

Machine learning algorithms for streamflow forecasting of Lower Godavari Basin

Rishith Kumar Vogeti^a, Bhavesh Rahul Mishra^b and K. Srinivasa Raju^{a,*}

^a Department of Civil Engineering, BITS Pilani Hyderabad Campus, Hyderabad, India

^b Department of Electrical and Electronics Engineering, BITS Pilani Hyderabad Campus, Hyderabad, India

*Corresponding author. E-mail: ksraju@hyderabad.bits-pilani.ac.in

ABSTRACT

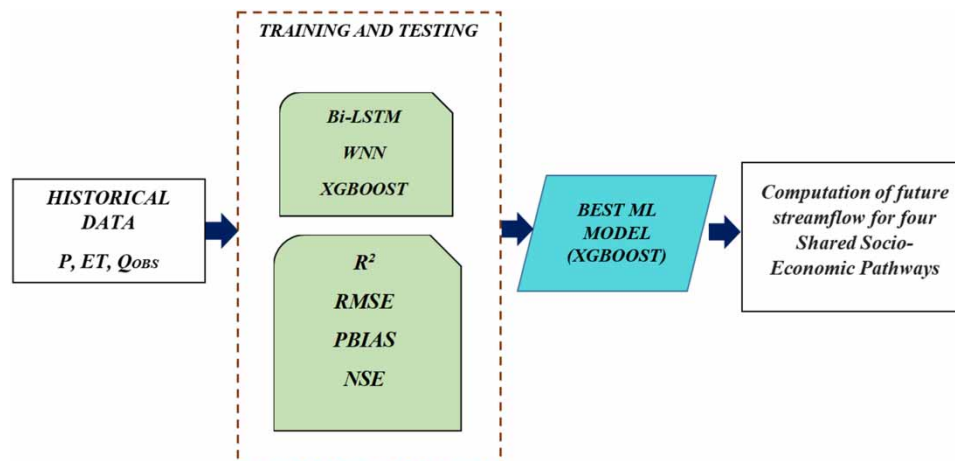
The present study applies three Machine Learning Algorithms, namely, Bi-directional Long Short-Term Memory (Bi-LSTM), Wavelet Neural Network (WNN), and eXtreme Gradient Boosting (XGBoost), to assess their suitability for streamflow projections of the Lower Godavari Basin. Historical data of 39 years of daily rainfall, evapotranspiration, and discharge were used, of which 80% applied for the model training and 20% for the validation. A Random Search method was used for hyperparameter tuning. XGBoost performed better than WNN, and Bi-LSTM with an R^2 , RMSE, NSE, and PBIAS of 0.88, 1.48, 0.86, and 29.3% during training, and 0.86, 1.63, 0.85, and 28.5%, during validation, indicating the model consistency. Therefore, it was further used for projecting streamflow from climate change perspective. Global Climate Model, Ec-Earth3 was employed in the present study. Four Shared Socioeconomic Pathways (SSPs) were considered and downscaled using Empirical Quantile Mapping. Eight decadal streamflow projections were computed – D1 to D8 (2021–2030 to 2091–2099) – exhibiting significant changes within the warm-up period. They were compared with three historical time periods of H1 (1982–1994), H2 (1995–2007), and H3 (2008–2020). The highest daily streamflow projections were observed in D1, D3, D4, D5, and D8 in SSP245 as per XGBoost analysis.

Key words: Bi-LSTM, hyperparameter tuning, SSP, streamflow, WNN, XGBoost

HIGHLIGHTS

- extreme Gradient Boosting (XGBoost) model displays an exceptional performance with $R^2 = 0.88$, $RMSE = 1.48$, $NSE = 0.87$, and $PBIAS = 29.3\%$ in training. These values in validation are 0.86, 1.48, 0.85, and 28.5%. XGBoost has the edge over the four variants of WNN and a Bi-LSTM model.
- The highest daily streamflow projections were observed in D₁, D₃, D₄, D₅, and D₈ in SSP245 as per XGBoost analysis.

GRAPHICAL ABSTRACT



This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY 4.0), which permits copying, adaptation and redistribution, provided the original work is properly cited (<http://creativecommons.org/licenses/by/4.0/>).

INTRODUCTION

Streamflow projections will assist decision-makers in perceiving a basin's water status, enabling them to make effective basin planning and management initiatives. In this regard, hydrological models play a crucial role, necessitating the identification of a reliable model. Another dimension is climate change, where the patterns of rainfall and temperatures are expected to vary in the future time horizons with increasing greenhouse gas emissions. It significantly impacts water resources (Konapala *et al.* 2020). Thus, it is essential to analyze the streamflow changes at a smaller time scale that can be useful to the decision makers (Galavi *et al.* 2019).

Simulating a catchment response with a great extent of non-linearity using a hydrological model is challenging (Kabir *et al.* 2022). In this context, diverse process-based hydrological models (Horton *et al.* 2022) were examined to achieve improved streamflow simulations. Many approximations in conceptualizing the model and high data requirements are the challenges, despite showing better interpretability of the underlying mechanisms (Moges *et al.* 2020). In this context, machine learning (ML) models were well suited for establishing intricate non-linear relationships within a system. Complementarily, computational efficiency is the added advantage. Unlike process-based hydrological models, ML models do not entail in-depth knowledge of the basin (Kim *et al.* 2021; Rozos *et al.* 2022). Many studies reported that Adaptive Neuro Fuzzy Inference Systems and Artificial Neural Networks (ANNs) were well suited for simulating streamflow (Galavi & Lee 2012; Galavi *et al.* 2013). However, one of the challenges that may encounter is overfitting. They frequently fail to capture long-term dependency from time series data.

Recurrent Neural Networks (RNNs) (Zhang *et al.* 2021) were used to address these shortcomings, demonstrating better streamflow accuracy with minimal training time than ANN. However, these models have a vanishing gradient problem. Accordingly, Long Short-Term Memory (LSTM) and its variants were investigated and found reliable in streamflow predictions (Mirzaei *et al.* 2021). Bi *et al.* (2020) compared a deep neural network, CAGANet, with the LSTM, Support Vector Machine (SVM), and the attention-mechanism-based LSTM incorporating data augmentation in forecasting daily streamflow in Qingxi river basin, Southwest China. They have found that CAGANet has shown exceptional capability with an NSE of 0.98, followed by LSTM variants with an NSE of 0.85–0.92. However, these models require high computational efforts. This situation necessitates simple hybrid models such as WNN to achieve satisfactory predictions.

Kasiviswanathan *et al.* (2016) applied ANN and WNN combined with Block Bootstrap (BB) sampling to forecast the Bow River Basin's streamflow, Alberta, Canada. They employed a percent volumetric error indicator. Comparative analysis revealed the dominance of WNN-BB over ANN-BB. Honorato *et al.* (2018) compared ANN and WNN over the Sobradinho dam in Northeast Brazil, in which the approximation of the wavelet decompositions was fed to ANN as inputs. The results confirmed that WNN proved superior to ANN in the lead time intervals of 1, 3, and 6 months, respectively, but had deteriorated in 12-month lead times. They opined that a high-frequency component improved model accuracy in shorter forecast horizons and vice versa for longer forecasts. Sharghi *et al.* (2019) developed an emotional ANN (EANN), WNN, and Wavelet-EANN for streamflow modeling of the Trinity and West Nishnabotna rivers. Wavelet-EANN is found promising for simulating streamflow on daily and monthly temporal scales. Criswell & Lin (2021) employed ANN, WNN, and Inverse Discrete Wavelet Transform (IDWT) to forecast river flow in Tittabawassee, USA. They revealed that WNN outperformed ANN and IDWT because of incorporating higher-level wavelet decomposition. They also opined that model accuracy could be improved using appropriate mother wavelets.

