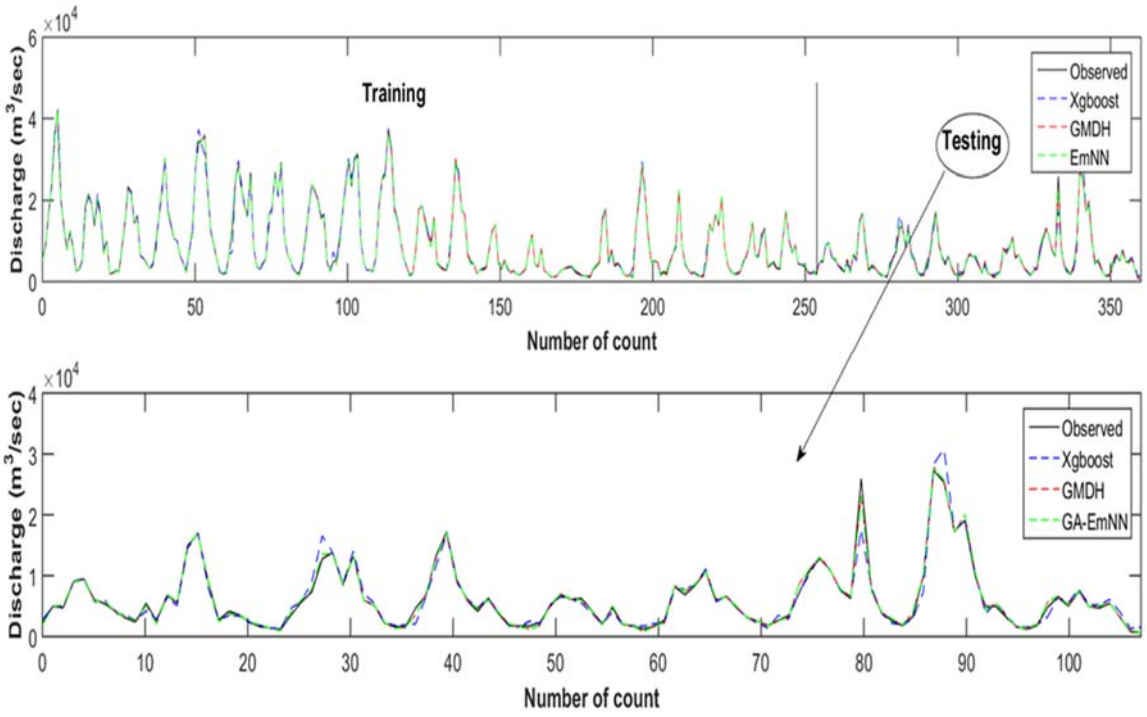


**Figure (8): Plot showing the prediction performance for streamflow during training and testing**

Figure 9 finally shows the violin plot of the models' performance in terms of error (observed-predicted) produced during the testing phase. The violin plot is an effective tool for visualizing the distribution of quantitative data at several levels. Moreover, through this plot we can visualize datasets of small sizes, density distribution, mean and median of quantitative data (Hintze and Nelson, 1998). From Figure 9, it is evident that the GA-EmNN model has a uniform distribution with minimum lower error outlier. The model performance in terms of error shows that both the models GMDH-NN and GA-EmNN have similar persistence in terms of error, but more deviation in terms of RMSE. Therefore, based on the above evidence, it can be concluded that the GA-EmNN model is the best model among the selected models for R-R modeling. The research findings conclude that the GA-EmNN model can be used as a better option for hydrological modeling and its allied fields.

