Journal of
Water & Climate Change

IWA PUBLISHING

Journal of Water and Climate Change Vol 15 No 2, 832 doi: 10.2166/wcc.2024.594

Deep learning algorithms and their fuzzy extensions for streamflow prediction in climate change framework

Rishith Kumar Vogeti [a], Rahul Jauhari[b], Bhavesh Rahul Mishra[c], K. Srinivasa Raju[a,*] and D. Nagesh Kumar [d]

[a] Department of Civil Engineering, BITS Pilani Hyderabad Campus, Hyderabad, India
[b] Department of Computer Science and Information Systems, BITS Pilani Hyderabad Campus, Hyderabad, India
[c] Department of Electrical and Electronics Engineering, BITS Pilani Hyderabad Campus, Hyderabad, India
[d] Department of Civil Engineering, Indian Institute of Science, Bangalore, India
*Corresponding author. E-mail: ksraju@hyderabad.bits-pilani.ac.in

RKV, 0000-0001-7493-4285

ABSTRACT

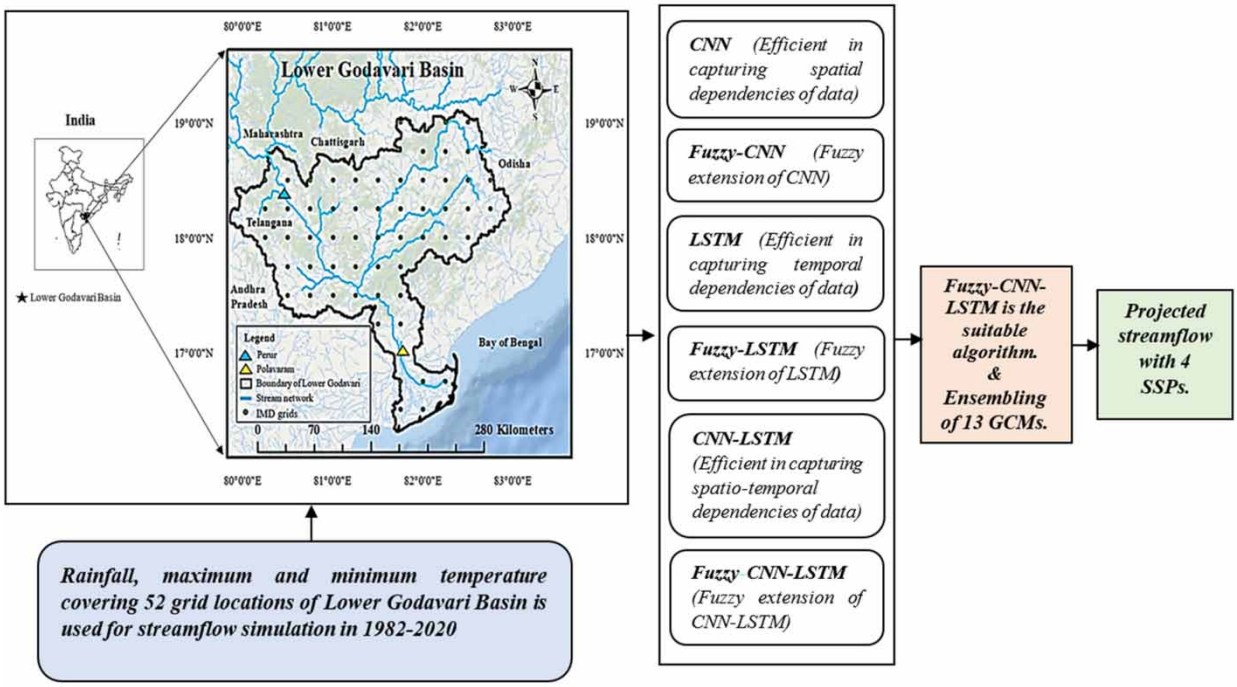
The present study analyzes the capability of convolutional neural network (CNN), long short-term memory (LSTM), CNN-LSTM, fuzzy CNN, fuzzy LSTM, and fuzzy CNN-LSTM to mimic streamflow for Lower Godavari Basin, India. Kling–Gupta efficiency (KGE) was used to evaluate these algorithms. Fuzzy-based deep learning algorithms have shown significant improvement over classical ones, among which fuzzy CNN-LSTM is the best. Thus, it is further considered for streamflow projections in a climate change context for four-time horizons using four shared socioeconomic pathways (SSPs). Average streamflow in 2041–2060, 2061–2080, and 2081–2090 are compared to that of 2021–2040 and it changed by +3.59, +7.90, and +12.36% for SSP126; +3.62, +8.28, and +12.96% for SSP245; +0.65, −0.01, and −0.02% for SSP370; +0.02, +0.71, and +0.06% for SSP585. In addition, two non-parametric tests, namely, Mann–Kendall and Pettitt were conducted to ascertain the trend and change point of the projected streamflow. Results indicate that fuzzy CNN-LSTM provides a more precise prediction than others. The identified variations in streamflow across different SSPs facilitate valuable insights for policymakers and relevant stakeholders. It also paves the way for adaptive decision-making.