Furthermore, studies have been conducted to identify a model structure that works effectively even without pre-processing, which was possible in XGBoost. The superiority of this model and its variants has been demonstrated in streamflow simulation studies conducted across various catchments. Sahour *et al.* (2021) applied XGBoost and Random Forest (RF) models for simulating streamflow. The performance of XGBoost was slightly better than RF. Ni *et al.* (2020) employed the Gaussian mixture model (GMM)-XGBoost, SVM, and XGBoost for monthly streamflow forecasting of two stations in the Yangtze River Basin. GMM-XGBoost displayed a performance superior to SVM and XGBoost in streamflow simulation. Bai *et al.* (2021) developed a hybrid model based on XGBoost, Bayesian optimization algorithm, and Gaussian process regression for forecasting streamflow. They applied the developed model and eight streamflow prediction methods to the Yangtze River Basin, China. The hybrid XGBoost model showed satisfactory performance compared with other models with an R^2 of 0.97, MAPE of 8.09%, and RMSE of 1.85.

Rabeanahary *et al.* (2021) applied ANN and Soil Water Assessment Tool (SWAT) in climate change perspective for streamflow forecasting over Mangoky River, Africa. They employed Shared Socioeconomic Pathways (SSPs), namely, SSP370 and SSP585. It was concluded that there would be a decrease in annual streamflow in the basin. Song *et al.* (2022) applied SWAT and LSTM to forecast streamflow using SSP245 and SSP585 scenarios over the Yeongsan Basin, South Korea. SWAT projected an increase in the future streamflow by 17.7%. Contrarily, LSTM predicted a decrease in streamflow by 13.6%. In summary, precise information about streamflow changes would be handy for policy-makers in decision-making.

It is observed from literature survey that three ML algorithms, LSTM, WNN, and XGBoost, were promising, as evident from different case studies and has limited applications. In addition, these models were never compared simultaneously for Indian conditions but were explored individually. Accordingly, the objectives framed are as follows:

- (1) To investigate the prediction performance of Bi-LSTM, WNN, and XGBoost from a historical perspective and identify the suitable one.
- (2) To project streamflow for eight decadal time horizons, D₁ (2021–2030), D₂ (2031–2040), D₃ (2041–2050), D₄ (2051–2060), D₅ (2061–2070), D₆ (2071–2080), D₇ (2081–2090), and D₈ (2091–2099) using suitable ML model obtained from (1).

CASE STUDY AND DATA COLLECTION

The Lower Godavari Basin (LGB) extends from the latitude of 16°19'–22°34'N and the longitude of 73°24'–83°4'E. The basin shares boundaries with Andhra Pradesh, Chhattisgarh, Odisha, Maharashtra, and Telangana. It has a basin length of 462 km (332 km from Perur to Polavaram) and a total catchment area of 39,180 km² (refer to Figure 1). The study area is in the lower reaches of the Godavari basin and is nearer to the Bay of Bengal. There is a considerable water requirement in the basin for various purposes like drinking, irrigation, industrial, stabilization of canals, inter-state dependencies, etc. (Polavaram Project Authority 2021). This requires efficient planning and management initiatives to ensure water security to satisfy the vast requirements of the basin. Accordingly, streamflow in the future is required to be estimated.

Rainfall, maximum and minimum temperatures for 52 grid locations of LGB for the period 1982–2020 were collected from India Meteorological Department. Daily discharge for Polavaram and Perur was obtained from Central Water Commission. Future rainfall, maximum and minimum temperatures for 2021–2099 were obtained using the global climate model, EC-Earth3, and downscaling method, Empirical Quantile Mapping for SSP126, SSP245, SSP370, and SSP585 (Mishra *et al.* 2020). Evapotranspiration was computed using the Penman–Monteith method (<https://www.fao.org/land-water/databases-and-software/eto-calculator/en/>). In this study, Ec-Earth3 was selected as it was found promising in the South Asian Region as evident from the previous research works (Mishra *et al.* 2020; Parding *et al.* 2020; Iqbal *et al.* 2021; Desmet & Ngo-Duc 2022) and also because of its high grid resolution of 0.7° (Liu *et al.* 2022). Average rainfall, evapotranspiration, and observed discharge were the inputs to Bi-LSTM, WNN, and XGBoost to simulate streamflow. Additionally, the future rainfall and evapotranspiration were averaged and fed into the best algorithm for projecting streamflows for SSP126, SSP245, SSP370, and SSP585.

Furthermore, decadal time horizons were chosen as they facilitate understanding the most significant changes within the warming range (Mahmood *et al.* 2021) that would be experienced in the basin. The historical streamflow is split into three-time horizons, H₁ (1982–1994), H₂ (1995–2007), and H₃ (2008–2020), for a comprehensive comparison of the hydrologic behaviour of the basin. Supplementary Material, Table S1 presents streamflow information for three historical time horizons.

DESCRIPTION OF METHODS

This section briefly explains the Bi-LSTM, WNN, and XGBoost methods.

Bi-LSTM (Bi-directional Long Short-Term Memory)

Bi-LSTM (Li *et al.* 2021), an improved version of LSTM, is a series processing model comprising two LSTMs – one that receives input forward (past to future) and the another backward (future to past) – thus providing computational flexibility. This Bi-LSTM is a cyclic RNN variant that adds a gating unit to address the issue of a long-term dependence on long-sequence data. It restrains the vanishing gradient and helps retain each cyclic unit's

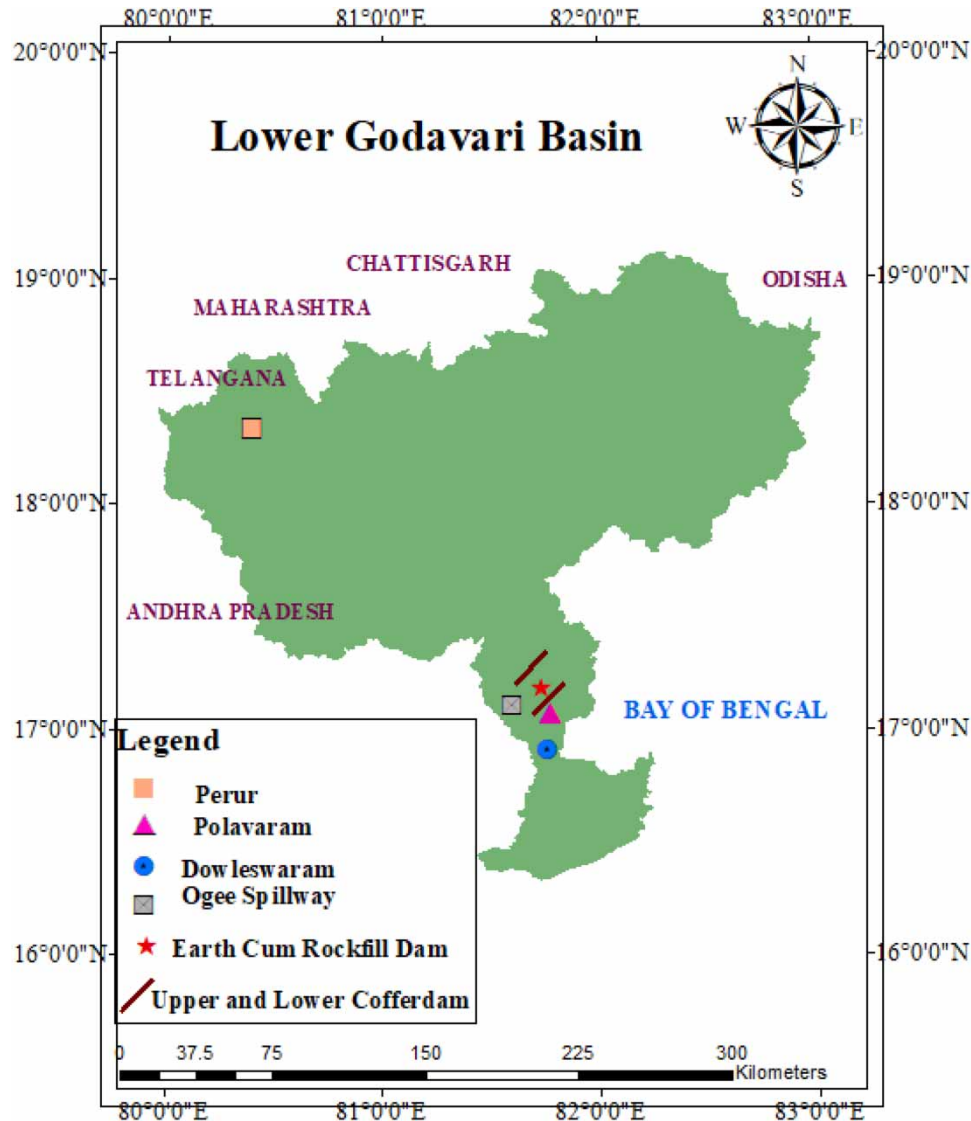


Figure 1 | Lower Godavari Basin.

various time-scale information features from the meteorological data. It uses three sigmoid functions and a *tanh* function introducing input, forget, and output gates. Input gates quantify the extent of new information to be carried into the memory. Forget gates determine whether previous time stamp information on averaged rainfall and evapotranspiration should be retained or removed. Cell gates specify the memory of LSTM, and output gates control the values of hidden states.