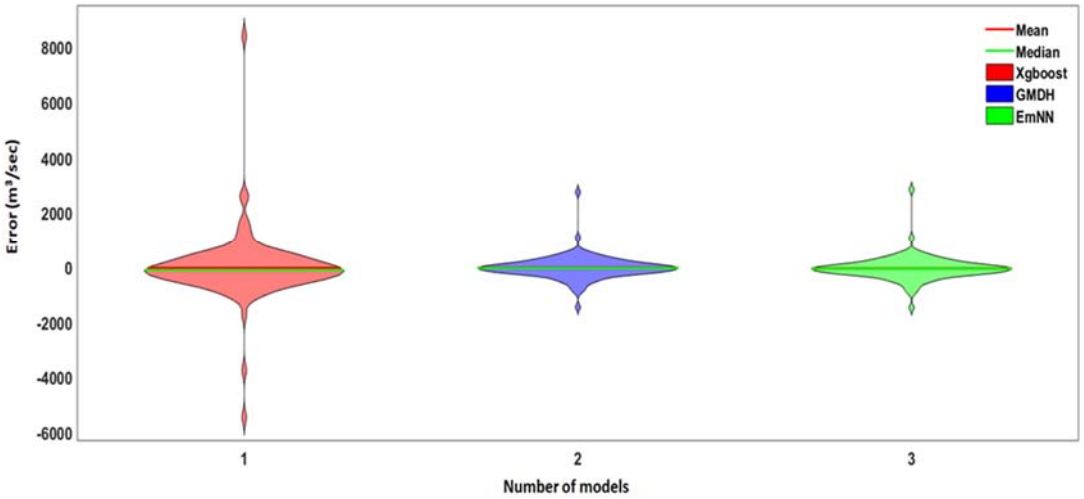


**Figure (9): Violin plot displaying the errror (observed -predicted) by the models during the testing phase**

**Concluding Remarks**

In data-driven-based R-R modeling, emphasis is mainly on the flow routing to get higher predictive performance than pre-processing of the input data. This study considers both aspects, pre-processing of raw data (treatment of missing values and inputs selection) as well as model development. This article approached a two-step procedure to select significant inputs *via* PCA and the selected inputs were used to build the different machine-learning models. The example discussed in this study shows that this approach is favorable over conventional methods where synthetical equivalent precipitation is crucial. Moreover, such method can effectively reduce the computational load through simplifying the model configuration through dimension reduction. The findings of this study recommend the GA-EmNN model as a potential alternative to assist hydrological engineering in the study of streamflow. Thus, PCA- ML (i.e., GA-EmNN, GMDH-NN and XGBoost) combination can be an efficient data-driven approach to R-R modeling. The future direction of the current work includes application of such techniques in solving other hydrological engineering problems and utilizing other state-of-the-art methods used for enhancing the predictive performance.

**Compliance with Ethical Standards**

Conflict of interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

**REFERENCES**


Abdi, H., and Williams, L.J. (2010). "Principal component analysis". Wiley Interdisciplinary Reviews: Computational Statistics, 2, 433-459.

Abrahart, R.J., Heppenstall, A.J., and See, L.M. (2007). "Timing error correction procedure applied to neural network rainfall—runoff modeling". Hydrological Sciences Journal, 52, 414-431.

Anastasakis, L., and Mort, N. (2001). "The development of self-organization techniques in modeling: A review of the group method of data handling (GMDH)". Research Report No. 813-University of Sheffield, Department of Automatic Control and Systems Engineering, 1-39.

Avdullahi, S., Fejza, I., and Tmava, A. (2012). "Protecting water resources from pollution in the Lake Batllava". Jordan Journal of Civil Engineering, 6 (4), 464-475.

Bartoletti, N., Casagli, F., Marsili-Libelli, S., Nardi, A., and Palandri, L. (2018). "Data-driven rainfall/runoff modeling based on a neuro-fuzzy inference system". Environmental Modeling and Software, 106, 35-47.

Bobba, A.G., Singh, V.P., and Bengtsson, L. (2000). "Application of environmental models to different hydrological systems". Ecological Modeling, 125, 15-49.

Bourdin, D.R., Fleming, S.W., and Stull, R.B. (2012). "Streamflow modeling: A primer on applications, approaches and challenges". Atmosphere-Ocean, 50, 507-536.