Key words: climate, CNN, fuzzy, LSTM, streamflow

HIGHLIGHTS

- Fuzzy CNN-LSTM shows a significant improvement in KGE in training and testing periods over others.
- Incorporating a fuzzy inference layer in deep learning algorithms has substantially improved peak flow simulation.
- Mann–Kendall and Pettitt tests were conducted to ascertain the trend and change point of the projected streamflow of the basin.

GRAPHICAL ABSTRACT



1. INTRODUCTION

Streamflow prediction in the climate change context is essential for adaptive decision-making across various domains, from water resource management to disaster preparedness (Kumar *et al.* 2023). It significantly influences environmental sustainability and socioeconomic stability and impacts society broadly. However, planning becomes complex due to non-linear interactions among related factors with far-reaching implications. In this regard, it is of utmost importance to select appropriate model, for example, rational method (Lapides *et al.* 2021), conceptual (Manikanta & Vema 2022), lumped (Rosli *et al.* 2022), semi-distributed (Sabzipour *et al.* 2023), and distributed hydrological models (Sabitha *et al.* 2023). However, several challenges persist with these models. The rational method and conceptual hydrological models tend to fail in simulating peak flow situations, as evidenced by studies reported by Lapides *et al.* (2021) and Anshuman *et al.* (2021). Lumped models exhibit limitations in accounting for spatial and parameter heterogeneity within catchments, while semi-distributed models have a limited ability to capture spatial variability. On the other hand, fully distributed hydrological models are hindered by high data and computational requirements, as highlighted by Sidle (2021).

In this context, deep learning algorithms (Kim & Kim 2021; Kumar *et al.* 2023; Liang *et al.* 2023; Shekar *et al.* 2023) have emerged as promising alternatives for dealing with the aforementioned challenges. They demonstrate a strong non-linear fitting ability and high computational capability and do not entail an in-depth underlying physical mechanism of the basin (Shen 2018; Sit *et al.* 2020; Xu & Liang 2021; Fu *et al.* 2022; Ghobadi & Kang 2023; Tripathy & Mishra 2024).

Furthermore, the convolutional neural network (CNN) has been widely adopted in hydrological studies for its effectiveness in capturing spatial dependencies within the data. Sheng *et al.* (2023) studied the advantages and disadvantages of CNN, recurrent neural networks (RNN), and transformers for their ability to forecast runoff regarding accuracy, robustness, and interpretability. Song (2022) employed CNN to simulate streamflow in the Jojong watershed, South Korea and has shown satisfactory performance. Shu *et al.* (2022) developed DirCNN (Direct CNN) and DRCNN (Direct Recursive CNN) and compared them with three artificial neural networks (ANN) variations for the case studies in China. Two variations of CNN outperformed the ANN. Khosravi *et al.* (2022) applied five deep learning algorithms hybridizing with metaheuristic BAT to simulate streamflow, Korkorsar catchment, Iran. CNN-BAT demonstrated superior simulation efficacy over others. In summary, CNN is superior in capturing inherent local spatial dependencies within the data. However, they fail to extract the temporal dependencies within the data.

In this context, long short-term memory (LSTM) is found to tackle the temporal dependencies, addressing the vanishing and exploding gradient issues efficiently (Ghimire *et al.* 2021). Cheng *et al.* (2020) employed ANN and LSTM to forecast the streamflow of the Nan and Ping River Basins, Thailand. LSTM was found to be suitable. The influence of the time lag was also studied. Xu *et al.* (2020) applied five algorithms for simulating the streamflow of the Hun River system. The impacts of different parameters and structures on the performance were also studied. LSTM outperformed multiple linear regression and back propagation, achieved an accuracy level similar to Xinanjiang (XAJ), and was slightly worse than soil water assessment tool (SWAT).