Forward pass mechanism

Let x_t, x_{t-1} are the inputs at times t and $t-1$; $c_t, a_t, c_{t-1}, a_{t-1}$ are the cell states at times t and $t-1$ and $h_t, g_t, h_{t-1}, g_{t-1}$ are outputs at times t and $t-1$; i_t, o_t , and f_t are input, output, and forget gates, respectively at time t ; W_i, W_o , and W_f are weight vectors. b_i, b_o , and b_f are biases of input, output, and forget gates. \emptyset represents sigmoid activation function. Input, output, and forget gates (Qi *et al.* 2019) are governed by Equations (1)–(3), which are as follows:

$$i_t = \emptyset (W_i |h_{t-1}, x_t| + b_i) \quad (1)$$

$$o_t = \emptyset (W_o |h_{t-1}, x_t| + b_o) \quad (2)$$

$$f_t = \emptyset (W_f |h_{t-1}, x_t| + b_f) \quad (3)$$

The activation functions Ψ of input, cell gate, and outputs are denoted by *softsign*, *tanh*, and Rectified Linear Activation Function (*relu*), respectively. The symbol \otimes represents the multiplication factor. W_{is} and b_{is} are the input state matrix and corresponding bias states. The obtained outputs c_t , h_t , a_t , g_t are presented in Equations (4)–(7):

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \Psi(W_{is}|h_{t-1}, x_t| + b_{is}) = f_t \otimes c_{t-1} + i_t \otimes \Psi(i_s) \quad (4)$$

$$h_t = o_t \otimes \Psi(c_t) \quad (5)$$

$$a_t = f_t \otimes a_{t-1} + i_t \otimes \Psi(W_{is}|g_{t-1}, x_t| + b_{is}) = f_t \otimes a_{t-1} + i_t \otimes \Psi(i_s) \quad (6)$$

$$g_t = o_t \otimes \Psi(a_t) \quad (7)$$

Let i_1, i_2, \dots, i_n denote data entered and c_{t-1}, a_{t-1} denote data obtained from Equations (4)–(7). The six distinct Bi-LSTM weights (Wang *et al.* 2019) will be updated until the desired output is achieved.

The working steps of Bi-LSTM are as follows:

- (1) Identify Bi-LSTM layer nodes, the number of neurons, learning rate, batch size, dropout, the number of epochs, and termination criteria.
- (2) Compute cell states (c_{t-1}, a_{t-1}).
- (3) Let the cell states pass through the forward and backward layers for updating six unique weights. This updating of weights (Wang *et al.* 2019) is continued until termination criteria are satisfied.

Internal components of the configuration model are referred to as parameters, and how these perform will depend on the training dataset used. The explicitly stated parameters that regulate the training process are known as hyperparameters and are established as part of the training process. In Bi-LSTM, all parameters are considered important and treated as hyperparameters. These are Bi-LSTM layer node (P_{L1}), Learning rate (P_{L2}), Batch size (P_{L3}), the number of neurons (P_{L4}), Epochs (P_{L5}), and Dropout (P_{L6}), respectively. The weights updated in input, forget, memory, and output gates are influenced by P_{L1} to P_{L6} (refer to Table 1). P_{L1} , P_{L4} , and P_{L5} positively influence the model performance when they are increased. They help to store the information in cell states and enhance the scope for efficiently updating weights. P_{L2} and P_{L3} govern the acceleration mechanism of weights updated. Lower values of P_{L2} and P_{L3} are preferable as the model could learn all the complex patterns effectively. P_{L6} helps improve the processing time, thereby reducing biases. Therefore, these are assessed for identifying optimum values using the Random Search hyperparameter tuning method. The Random Search method is

Table 1 | Hyperparameters of Bi-LSTM

Index (1)	Parameters, their range, and optimal value (2)	Remarks (3)
P_{L1}	Bi-LSTM layer node 16–256 (64)	Bi-LSTM layer nodes are hidden layer nodes that aid in data conveyance from input, forget, memory, and output gates. The larger the Bi-LSTM layer nodes, the more comprehensive the information from the given data stored in the cell state, resulting in a better output.
P_{L2}	Learning rate 0–1 (0.2)	The learning rate describes the step size at which the weights are altered to achieve minimal loss function. However, larger learning rates may result in underfitting, while very low values may result in an overfitting scenario.
P_{L3}	Batch size 16–256 (64)	Batch size specifies the number of samples sent to the network instantly. More batch sizes result in an erroneous capture of complex data patterns.
P_{L4}	Number of neurons 16–256 (128)	The number of neurons available in a dense model layer plays a major role in model performance. Overfitting may arise if there are more neurons than what is desirable.
P_{L5}	Epochs 10–100 (15)	Epoch refers to one cycle through the entire training dataset. However, a more significant number of epochs will consume memory, rapidly increasing the processing time.
P_{L6}	Dropout 0–0.5 (0.25)	It refers to data or noise intentionally dropped from a neural network to improve processing and reduce overall computational time.

chosen for tuning due to its ease of applicability and ability to generate infinitive parameter variations (Elgeldawi *et al.* 2021). The ranges of parameters related to Bi-LSTM with remarks are described in Table 1 (columns 2–3).

Hyperparameters influencing Bi-LSTM and their optimal values of parameters are presented in Table 1 (column 2). The optimal value of bi-layer nodes is closer to the lower threshold value, implying that a smaller number of forward and backward hidden layer nodes are sufficient to provide the desired output while preserving the most insightful information in the cell state. The optimal learning rate is detected nearer to the lower threshold, indicating that a smaller step size is picked to increase the likelihood of obtaining a minimal loss function. Optimal batch size and bi-layer node values are similar. Greater bi-layer nodes allow better conveyance of weight modifications from input, forget, memory, and output gates, resulting in superior output.

In contrast, the smallest batch size is recommended for improved model performance. An ideal number of neurons lie precisely in the middle of the lower and upper threshold limits. Interestingly, the thresholds for batch size, bi-layer node number, and the number of neurons are similar. The optimal number of epochs is closer to the lower threshold, implying a relatively short computational time. The optimal dropout value is found exactly at the center value of the lower and upper thresholds, implying that the input data is moderately noisy.

WNN (Wavelet Neural Network)

WNN (Guo *et al.* 2022) works based on signal processing. This study chooses the WNN method to pre-process the meteorological inputs by removing noise through wavelet decomposition, thereby enhancing the simulated streamflow accuracy over traditional ANN. This pre-processing using wavelet aids in better-capturing patterns from the given meteorological information. The philosophy behind using wavelets is to capture adequate information from primary meteorological data and decompose the same into details and approximations with the help of wavelet transforms. These behave distinctly in each sub-time series in comparison with the primary data. The decomposition process can be iterated with successive approximations so that the primary data is partitioned into many lower-resolution components. Scaling (dilation) and shifting (translation) are the two prominent factors in wavelet decomposition. This wavelet transforms processed via high-pass and low-pass filters (Alexandridis & Zapanis 2013).