Bowden, G.J., Dandy, G.C., and Maier, H.R. (2005). "Input determination for neural network models in water resources applications: Part 1- Background and methodology". Journal of Hydrology, 301, 75-92.

Brunner, G.W. (2010). "HEC-RAS river analysis system: Hydraulic reference manual". Ver. 4.1, US Army Corps of Engineers, Institute for Water Resources, Hydrologic Engineering Center. 1-417.

Buuren, S.V., and Groothuis-Oudshoorn, K. (2010). "Mice: Multivariate imputation by chained equations". The R Journal of Statistical Software, 1-68.

Chatfield, C. (2018). "Introduction to multivariate analysis". Routledge.

Chau, K.-W. (2017). "Use of meta-heuristic techniques in rainfall-runoff modeling". Multidisciplinary Digital Publishing Institute.

Chen, T., and Guestrin, C. (2016). "XGBoost: A scalable tree-boosting system". In: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, ACM, 785-794.

Chen, T., He, T., and Benesty, M. (2016). "XGboost: Extreme gradient boosting". R-package, version 0.4-4.

Chow, V.T. (1964). "Handbook of applied hydrology".

Dag, O., and Yozgatligil, C. (2016). "GMDH: An R-package for short-term forecasting *via* GMDH-type neural network algorithms". The R Journal of Statistical Software, 8, 379-386.

Douinot, A., Roux, H., and Dartus, D. (2017). "Modeling errors' calculation adapted to rainfall–runoff model user expectations and discharge data uncertainties". Environmental Modeling and Software, 90, 157-166.

Galelli, S., and Castelletti, A. (2013). "Assessing the predictive capability of randomized tree-based ensembles in streamflow modeling". Hydrology and Earth System Sciences, 17, 2669-2684.

Hintze, J.L., and Nelson, R.D. (1998). "Violin plots: A box plot-density trace synergism". The American Statistician, 52, 181-184.

Hotelling, H. (1933). "Analysis of complex statistical variables into principal components". Journal of Educational Psychology, 24, 417.

Hu, T., Wu, F., and Zhang, X. (2007). "Rainfall–runoff modeling using principal component analysis and neural network". Hydrology Research, 38, 235-248.

Ivakhnenko, A.G. (1968). "The group method of data handling: A rival of the method of stochastic approximation". Soviet Automatic Control, 13 (3), 43-55.

Jolliffe, I. (2011). "Principal component analysis". Springer.

Joshi, B.R., and Yadav, S.M. (2021). "Accounting for seasonal land-use trends in improving the predictability of irrigation needs in watersheds". Jordan Journal of Civil Engineering, 15 (2), 292-304.

Kalteh, A.M. (2016). "Improving forecasting accuracy of streamflow time series using least squares support vector machine coupled with data-preprocessing techniques". Water Resources Management, 30, 747-766.

Kokkonen, T.S., and Jakeman, A.J. (2001). "A comparison of metric and conceptual approaches in rainfall-runoff modeling and its implications". Water Resources Research, 37, 2345-2352.

Li, W., Yue, H.H., Valle-Cervantes, S., and Qin, S.J. (2000). "Recursive PCA for adaptive process monitoring". Journal of Process Control, 10, 471-486.

Li, X., Maier, H.R., and Zecchin, A.C. (2015). "Improved PMI-based input variable selection approach for artificial neural networks and other data-driven environmental and water resource models". Environmental Modeling and Software, 65, 15-29.

McIntyre, N.R., and Wheater, H.S. (2004). "A tool for risk-based management of surface water quality". Environmental Modeling and Software, 19, 1131-1140.

Mehr, A.D., and Kahya, E. (2017). "A Pareto-optimal moving average multigene genetic programming model for daily streamflow prediction". Journal of Hydrology, 549, 603-615.

Moazenzadeh, R., Mohammadi, B., Shamshirband, S., and Chau K.-W. (2018). "Coupling a firefly algorithm with support vector regression to predict evaporation in northern Iran". Engineering Applications of Computational Fluid Mechanics, 12, 584-597.