Forghanparast & Mohammadi (2022) employed LSTM, CNN, self-attention LSTM, and extreme learning machine (ELM) in the tributaries of the Colorado River, Texas. LSTM performed better than ELM and CNN and competed with self-attention LSTM. Li *et al.* (2022a) employed improved LSTM, CNN, random forest (RF), decision tree regressor (DTR), and LSTM to Yarkant River. Improved LSTM was superior to others. Girihagama *et al.* (2022) employed the conventional and attention-based LSTM in 10 watersheds, Ottawa River, Canada. The attention-based model was superior to the conventional. Arsenault *et al.* (2023) studied the ability of LSTM, HSAMI, hydrological model of the École de technologie supérieure (HMETS), and Génie Rural à 4 paramètres Journalier (GR4J) at ungauged catchments in Northeastern part of North America. LSTM outperformed hydrological models for 93–97% of catchments.

Numerous studies were reported applying hybrid deep learning algorithms to captivate the merits of individual algorithms. Ni *et al.* (2020) proposed wavelet-LSTM and convolutional LSTM in a case study in China. The prediction accuracy was compared with LSTM and multi-layer perceptron (MLP). Proposed models improved the forecasting ability compared to LSTM. Li *et al.* (2022b) applied CNN-LSTM and SWAT to the Hun River Basin, China. CNN-LSTM had an edge over SWAT due to its non-linear learning ability. Bakhshi *et al.* (2023) hybridized LSTM with Kalman filter (KF), i.e., LSTM-KF, to improve streamflow forecasting for Dez Dam, Iran. They compared the results with LSTM, LSTM-Unscented Kalman Filter (UKF), LSTM-Kalman Filter Smooth (KFS), and LSTM-Unscented Kalman Filter Smooth (UKFS). LSTM-UKFS was found to be promising among employed. Ng *et al.* (2023) reviewed different hybrid deep learning algorithms and discussed their strengths, limitations, and future directions in streamflow forecasting.

Despite the frequent dominance of hybrid algorithms, they may underperform due to complex architecture. Vatanchi *et al.* (2023) found adaptive neuro fuzzy inference system and ANN better than CNN-GRU-LSTM and Bi-LSTM in simulating streamflow in Colorado River, USA. The study further suggests to examine the effect of data uncertainty.

In this context, it was opined that integrating the fuzzy inference layer with deep learning and hybrid architectures would provide a new direction leveraging the strength of fuzzy logic (Kambalimath & Deka 2020) to address the imprecision and vagueness in the data. Complimentarily, the role of climate change is indispensable in water resources planning and is reported by various researchers in the context of deep learning algorithms.

Nguyen *et al.* (2023) applied three global climate models (GCMs), coupled model intercomparison project phase 5 (CMIP5) framework for projecting streamflow of the Srepok, Sesan, and Sekong River Basins in Vietnam for two representative concentration pathways (RCPs) 4.5 and 8.5 scenarios for 2020–2100. Streamflow trends in the Sesan and Srepok basins are increasing, and vice versa in the Sekong basin. Roushangar & Abdelzad (2023) used three GCMs to project streamflow in Urmia and Amel cities in Iran for 2021–2040 using RCPs 2.6, 4.5, and 8.5. Mean runoff during the spring season for Urmia; spring and winter seasons for Amel would decrease compared to historical data. However, minimal studies are reported in the CMIP6 framework. Singh *et al.* (2023) used an ensemble of six CMIP6-GCMs for streamflow projection in the Sutlej River basin for 2041–2070 and 2071–2100 using two shared socioeconomic pathways (SSPs). It was concluded that streamflow would decrease from June to November in both scenarios. Song *et al.* (2022) deployed an ensemble of 11 CMIP6-GCMs to project streamflow in the Yeongsan River basin, South Korea, for 2021–2100 for SSP245 and SSP585. The study revealed a decrease in streamflow for both the SSPs.

Two notable gaps were identified from the literature review here and elsewhere. Firstly, there is a lack of fuzzy-based hybrid algorithms that would have improved simulation efficacy, and secondly, their application in climate change-based streamflow prediction.

The objectives are formulated accordingly for the case study of the Lower Godavari Basin, India, which are as follows:

1. To identify the competent algorithm among CNN, LSTM, CNN-LSTM, and their extensions, fuzzy CNN, fuzzy LSTM, and fuzzy CNN-LSTM.
2. To project streamflow using a competent algorithm from a climate change context.