A typical WNN is a three-layered structure comprising input, hidden, and output layers. In the input layer, the primary data is established. Hidden layer(s) comprises hidden units termed as wavelons. In hidden layers, primary input data introduced is transformed into scaling and shifting mother wavelet versions. Finally, the approximations of target values are estimated in the output layer. The mathematical formulation of WNN is formulated as follows:

$$y(x) = w_{\alpha} + \sum_{j=1}^{\alpha} w_j M(z_{ij}) + \sum_{i=1}^n w_{ij} \cdot x_i \quad (8)$$

where $y(x)$ is the simulated output values; α is the number of wavelons, x_i is the input value; w_i and w_j are the input and hidden network weights; w_{α} is constant; n is the number of inputs; z_{ij} is the input to wavelon normalized with $((x_i - b)/a)$; b and a are the translation and dilation parameters; and $M(z_{ij})$ is intermediate outputs based on chosen mother wavelet.

Equation (8) typically represents the single hidden layer mechanism of feed-forward WNN. When no hidden units exist, Equation (8) acts like a linear model. The first term in the equation represents the constant (w_{α}). The second term represents the hidden layer quantified as the product of associated network weight (w_j) and a mother wavelet. The mother wavelet is a function of weighted input, translation, and dilation parameters. The third term represents the product of input network weight and corresponding input value.

The working steps of WNN are as follows:

- (1) Identification of hidden units, the number of epochs, termination criteria, type of mother wavelet, and initialization of a and b .
- (2) Assigning initial weight vectors (network weights w_i and w_j).
- (3) Computation of z_{ij} based on a and b , corresponding $M(z_{ij})$ and $y(x)$.
- (4) Comparison of $y(x)$ with observed to compute the associated error.
- (5) Updating weights and other parameters until termination criteria are satisfied.

Mother wavelets

Mother wavelets (Stepanov 2017) are the WNN's crucial components, which decide the pattern of scaling and shifting factors within the wavelons in the hidden layer. These play a vital role in decomposition, possess high shifting sensitivity, and significantly impact the overall model performance. Four mother wavelets – Gaussian, Mexican hat, Morlet, and Shannon – are used in the present study due to their simplicity and easily understandable mathematical background. In addition, Gaussian, Mexican hat, and Morlet are non-orthogonal mother wavelet functions with the merit of functioning in discrete and continuous transforms. Relevant mathematical expressions are as follows (Equations (9)–(12)):

$$\text{Gaussian, } M(z_{ij}) = z_{ij} \cdot e^{z_{ij}^2/2} \quad (9)$$

$$\text{Mexican hat, } M(z_{ij}) = (1 - z_{ij}^2) \cdot e^{z_{ij}^2/2} \quad (10)$$

$$\text{Morlet, } M(z_{ij}) = e^{-z_{ij}^2/2} \cdot \cos 5z_{ij} \quad (11)$$

$$\text{Shannon, } M(z_{ij}) = \sin c(t) \cdot e^{-2\pi it} \quad (12)$$

WNN mechanism is significantly influenced by the parameters described in Table 2. Here also, all the parameters are considered important and treated as hyperparameters. These are the number of hidden layers (P_{W1}), the number of wavelons in the first layer (P_{W2}), the number of wavelons in the second layer (P_{W3}), activation function (P_{W4}), dropout (P_{W5}), and learning rate (P_{W6}), respectively. The parameters P_{W1} , P_{W2} , P_{W3} , P_{W4} , and P_{W6} control the network weights while the input data is navigated via hidden layers to generate the desired outcome. P_{W5} is an intentionally dropped wavelons parameter from the neural network to improve the processing and reduce the overall computation time. Table 2 presents the parameter ranges with remarks on four wavelets employed (columns 3–7).

Optimal values of hyperparameters influencing WNN for four wavelets – Gaussian, Mexican hat, Morlet, and Shannon – are presented in Table 2 (columns 3–6). Two hidden layers were identified in the Gaussian with 256, 32 in the first and second hidden layers, respectively. These are 128, 32 wavelons in Morlet wavelets. On the other hand, the Mexican hat has a single hidden layer with 128 wavelons. In the Shannon wavelet, no hidden layer is identified. Mexican hat uses *relu* activation function, whereas all other wavelets use *tanh* activation function to activate weights and biases. Dropout values are similar in Gaussian, Mexican hat, and Morlet wavelets. In contrast, no dropout is seen in the Shannon wavelet. Compared with the other two mother wavelets, the learning rates for Shannon and Morlet wavelets are lower and higher, respectively.

XGBoost (eXtreme Gradient Boosting)

XGBoost (Wang *et al.* 2022) is designed to be computationally efficient and highly effective. This method is chosen for its simpler yet effective structure and does not need any pre-processing or updating of information in gated units. This statistical framework comprises features, weak learners, and an additive model with the aim of loss function minimization. Decision trees are considered weak learners and are constructed based on similarity scores. These are constrained by the number of leaves, nodes, splits, or layers. Shorter decision trees possessing a single split are called stump/decision stump. Larger decision tree splits generally range from 4 to 8. However, more splits can be used, which may be computationally expensive. The additive trees are introduced without replacing the existing trees. A gradient descent procedure can be implemented to minimize the losses due to adding trees.

XGBoost approximate loss function with Taylor series for a weak learner and to construct a tree. The separation that yields in highest loss reduction are selected. The constructed tree begins at node '*i*' divided into either left or right branches, depending on the chosen separation criteria. Loss reduction can be calculated, and the branch with the highest loss reduction is preferred. XGBoost can be mathematically expressed (Chen

Table 2 | Hyperparameters of WNN

Index (1)	Parameters (2)	Range (Optimal value) for Gaussian (3)	Range (Optimal value) for Mexican hat (4)	Range (Optimal value) for Morlet (5)	Range (Optimal value) for Shannon (6)	Remarks (7)
P _{W1}	Number of hidden layers	0–3 (2)	0–3 (1)	0–3 (2)	0–3 (0)	The number of hidden layers is associated with an input and output layer, allowing wavelons to take input and generate output. Overfitting takes place due to an increase in hidden layers. Although fewer hidden layers reduce computational work, it may result in underfitting.
P _{W2}	Number of wavelons in the first layer	32–512 (256)	32–512 (128)	32–512 (128)	NA	The number of wavelons refers to the hidden units in the first hidden layer that would navigate the updated weighted inputs through the corresponding hidden layers. A large number of wavelons increases the accuracy of the model.
P _{W3}	Number of wavelons in the second layer	32–512 (32)	NA	32–512 (32)	NA	The number of wavelons refers to the hidden units in the second hidden layer that would navigate the updated weighted inputs through the first hidden layer resulting in an output. A large number of wavelons increases the accuracy of the model.
P _{W4}	Activation function	tanh, relu (tanh)	tanh, relu (relu)	tanh, relu (relu)	tanh, relu (tanh)	This function controls the activation and deactivation of wavelons as they move from the input to the output layer. The suitable activation function is chosen based on the type of problem and the mother wavelet.
P _{W5}	Dropout	0–0.5 (0.25)	0–0.5 (0.25)	0–0.5 (0)	0–0.5 (0.25)	Dropout is a parameter that allows specific wavelons to be selectively removed from a neural network to optimize processing and reduce computing time. If the dropout rate exceeds a specific threshold, it shows that the model is unsuitable for the provided input datasets. Higher dropout indicates that the given datasets are noisy.
P _{W6}	Learning rate	0–1 (8.68 × 10 ⁻⁴)	0–1 (10 ⁻³)	0–1 (2.04 × 10 ⁻³)	0–1 (5.92 × 10 ⁻⁴)	Learning rate determines the step size where weights are updated to achieve the smallest loss function. Higher learning speeds up convergence, but it may result in the loss of intricate patterns from information available, leading to underfitting.