Moosavi, V., Talebi, A., and Hadian, M.R. (2017). "Development of a hybrid wavelet packet-group method of data handling (WPGMDH) model for runoff forecasting". Water Resources Management, 31, 43-59.

Mosavi, A., Ozturk, P., and Chau, K.-W. (2018). "Flood prediction using machine-learning models: Literature review". Water, 10, 1536.

Mulvaney, T.J. (1851). "On the use of self-registering rain and flood gauges in making observations of the relations of rainfall and flood discharges in a given catchment". Proceedings of the Institution of Civil Engineers of Ireland, 4, 19-31.

Nielsen, D. (2016). "Tree boosting with XGBoost: Why does XGBoost win every machine-learning competition?" Master Thesis, Norwegian University of Science and Technology (NTNU).

Nourani, V. (2017). "An emotional ANN (EANN) approach to modeling rainfall-runoff process". Journal of Hydrology, 544, 267-277.

Papacharalampous, G., Tyralis, H., and Koutsoyianni, D. (2019). "Comparison of stochastic and machine-learning methods for multi-step ahead forecasting of hydrological processes". Stochastic Environmental Research and Risk Assessment, 33, 481-514.

Remesan, R., Bray, M., and Mathew, J. (2018). "Application of PCA and clustering methods in input selection of hybrid runoff models". Journal of Environmental Informatics, 31, 137-152.

Roshni, T., Jha, M.K., and Drisya, J. (2020). "Neural network modeling for groundwater-level forecasting in coastal aquifers". Neural Computing and Applications, 32, 12737-12754.

Sharghi, E., Nourani, V., Molajou, A., and Najafi, H. (2019). "Conjunction of emotional ANN (EANN) and wavelet transform for rainfall-runoff modeling". Journal of Hydroinformatics, 21, 136-152.

Sharghi, E., Nourani, V., Najafi, H., and Molajou, A. (2018). “Emotional ANN (EANN) and wavelet-ANN (WANN) approaches for Markovian and seasonal-based modeling of rainfall-runoff process". Water Resources Management, 32, 3441-3456.

Shoaib, M., Shamseldin, A.Y., Khan, S., Khan, M.M., Khan, Z.M., and Melville, B.W. (2018). “A wavelet-based approach for combining the outputs of different rainfall–runoff models". Stochastic Environmental Research and Risk Assessment, 32, 155-168.

Solomatine, D.P., and Dulal, K.N. (2003). “Model trees as an alternative to neural networks in rainfall-runoff modeling". Hydrological Sciences Journal, 48, 399-411.

Taormina, R., Chau, K.-W., and Sivakumar, B. (2015). “Neural network river forecasting through baseflow separation and binary-coded swarm optimization". Journal of Hydrology, 529, 1788-1797.

Van Buuren, S. (2018). “Flexible imputation of missing data". CRC/Chapman and Hall, FL: Boca Raton. A Chapman and Hall Book.

Voinov, A., Fitz, C., Boumans, R., and Costanza, R. (2004). “Modular ecosystem modeling". Environmental Modeling and Software, 19, 285-304.

Wang, W.-C., Chau, K.-W., Qiu, L., and Chen, Y.-B. (2015). “Improving forecasting accuracy of medium- and long-term runoff using artificial neural network based on EEMD decomposition". Environmental Research, 139, 46-54.

Wang, Y., Guo, S., Chen, H., and Zhou, Y.(2014). “Comparative study of monthly inflow prediction methods for the Three Gorges Reservoir". Stochastic Environmental Research and Risk Assessment, 28, 555-570.

Wu, C., and Chau, K. (2011). “Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis". Journal of Hydrology, 399, 394-409.

Wu, W., Dandy, G.C., and Maier, H.R. (2014). “Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking-water quality modeling". Environmental Modeling and Software, 54,108-127.

Yaseen, Z.M., Sulaiman, S.O., Deo, R.C., and Chau, K.-W. (2018). “An enhanced extreme learning-machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction". Journal of Hydrology. 569, 387-408.