In summary, developing a fuzzy-based hybrid algorithm(s) is a major innovation and contribution to the research community (de Campos Souza 2020). Also, predicting streamflow in the climate change framework helps policymakers formulate short-term and long-term planning strategies, facilitating efficient basin planning and management.

For simplification purposes, CNN-LSTM, fuzzy CNN, fuzzy LSTM, and fuzzy CNN-LSTM are denoted as C-LSTM, F-CNN, F-LSTM, and FC-LSTM.

2. STUDY AREA AND DATASET DESCRIPTION

Lower Godavari Basin (Figure 1) lies between latitude 17° 00′ to 19° 00′ N and longitude 80° 00′ to 83° 4′ E. Andhra Pradesh, Chhattisgarh, Odisha, Maharashtra, and Telangana states share the borders with the basin. It comprises a 39,180 km^2 catchment area and 462 km basin length. Basin experiences humid and dry sub-humid climates. During the southwest monsoon (July to August), rainfall and relative humidity are both high in the basin and low in April and May. The basin receives 1,096 mm of yearly rainfall (Sarkar 2022). The average temperature ranges from 26 to 44 °C. Evaporation rates range from 1,401 to 2,606.40 mm per year (Jhajharia *et al.* 2021). The basin has a significant water requirement for various purposes such as inter-state dependencies, canal stabilization, drinking, industrial, irrigation, etc. (Polavaram Project Authority 2021). Sustainable initiatives are required to tackle increasing demands in the context of climate change.

The present study considers rainfall, maximum and minimum temperatures from 1982 to 2020, i.e., 468 monthly records, as they are the most easily accessible vital variables influencing the streamflow (Parisouj *et al.* 2020). In addition, these variables are accessible even in future time segments (Meher & Das 2019). Observed streamflow information for Polavaram and Perur for the same period is obtained from the Central Water Commission. About 80% of records were employed for training and the remaining for testing (Wegayehu & Muluneh 2022; Syed *et al.* 2023). Future meteorological data for 2021–2099, i.e., 948 monthly records were predicted from 13 CMIP6-GCMs using the Empirical Quantile Mapping method for four SSPs (van Vuuren *et al.* 2017; Mishra *et al.* 2020).

3. DESCRIPTION OF ALGORITHMS

This section describes the algorithms for assessing their applicability to the chosen case study. Also, this section includes two non-parametric tests, Mann–Kendall & Pettitt and Kling–Gupta efficiency.