& Guestrin 2016) as follows (refer to Equations (13)–(15)):

$$y_i^{(t)} = y_i^{(t-1)} + \eta \cdot w_j^* \quad (13)$$

$$L(y_i, p_i) = \frac{1}{2} (y_i - p_i)^2 \quad (14)$$

$$w_j^* = \frac{a_1 + b_1}{2 + \lambda} \quad (15)$$

Here, $y_i^{(t)}$ is the predicted value based on an initial guess; $y_i^{(t-1)}$ is the initial guess; w_j^* is the similarity score for the j th leaf node; η is the learning rate; y_i is the observed values; p_i is the predicted values; $L(y_i, p_i)$ is the loss function; a_1, b_1 are the ordinates of a particular leaf node; and λ is the regularization parameter.

The working steps of XGBoost are as follows:

- (1) Identification of tree depth, number of leaves, verbosity, learning rate, regularization parameter, termination criteria, and other related parameters.
- (2) Initial guess of predictor values $y_i^{(t-1)}$, selection of the type of loss function.
- (3) Selection of similarity score as splitting criteria for the decision trees. The corresponding leaf nodes having maximum similarity scores are chosen.
- (4) Computation of predictor values $y_i^{(t)}$.
- (5) Continuity of epochs till the objective of minimizing the loss function is achieved.

Here, the parameters considered are subsample (P_{X1}), the number of estimators (P_{X2}), minimum child weight (P_{X3}), maximum depth (P_{X4}), learning rate (P_{X5}), colsample bytree (P_{X6}), lambda (P_{X7}), alpha (P_{X8}), verbosity (P_{X9}), and maximum bin (P_{X10}), respectively. Of the parameters mentioned, P_{X1} to P_{X6} significantly influence the loss function; thus, they are considered hyperparameters. P_{X1} is used to select subsample data that should be considered for training purposes. P_{X2} , P_{X3} , P_{X4} , and P_{X6} govern the partitioning of decision trees, impacting the similarity score. P_{X5} decides the pace of the learning process from which iterated predictor values are obtained with a minimal loss function. P_{X7} to P_{X10} do not significantly influence the simulated predictor values. Thus, they are assigned default values. P_{X7} and P_{X8} are regularization parameters added as a penalty to the loss function to minimize the overall error and to counter the overfitting scenario. Table 3 presents the parameter ranges with remarks (columns 2–3).

Optimal values of hyperparameters influencing the XGBoost are presented in Table 3 (column 2). The optimal number of estimators is the same as the lower threshold, revealing that the model's minimal value of estimators is

Table 3 | Hyperparameters of XGBoost

S.No (1)	Parameters, their range, and optimal value (2)	Remarks (3)
P_{X1}	Subsample 0.5–1 (0.8)	Subsample is a randomly chosen portion of the trained data before building decision trees which would help avoid overfitting.
P_{X2}	Number of estimators 100–1,000 (100)	The number of estimators is the actual population of estimators used in forming decision trees for achieving the minimum loss function. However, the number of estimators picked for a specific problem should be determined, resulting in a considerable reduction in the loss function.
P_{X3}	Minimum child weight 0–10 (5.5)	The weight assigned for deciding the successive partitioning of the decision tree is described as the minimum child weight. The greater the minimal child weight values, the more conservative the partitioning in the building tree. Lower values are recommended to produce more decision trees, which improve the convergence rate; nevertheless, lower values are computationally more expensive.
P_{X4}	Maximum depth 3–15 (8)	The number of leaves from the root to the farthest leaf defines the maximum depth. The greater the maximum depth of a tree, the more complex the model, which overfits and aggressively consumes memory. Lower than optimal values, on the other hand, would result in the development of an insufficient number of decision trees, which would have a negative impact on the convergence criteria.
P_{X5}	Learning rate 0–1 (0.55)	The learning rate is the step size at which the weights are updated to get the minimal loss function. A lower learning rate would increase the likelihood of pinpointing precise outcomes. The higher the learning values, the more conservative the boosting procedure develops.
P_{X6}	Colsample bytree 0.5–1 (0.65)	Colsample bytree is the column subsample ratio used in the development of each tree. The value specifies the fraction of columns to be subsampled. Higher values of colsample make the model more conservative.
P_{X7}	Lambda Default (1)	Lambda value indicates the L_2 regularization term on leaf weights which are used for adjusting the loss function and countering the overfitting by summing up the square of feature coefficients.
P_{X8}	Alpha (α) Default (2.75)	Alpha values indicate the L_1 regularization term on leaf weights, which are used to adjust the loss function and counter the overfitting by summing up the feature coefficients.
P_{X9}	Verbosity Default (1)	Verbosity is the adjustment made in the XGBoost to facilitate visualization of the training process.
P_{X10}	Maximum bin Default (256)	The maximum bin is chosen to bucket the feature values.

adequate for this problem. The model is inferred to be conservative in developing decision trees for attaining the minimal loss function. Optimal values for minimum child weight and colsampling are found slightly inclined towards the upper threshold. The optimal tree depth value leans slightly towards the lower threshold value, suggesting that the model is simple. The optimal learning rate value is slightly inclined towards the upper threshold, inferring fairly low computational time with a higher convergence rate.

RESULTS AND DISCUSSION

Comparison of the observed and simulated discharge

The simulated discharges obtained using hyperparameter tuning of the three ML models are compared with the observed discharge. It is further evaluated using R^2 , RMSE, NSE, and PBIAS metrics, and the related discussion follows (refer to Table 4). Relevant observations (in addition to remarks) are presented in Table 4. WNN-Gaussian wavelet has displayed similar model performance in the training period compared with the WNN-Mexican hat. The model has fallen short in the validation period based on R^2 obtained. WNN-Mexican hat displayed better performance than the WNN-Gaussian wavelet in the training and validation periods with relatively lower RMSE values. Despite a higher learning rate, WNN-Morlet is found to dominate all other WNN models in the training period. However, it failed to show consistency in the validation period compared with other WNN models.

The WNN-Gaussian was inferior to other WNN models, even at low learning rates. WNN-Shannon wavelet showed a better consistency than other WNN models in the training and validation periods based on the R^2

Table 4 | Information on metrics of Bi-LSTM, WNN, and XGBoost in the training and validation periods

S.No	Model	Training				Validation				Remarks
		R^2	RMSE	NSE	PBIAS (%)	R^2	RMSE	NSE	PBIAS (%)	
1	Bi-LSTM	0.60	1.73	0.59	12.4	0.58	1.96	0.57	13.0	This model has shown a satisfactory performance with an R^2 of 0.6 and 0.58 in the training and validation periods. This model structure falls short compared with XGBoost, although it proved better than WNN.
2	WNN (Gaussian)	0.23	2.12	0.23	0.3	0.22	2.20	0.22	1.0	The model has shown unsatisfactory performance both in the training and validation periods.
3	WNN (Mexican hat)	0.22	2.06	0.23	0.3	0.22	2.13	0.22	1.0	Relatively low metrics were displayed by Mexican hat in the present study compared with other WNN models. Overall, the model has been unsatisfactory during training and validation periods.
4	WNN (Shannon)	0.26	2.30	0.26	1.14	0.25	2.31	0.25	1.46	This wavelet has shown a better consistency in the validation period than the other employed mother wavelets, although the overall model performance is unsatisfactory in both the training and validation periods.
5	WNN (Morlet)	0.29	2.06	0.29	0.9	0.22	2.39	0.21	1.8	Morlet wavelet is slightly better than other mother wavelets in training but lacks consistency in validation. Overall model performance can be rated as unsatisfactory for the present study.
6	XGBoost	0.88	1.49	0.86	29.3	0.86	1.63	0.85	28.5	The model is exceptional, with an R^2 of 0.875 in training. It has also displayed a great consistency in validation with an R^2 of 0.863. Thus, this model helps derive reliable streamflow forecasts compared with other models employed in this study.

value. However, the RMSE values were relatively higher than the other WNN models. Similar inferences are drawn for NSE and PBIAS.

The WNN could not display better simulation accuracy in any of the four variants due to a lack of spatial information on the average rainfall. Bi-LSTM has shown a satisfactory performance dominating all WNN models. However, it fell short when compared with XGBoost. Possible reasons for this under-performance of Bi-LSTM could be the gated units' inability to extract information from the provided data. XGBoost has shown exceptional performance compared with Bi-LSTM and WNN, as observed from the identified metrics in the training and validation periods, as shown in [Table 4](#).

The simulated hydrographs obtained by XGBoost during training (1982–2013) and validation (2014–2020) matched well with the observed discharges having minimal deviations. No phase errors were observed in both training and validation periods. Amplitude errors are observed due to underestimations of the model in the years 1983, 1988, 2002, 2003, 2006, and 2008 in the training period and 2018 and 2020 in the validation. This may be due to spatial variability of rainfall and consideration of the continuous simulation approach, which has an adverse impact on peak streamflow ([Potdar et al. 2021](#)). The trained and tested parameters obtained from XGBoost using historical data hold good even for future time horizons in projecting streamflow.

Projecting streamflow using XGBoost

XGBoost is further considered for forecasting streamflow in the study area for 2021–2099 using the climate data obtained ([Mishra et al. 2020](#)). The future streamflow time period is split into eight decadal horizons, namely, D₁ (2021–2030), D₂ (2031–2040), D₃ (2041–2050), D₄ (2051–2060), D₅ (2061–2070), D₆ (2071–2080), D₇ (2081–2090), and D₈ (2091–2099) for in-depth analysis. They are also compared with the historical time horizons; namely, H₁, H₂, and H₃ (refer to Supplementary Material, Table S1), to assess how streamflow varies/deviates in eight decadal future time horizons.

The daily average streamflow, highest streamflow, occurrence of significant streamflow events, and percentage increase in daily average streamflow for eight decadal time horizons SSP-wise are discussed in this section (refer to [Table 5](#)).

Daily streamflow analysis

Percentages of the streamflow events observed in less than 10 mm/day (ratio of the total number of records less than 10 mm/day to the total number of records for SSP585 within each decade) are 96.39, 97.48, 96.28, 93.43, 98.52, 90.34, 94.19, and 96.23% for D₁ to D₈, respectively (column 3). A similar trend is observed for other SSPs. D₇ has 41 events in the 30–50 mm/day in SSP245. These are 67 for D₆ in 20–30 mm/day in SSP585 category; 257 in D₆ for 10–20 mm/day in SSP126 category (columns 6 and 4), respectively. However, no streamflow events were identified in D₁, D₂, D₃, and D₅ (column 7). The number of streamflow events greater than 65 mm/day is significantly found in D₆ and D₇ in SSP585. It is also observed that no significant streamflow events are identified in the remaining future time horizons (column 8).

The highest streamflow observed for D₁, D₃, D₄, D₅, and D₈ were 45.95, 51.54, 59.66, 38.25, and 48.77 mm/day in the SSP245 scenario. These are D₆ and D₇, with 72.82 and 79.04 mm/day in the case of SSP585. It is D₂ with 44.33 mm/day in SSP126 (column 9). An increasing trend is observed in daily average streamflows (column 10) from D₁ to D₄ for SSP126 to SSP585. On the contrary, an increase is observed from SSP126 to SSP245 for D₅, D₇, and D₈. After that, a decrease is observed from SSP245 to SSP370 and an increase from SSP370 to SSP585.

Monthly average streamflow analysis

The monthly average streamflow observed in eight decadal future time horizons is discussed in this section (refer to [Figure 2\(a\)–2\(h\)](#) for D₁ to D₈). It is observed that streamflow contributed majorly in July–September. From July–September, monthly streamflow contributions for D₁ were 166.53, 176.25, 172.23, and 166.36 mm for SSP126 to SSP585, respectively. They were significantly lower than H₁ (236.61 mm), slightly lower than H₂ (205.56 mm), and H₃ (196.65 mm).

However, the remaining months (other than July–September) have higher streamflow of 88.53, 110.52, 127.94, and 141.86 mm compared with historical scenarios of 65.35, 66.43, and 48.8 mm ([Figure 2\(a\)](#)). The status of D₂, D₃, D₅, and D₈ are similar to that of D₁ ([Figure 2\(b\)](#), [2\(c\)](#), [2\(e\)](#) and [2\(h\)](#)). Monthly streamflow contributions of D₄ from July to September were found to be 236.99, 252.98, 245.83, and 239.73 mm and higher than H₁, H₂, and H₃. Similarly, streamflow contributions for the remaining months are dominant at 177.68, 171.08, 220.42, and 231.27 mm compared with historical scenarios ([Figure 2\(d\)](#)). The situation of D₆ is almost similar to that of

Table 5 | Streamflow information of all time horizons

Horizons (1)		Total number of daily records (2)	Streamflow records lesser than 10 (mm/day) (3)	Streamflow records between 10–20 (mm/day) (4)	Streamflow records between 20–30 (mm/day) (5)	Streamflow records between 30–50 (mm/day) (6)	Streamflow records between 50–65 (mm/day) (7)	Streamflow records greater than 65 (mm/day) (8)	Highest streamflow (mm/day) (9)	Daily average streamflow (mm/day) (10)
D ₁	SSP126	3,652	3,512	117	15	8	0	0	41.34	2.10
	SSP245	3,652	3,502	110	21	19	0	0	45.95	2.35
	SSP370	3,652	3,510	111	20	11	0	0	40.96	2.47
	SSP585	3,652	3,520	102	18	12	0	0	38.14	2.53
D ₂	SSP126	3,653	3,542	84	19	8	0	0	44.33	2.25
	SSP245	3,653	3,529	78	34	12	0	0	38.39	2.32
	SSP370	3,653	3,545	69	28	11	0	0	36.19	2.46
	SSP585	3,653	3,561	59	25	8	0	0	35.95	2.51
D ₃	SSP126	3,652	3,497	121	22	12	0	0	38.56	2.48
	SSP245	3,652	3,490	122	22	17	1	0	51.54	2.78
	SSP370	3,652	3,497	123	16	16	0	0	37.81	2.79
	SSP585	3,652	3,516	106	13	17	0	0	41.45	2.89
D ₄	SSP126	3,653	3,412	178	33	30	0	0	49.60	3.41
	SSP245	3,653	3,369	200	41	34	9	0	59.66	3.48
	SSP370	3,653	3,392	181	38	38	4	0	51.41	3.83
	SSP585	3,653	3,413	166	32	40	2	0	50.58	3.87
D ₅	SSP126	3,652	3,584	60	8	0	0	0	28.02	1.98
	SSP245	3,652	3,570	67	14	1	0	0	38.25	2.12
	SSP370	3,652	3,610	41	1	0	0	0	21.73	1.99
	SSP585	3,652	3,598	48	6	0	0	0	24.78	2.27
D ₆	SSP126	3,653	3,288	257	65	31	11	1	65.72	4.10
	SSP245	3,653	3,422	168	41	21	1	0	53.58	3.04
	SSP370	3,653	3,350	208	54	28	13	0	63.87	3.86
	SSP585	3,653	3,300	237	67	31	11	7	72.82	4.41
D ₇	SSP126	3,652	3,459	130	38	23	2	0	54.86	3.26
	SSP245	3,652	3,438	141	20	41	11	1	66.59	3.31
	SSP370	3,652	3,476	123	18	26	8	1	69.32	2.98
	SSP585	3,652	3,440	147	24	29	7	5	79.04	3.40
D ₈	SSP126	3,287	3,163	102	20	2	0	0	35.72	2.60
	SSP245	3,287	3,146	114	16	11	0	0	48.77	2.74
	SSP370	3,287	3,206	71	10	0	0	0	27.70	2.55
	SSP585	3,287	3,163	105	18	1	0	0	31.59	2.90

Bold numbers represent the highest value.