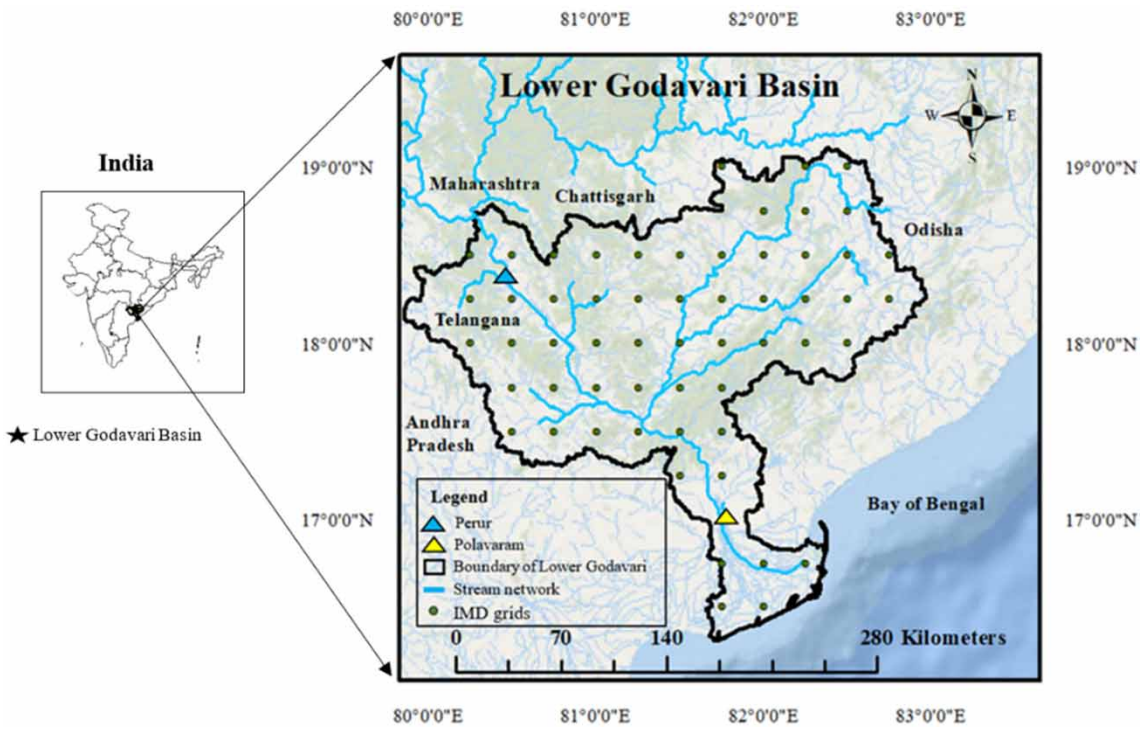


Figure 1 | Lower Godavari Basin showing 52 grids.

3.1. CNN and its fuzzy extension

CNN is based on the intuition of local neural connectivity inspired by the cognitive mechanism of the animal visual cortex (Lin & Jhang 2022). It uses convolution operation, which integrates input and filter matrix using dot product, enabling feature extraction. The architecture of CNN is presented in Figure 2(a). Firstly, the input layer provides the data to the model. The time and space-specific information from the past are used to map the output variable. Observations of input variables at distinct times form the elements of the matrix. The next layer in the workflow is the convolution layer (or feature extraction layer). It is the crucial layer used in feature extraction. It comprises a convoluted matrix, a product of filter and input matrices.

The next layer is the pooling layer, which is utilized to truncate the size of the convoluted matrix. The pooling operation truncates the number of learnable parameters by calculating each row's average value. It scales down the network and captures the most pertinent elements of the input layer. The pooled feature map is transformed into a one-dimension vector in a flattened layer. The next layer is the fully connected layer. Diverse features learned from the previous layers are transformed into a dense vector. In the final layer, the output is determined by the softmax function and is expressed as the following equation:

$$y_{p,i}^{c} = \frac{e^{z_i^c}}{\sum_{c'=1}^{N_{SM}} e^{z_i^{c'}}} \tag{1}$$

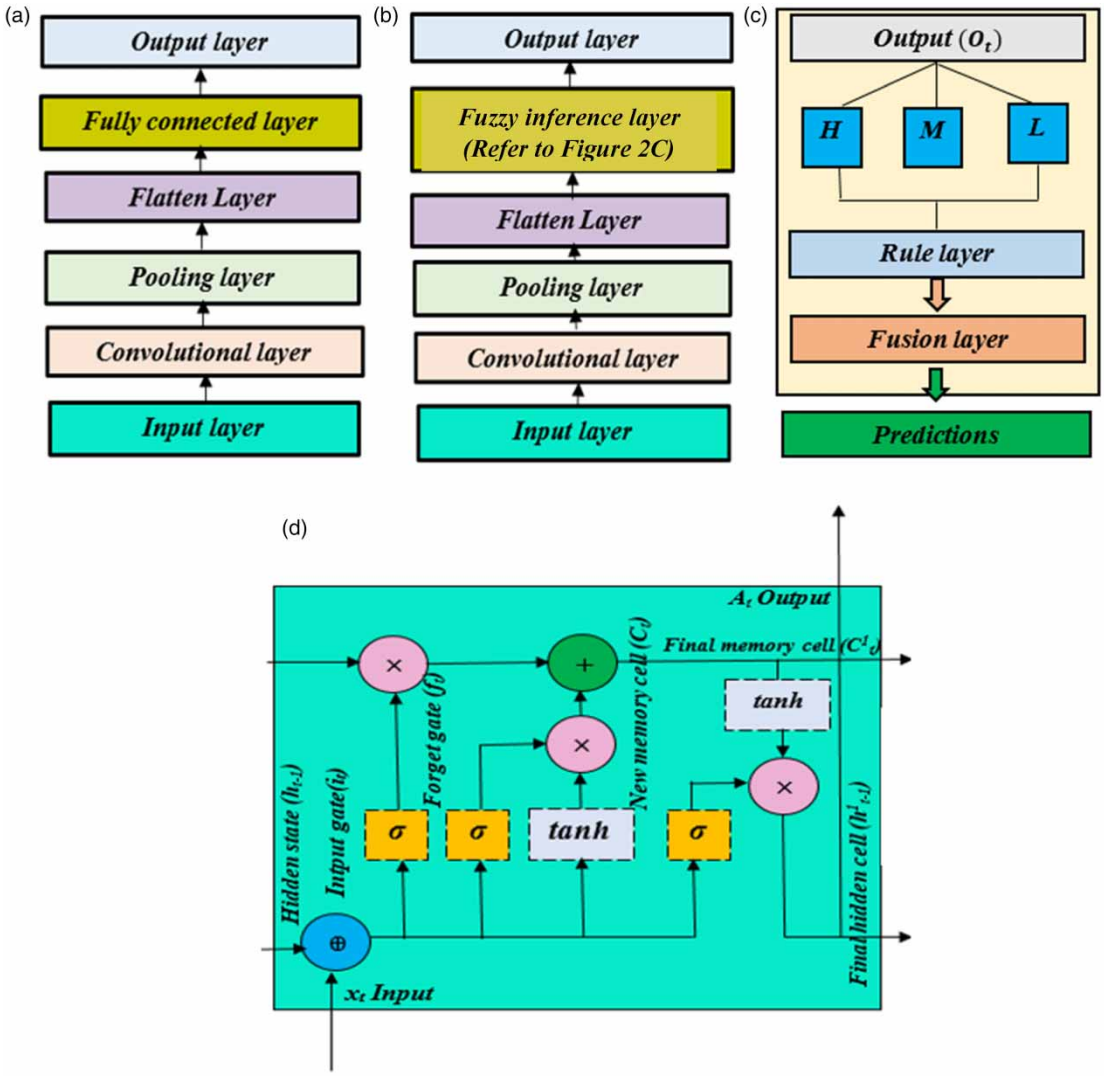


Figure 2 | Architecture of (a) CNN, (b) F-CNN, (c) Fuzzy inference layer, and (d) LSTM.

where $y^c_{p,i}$ is the output for p pooled features for cth channel element computed for ith data; N_{SM} is the number of softmax units and z^c_i and $z^{c'}_i$ are the elements of the dense fully connected layer. The loss function of cross-entropy error is computed between the observed and simulated data epochwise. The process is continued until the termination criterion is satisfied.

F-CNN (Lin & Jhang 2022) leverages the benefits of both CNN and fuzzy logic. Generally, a typical CNN architecture uses convolutional and pooling layers, which exhibit a top-notch ability in feature extraction. Further, these extracted features are summed up using a fully connected layer to obtain output. Contrastingly, F-CNN uses a fuzzy inference layer instead of a fully connected layer, which can sum up the fuzzy extracted features more remarkably. Figure 2(b) presents F-CNN architecture. Diverse features that were learned from the previous layers are discretized into three linguistic terms, high (H), medium (M), and low (L), using a symmetric Gaussian membership function due to its advantage over others (Hsu *et al.* 2020; Langeroudi *et al.* 2022) and are expressed as follows:

$$H = e^{-\frac{(x_i + me)^2}{2\sigma^2}} \tag{2}$$

$$M = e^{-\frac{x_i^2}{2\sigma^2}} \tag{3}$$

$$L = e^{-\frac{(x_i - me)^2}{2\sigma^2}} \tag{4}$$

where me and σ represent the mean and standard deviations of the given data. These fuzzified values of features are further used for creating knowledge-based rules to obtain better inferences. In the present study, the *itertools* function generates the fuzzy rules. Considering $x_i,\ i = 0, 1, 2 \ldots n$ as inputs and $y_j,\ j = 0, 1, 2 \ldots m$ as outputs, the lth fuzzy rule R^l can be expressed as follows:

$$R^l\text{:If } x_i \text{ is } S^l_i \ldots\ldots \text{ and } x_n \text{ is } S^l_n \text{ then } y_j \text{ is } w^l_1 \text{ and } y_m \text{ is } w^l_m \tag{5}$$

where S^l_i is the fuzzy set of the ith input and lth fuzzy rule. w^l_m represents the weight of the filter at lth fuzzy rule. The fuzzy output inferences q having feature maps p_f are computed as $\emptyset^{p_f}_q$ (Equation (6))

$$\emptyset^{p_f}_q = \prod_{i=1}^{h\times w} \mu_{F^q_i} \tag{6}$$

where $\mu_{F^q_i}$ represents the membership function for ith feature element. $h \times w$ represents the size of the final convolution layer, which contains $\boldsymbol{h}$ maps and $\boldsymbol{w}$ sets. Thus, $\mu_{F^q_i}(\xi^q_i)$ is presented as follows:

$$\mu_{F^q_i}(\xi^q_i) = \begin{bmatrix} H(\xi^q_{p_f,1}) & \times & H(\xi^q_{p_f,i}) \ldots\ldots & H(\xi^q_{p_f,h\times w}) \\ M(\xi^q_{p_f,1}) & \times & M(\xi^q_{p_f,i}) \ldots\ldots & M(\xi^q_{p_f,h\times w}) \\ L(\xi^q_{p_f,1}) & \times & L(\xi^q_{p_f,i}) \ldots\ldots & L(\xi^q_{p_f,h\times w}) \end{bmatrix} \tag{7}$$

where ξ^q_i represents the feature values after dropout in p_f. In the last layer, a total of $3^{h\times w}$ (Here, three refers to H, M, and L) number of inferences for $\boldsymbol{T}$ records is combined as follows:

$$z = w\lambda \tag{8}$$

$$\lambda = \begin{bmatrix} \emptyset^1_1 \\ \emptyset^1_2 \\ . \\ . \\ \emptyset^T_{3^{h\times w}} \end{bmatrix} w = \begin{bmatrix} w_{1,1} & \ldots\ldots & w_{1,3^{h\times w}} \\ w_{2,1} & \ldots\ldots & w_{2,3^{h\times w}} \\ w_{3,1} & \ldots\ldots & w_{3,3^{h\times w}} \end{bmatrix} \text{and } z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \tag{9}$$

Further softmax regression function is applied to obtain y (Equation (10))

$$\mathrm{y} = a = \frac{1}{e^{z_1} + e^{z_2} + e^{z_3}} \begin{bmatrix} e^{z_1} \\ e^{z_2} \\ e^{z_3} \end{bmatrix} \tag{10}$$

The output varies upon considering different activation functions ReLU, LeakyReLU, and tan$h$, and the corresponding output values are presented as follows:

$$y = \mathrm{ReLU}(z) = \mathrm{Max}(0, z) \tag{11}$$

$$y = \mathrm{LeakyReLU}(z) = \mathrm{Max}(0.01z, z) \tag{12}$$

$$y = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{13}$$

The backpropagation of the error is performed by updating the gradient values computed using the chain rule.

3.2. LSTM and its fuzzy extension

LSTMs are a unique RNN type that uses memory blocks as special storage units in their hidden layers (Hochreiter & Schmidhuber 1997). This information from the layers is navigated through gating units. These exhibit exceptional ability in capturing long-term dependencies from a large time series. These can overcome challenges such as oscillating weights and huge computation time issues posed by exploding and vanishing gradients. Figure 2(d) presents the LSTM architecture. The input layer receives input data and transmits its output through cell gating units and hidden layers. Three gate units regulate the flow of input information in cells (Sivakumar & Uyyala 2021). The new memory cell and final hidden cell retrieve short-term and long-term memory from useful information navigated from the three gating units. First, the input signal x_t passes through the forget gate, which removes unwanted information from the former state and the output of the upper hidden layer, h_{t-1}. The new information in the system will be accumulated in the input gate. The update gate plays a significant role in adding new elements, whereas the cell states are responsible for extracting meaningful information. The output gate controls the information propagation from the most recent cell to the final state. The following are the related equations:

$$\text{Forget gate}(\boldsymbol{f_t}) = \sigma(W_f x_t + A_f h_{t-1} + b_f) \tag{14}$$

$$\text{Input gate}(\boldsymbol{i_t}) = \sigma(W_i x_t + A_i h_{t-1} + b_i) \tag{15}$$

$$\text{New memory cell } (\boldsymbol{c_t^{'}}) = \tanh(W_c xt + A_c h_{t-1} + b_i) \tag{16}$$

$$\text{Final memory cell } (\boldsymbol{c_t}) = f_t c_{t-1} + i_t c_t' \tag{17}$$

$$\text{Final hidden cell } (\boldsymbol{h_t}) = o_t \tanh(c_t) \tag{18}$$

$$\text{Output gate } (\boldsymbol{o_t}) = \sigma(W_o x_t + A_o h_{t-1} + b_o) \tag{19}$$

where W_f, W_i, W_c, and W_o represent input weight vectors; A_f, A_i, A_c, and A_o represent output weight vectors; b_f, b_i, b_c and b_o represent the bias vectors; c'_{t-1} and c'_t represent new memory cells at times $t-1$ and t; c_t represents the final memory cell at time t; h_{t-1} and h_t represent the final hidden cell at times $t-1$ and t ; σ represents the sigmoid function for gating purposes while the function tan$h$ is used for storing the new memory cell c'_t to enable a faster convergence rate.