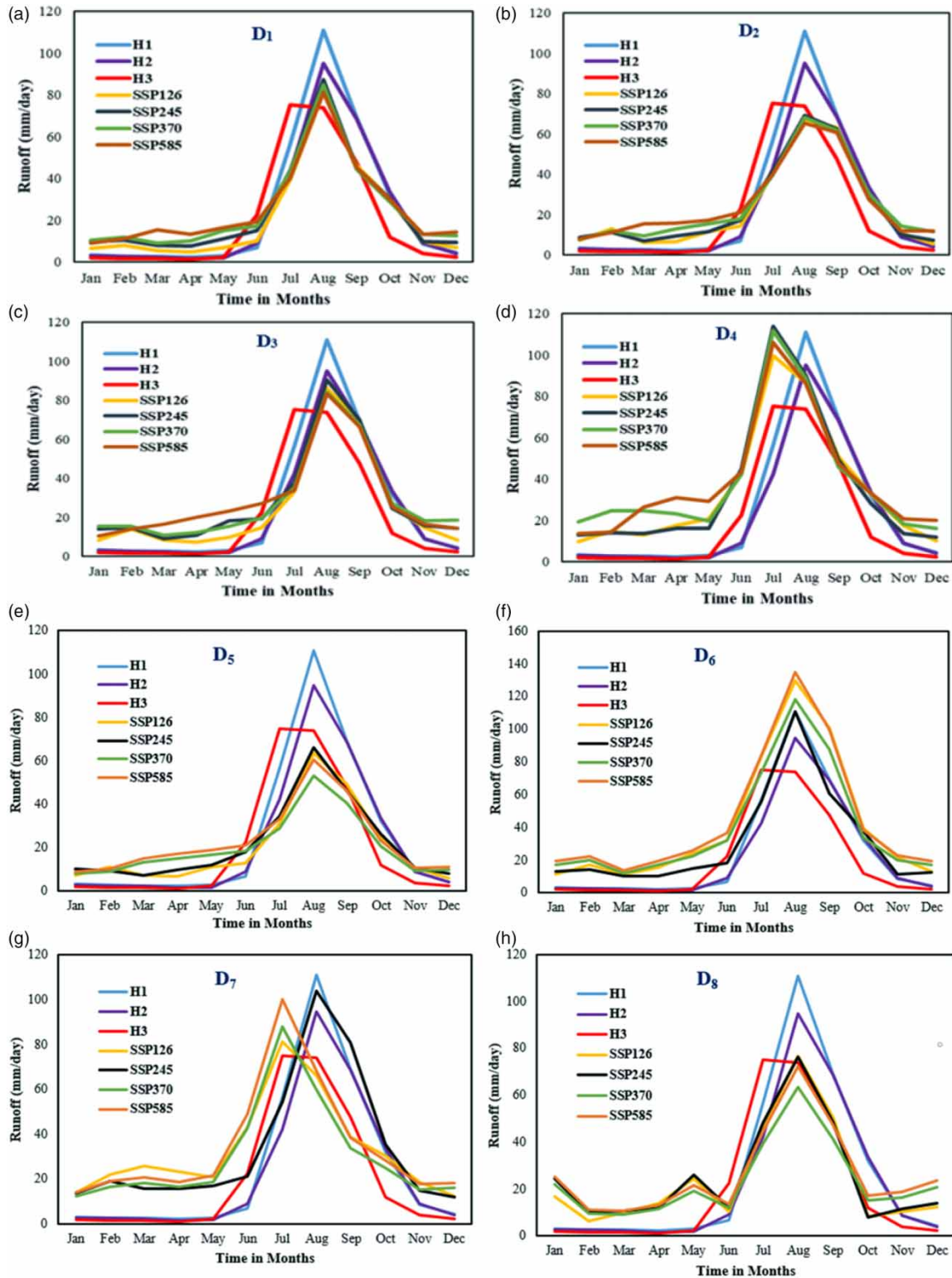


Figure 2 | (a–h) Month-wise streamflow average variations in D₁ to D₈ and three historical time horizons.

D₄. The only difference is streamflow contribution in July–September in SSP245 is 226.8 mm and is less than H₁ (Figure 2(f)). The monthly streamflow contributions of D₇ from July to September were 185.99, 239, 181.1, and 206.52 mm, which are lower than H₁, H₂, and H₃ for SSP126 and SSP370. These are (a) higher than all three historical time horizons in the case of SSP245, (b) lower than H₁, and (c) higher than the other two historical time horizons in the case of SSP585. However, the remaining months have higher streamflow of 210.3, 164.15, 181.5, and 206.96 mm compared with historical scenarios (Figure 2(g)).

The highest percentage increase in future streamflow average when compared with H₁, H₂, and H₃, respectively, are: D₆ for SSP126 with 114.66, 138.37, and 164.52%, respectively; D₄ for SSP 245 with 82.2, 102.33, and 124.52%, respectively; D₆ for SSP370 with 102.01, 124.24, and 149.03%, respectively; D₆ for SSP585 with 130.9, 156.4, and 184.52%, respectively.

The lowest percentage increase in future streamflow average when compared with H₁, H₂, and H₃, respectively, are: D₄ for SSP126 with 3.66, 15.12, and 27.74%; D₅ for SSP 245 with 10.99, 23.26, and 36.77%; D₅ for SSP 370 with 4.19, 15.70, and 28.39%, respectively; D₅ for SSP 585 with 18.85, 31.98, and 46.45%, respectively.

In each future time horizon, the highest deviation is observed from H₃, followed by H₂ and H₁. It indicates that future streamflow is significantly higher than the historical time horizon, likely to serve the requirements of LGB. Interestingly, deviations are in increasing order from SSP126 to SSP585 for D₁. For example, deviation of H₁ from D₁ in case of SSP126, SSP245, SSP370, and SSP585 is 9.95, 23, 29.3, and 32.5%. Similar are the inferences for D₂ to D₄.

SUMMARY AND CONCLUSION

Three ML models, Bi-LSTM, WNN, and XGBoost, were applied to simulate streamflows of the LGB. Four mother wavelets, namely, Gaussian, Mexican hat, Morlet, and Shannon, were employed in WNN. Random Search-based hyperparameter tuning was employed to fine-tune the parameters in these methods. Thirty-nine years of daily rainfall, evapotranspiration, and streamflow data were used for training and validation. XGBoost was found superior and was applied for projecting streamflow in the basin. Eight future decadal time horizons were proposed to study the climate change impacts. The following are the notable conclusions:

- XGBoost performed exceptionally well in the training period with an R^2 , RMSE, NSE, and PBIAS of 0.88, 1.48, 0.86, and 29.3% during training, and corresponding values during validation are 0.86, 1.63, 0.85, and 28.5%, respectively.
- Bi-LSTM has shown a satisfactory performance with an R^2 of 0.6 in training and 0.58 in validation periods.
- Morlet mother wavelet has shown better metrics in training, even with a high learning rate.
- The highest streamflows were identified mainly in the SSP245 scenario in future decadal time horizons.
- SSP245 is observed to have a higher percentage of July–September monthly streamflow contributions than the other SSPs in all future decadal time horizons. Streamflow contributions in the remaining months were more prominent than July–September in eight future decadal time horizons compared with the three historical time horizons.

ACKNOWLEDGEMENTS

This research work is supported by the Council of Scientific and Industrial Research, New Delhi, through Project no. 22(0782)/19/EMR-II dated 24.7.19. Special acknowledgements and gratitude to Prof Vimal Mishra, IIT Gandhinagar, for making CMIP6 data available. The authors are thankful to Prof D. Nagesh Kumar, Indian Institute of Science, Bangalore and Dr RN Sankhua, Mr Shankar Rao, officials of Godavari River Basin, for providing time for brainstorming discussions; IMD and CWC for providing necessary data.

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

- Alexandridis, A. K. & Zapranis, A. D. 2013 *Wavelet neural networks: a practical guide*. *Neural Networks* **42**, 1–27.
- Bai, H., Li, G., Liu, C., Li, B., Zhang, Z. & Qin, H. 2021 *Hydrological probabilistic forecasting based on deep learning and Bayesian optimization algorithm*. *Hydrology Research* **52**, 927–943.
- Bi, X. Y., Li, B., Lu, W. L. & Zhou, X. Z. 2020 *Daily runoff forecasting based on data-augmented neural network model*. *Journal of Hydroinformatics* **22**, 900–915.
- Chen, T. & Guestrin, C. 2016 XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

- Criswell, J. A. & Lin, E. B. 2021 River flow forecasting using an inverse wavelet transform neural network approach. *International Journal of Applied Mathematics, Computational Science and Systems Engineering* **3**, 67–70.
- Desmet, Q. & Ngo-Duc, T. 2022 A novel method for ranking CMIP6 global climate models over the Southeast Asian region. *International Journal of Climatology* **42** (1), 97–117.
- Elgeldawi, E., Sayed, A., Galal, A. R. & Zaki, A. M. 2021 Hyperparameter tuning for machine learning algorithms used for Arabic sentiment analysis. *Informatics* **8**, 79.
- Galavi, H. & Lee, T. S. 2012 Neuro-fuzzy modelling and forecasting in water resources. *Scientific Research and Essays* **7** (24), 2112–2121.
- Galavi, H., Mirzaei, M., Shul, L. T. & Valizadeh, N. 2013 Klang River-level forecasting using ARIMA and ANFIS models. *Journal of American Water Works Association* **105** (9), E496–E506.
- Galavi, H., Kamal, M. R., Mirzaei, M. & Ebrahimian, M. 2019 Assessing the contribution of different uncertainty sources in streamflow projections. *Theoretical and Applied Climatology* **137** (1), 1289–1303.
- Guo, T., Zhang, T., Lim, E., López-Benítez, M., Ma, F. & Yu, L. 2022 A review of wavelet analysis and its applications: challenges and opportunities. *IEEE Access* **10**, 58869–58903.
- Honorato, A. G. D. S. M., Silva, G. B. L. D. & Guimaraes Santos, C. A. 2018 Monthly streamflow forecasting using neuro-wavelet techniques and input analysis. *Hydrological Sciences Journal* **63** (15–16), 2060–2075.
- Horton, P., Schaeffli, B. & Kauzlaric, M. 2022 Why do we have so many different hydrological models? A review based on the case of Switzerland. *Wiley Interdisciplinary Reviews: Water* **9** (1), e1574.
- Iqbal, Z., Shahid, S., Ahmed, K., Ismail, T., Ziarh, G. F., Chung, E. S. & Wang, X. 2021 Evaluation of CMIP6 GCM rainfall in mainland Southeast Asia. *Atmospheric Research* **254**, 105525.
- Kabir, T., Pokhrel, Y. & Felfelani, F. 2022 On the precipitation-induced uncertainties in process-based hydrological modeling in the Mekong River Basin. *Water Resources Research* **58** (2), e2021WR030828.
- Kasiviswanathan, K. S., He, J., Sudheer, K. P. & Tay, J. H. 2016 Potential application of wavelet neural network ensemble to forecast streamflow for flood management. *Journal of Hydrology* **536**, 161–173.
- Kim, T., Yang, T., Gao, S., Zhang, L., Ding, Z., Wen, X. & Hong, Y. 2021 Can artificial intelligence and data-driven machine learning models match or even replace process-driven hydrologic models for streamflow simulation?: a case study of four watersheds with different hydro-climatic regions across the CONUS. *Journal of Hydrology* **598**, 126423.
- Konapala, G., Kao, S. C., Painter, S. L. & Lu, D. 2020 Machine learning assisted hybrid models can improve streamflow simulation in diverse catchments across the conterminous US. *Environmental Research Letters* **15** (10), 104022.
- Li, F., Guangwen, M. A., Shijun, S. & Huang, W. 2021 An ensemble modeling approach to forecast daily reservoir inflow using bidirectional long-and short-term memory (Bi-LSTM), variational mode decomposition (VMD), and energy entropy method. *Water Resources Management* **35**, 2941–2963.
- Liu, F., Xu, C., Long, Y., Yin, G. & Wang, H. 2022 Assessment of CMIP6 model performance for air temperature in the arid region of northwest China and subregions. *Atmosphere* **13** (3), 454.
- Mahmood, R., Donat, M. G., Ortega, P., Doblus-Reyes, F. J. & Ruprich-Robert, Y. 2021 Constraining decadal variability yields skillful projections of near-term climate change. *Geophysical Research Letters* **48** (24), e2021GL094915.
- Mirzaei, M., Yu, H., Dehghani, A., Galavi, H., Shokri, V., Mohsenzadeh Karimi, S. & Sookhak, M. 2021 A novel stacked long short-term memory approach of deep learning for streamflow simulation. *Sustainability* **13** (23), 13384.
- Mishra, V., Bhatia, U. & Tiwari, A. D. 2020 Bias-corrected climate projections for south Asia from coupled model intercomparison project-6. *Scientific Data* **7**, 338.
- Moges, E., Demissie, Y., Larsen, L. & Yassin, F. 2020 Review: sources of hydrological model uncertainties and advances in their analysis. *Water* **13**, 28.
- Ni, L., Wang, D., Wu, J., Wang, Y., Tao, Y., Zhang, J. & Liu, J. 2020 Streamflow forecasting using extreme gradient boosting model coupled with Gaussian mixture model. *Journal of Hydrology* **586**, 124901.
- Parding, K. M., Dobler, A., McSweeney, C. F., Landgren, O. A., Benestad, R., Erlandsen, H. B. & Loukos, H. 2020 GCMeval – an interactive tool for evaluation and selection of climate model ensembles. *Climate Services* **18**, 100167.
- Polavaram Project Authority Annual Report 2020–21 Ministry of Jal Shakti, Department of Water Resources, River Development & Ganga Rejuvenation, Government of India.
- Potdar, A. S., Kirstetter, P. E., Woods, D. & Saharia, M. 2021 Toward predicting flood event peak discharge in ungauged basins by learning universal hydrological behaviors with machine learning. *Journal of Hydrometeorology* **22** (11), 2971–2982.
- Qi, Y., Zhou, Z., Yang, L., Quan, Y. & Miao, Q. 2019 A decomposition-ensemble learning model based on LSTM neural network for daily reservoir inflow forecasting. *Water Resources Management* **33** (12), 4123–4139.
- Rabazanahary, T., Finaritra, M., Rahaman, H. & Zhai, J. 2021 Assessment of the future impact of climate change on the hydrology of the Mangoky River, Madagascar using ANN and SWAT. *Water* **13**, 1239.
- Rozos, E., Dimitriadis, P. & Bellos, V. 2022 Machine learning in assessing the performance of hydrological models. *Hydrology* **9**, 1–17.
- Sahour, H., Gholami, V., Torkaman, J., Vazifedan, M. & Saeedi, S. 2021 Random forest and extreme gradient boosting algorithms for streamflow modeling using vessel features and tree-rings. *Environmental Earth Sciences* **80**, 1–14.
- Sharghi, E., Nourani, V., Molajou, A. & Najafi, H. 2019 Conjunction of emotional ANN (EANN) and wavelet transform for rainfall-runoff modeling. *Journal of Hydroinformatics* **21**, 136–152.
- Song, Y. H., Chung, E. S. & Shahid, S. 2022 Differences in extremes and uncertainties in future runoff simulations using SWAT and LSTM for SSP scenarios. *Science of The Total Environment* **838**, 156162.

- Stepanov, A. B. 2017 Construction of activation functions for wavelet neural networks. In *XX IEEE International Conference on Soft Computing and Measurements (SCM)*. IEEE, pp. 397–399.
- Wang, H. H., Yu, L., Tian, S. W., Peng, Y. F. & Pei, X. J. 2019 [Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network](#). *Applied Intelligence* **49** (8), 3016–3026.
- Wang, S., Peng, H., Hu, Q. & Jiang, M. 2022 Analysis of runoff generation driving factors based on hydrological model and interpretable machine learning method. *Journal of Hydrology: Regional Studies* **42**, 101139.
- Zhang, J., Chen, X., Khan, A., Zhang, Y. K., Kuang, X., Liang, X. & Nuttall, J. 2021 [Daily runoff forecasting by deep recursive neural network](#). *Journal of Hydrology* **596**, 126067.

First received 14 August 2022; accepted in revised form 20 October 2022. Available online 1 November 2022