F-LSTM (Li *et al.* 2020; Langeroudi *et al.* 2022) is an architecture that combines fuzzy logic with LSTM. It comprises a similar architecture to that of LSTM. The information obtained from the output gate of the LSTM cell is further passed into the fuzzy inference layer, as presented in Figure 2(c) and 2(d). It is the crucial part of the architecture, which comprises the membership layer (β), rule layer ($\boldsymbol{\alpha}$), and fusion layer (F). The mathematical expression of the membership layer is similar to that of the F-CNN (Equations (2)–(4)). The fuzzified features obtained from the membership layer form a tensor layer (τ). The dimensions of the τ are represented by batch size and membership layer units. After forming τ, the rules are generated based on the features and the fuzzy sets considered. Fuzzy rules are generated using a multiplication layer for generating the fuzzy inferences. The $\emptyset_q^p$, $\mu_{F_i^q}(x_i^q)$, and $\boldsymbol{\alpha}$ are computed similarly to F-CNN (Equations (5) and (7–9)). After the formation of the

$\boldsymbol{\alpha}$, the last layer used in this architecture is the fusion layer. This layer primarily utilizes three operations: concatenation, linear transformation, and applications of activation functions for combining the information of the $\boldsymbol{\alpha}$ and β for producing the final fusion layer output. Firstly, β and $\boldsymbol{\alpha}$ are linked using the concatenation operation and are represented by $F(\beta, \boldsymbol{\alpha})$. This undergoes a linear transformation having weights (w_i) and biases (b_j) (Equation (20))

$$L_T = w_i F(\beta, \boldsymbol{\alpha}) + b_j \tag{20}$$

where L_T is the output obtained from the linear transformation. It is passed through ReLU, LeakyReLU, or tan$h$ activation functions to generate the outcome (Equations (11)–(13)).

3.3. C-LSTM and its fuzzy extension

C-LSTM is a hybrid algorithm that leverages the strengths of individual algorithms to improve simulation accuracy. CNN is exceptional at capturing high-dimensional spatial features from the raw data with the help of convolution filters. However, they are less capable of establishing long-term temporal dependencies. On the other hand, LSTM displays its dominance in capturing irregular trends from long-term temporal dependencies with the help of gating and memory units. They are, however, unsuitable for capturing high-dimensional spatial features from the data. In this regard, C-LSTM architectures can incorporate spatiotemporal aspects of the data by combining CNN and LSTM, making them ideal for regression analysis. The architecture of C-LSTM is presented in Figure 2(a) and 2(d). The C-LSTM workflow begins with the input layer. Data from the input layer are transmitted to subsequent layers to extract input features. Results are conveyed to the LSTM cell from the pooling layer. Gating units retrieve useful information. The new and hidden cell states update this information, and the final output is obtained.

FC-LSTM (Bao *et al.* 2022) combines the capabilities of CNN and LSTM in the context of fuzzy logic. It facilitates the production of precise projections while handling noisy data. FC-LSTM has a similar architecture to C-LSTM, incorporating a fuzzy inference layer instead of a fully connected one from CNN. The FC-LSTM architecture is presented in Figure 2(b) and 2(d).

The Mann–Kendall test is utilized to ascertain trends in long-term temporal data. The existence of a trend is considered a null hypothesis (H_o), which utilizes Kendall score (Z_S) (Wang *et al.* 2020). It is expressed as follows:

$$Z_S = \frac{S-1}{\sqrt{\mathrm{Var}(S)}} \tag{21}$$

where S represents the Kendall statistic, which is the sum of the signed differences across observations x_j and x_i at times j and i, respectively. Var(S) represents the difference between the total variance of n samples and the variance attributed to the concordant pairs. Sign of Z_S indicates the direction of the trend positive (or negative), i.e., increasing (or decreasing). p-value helps derive the hypothesis's validity at a certain significance level (α), and below this threshold is considered statistically significant. The Pettitt test (Li *et al.* 2014) detects a change point in a time series dataset. It is based on U-Statistic, which depends on signed differences. KGE analyzes the performance of the model and mathematical expression as follows (Mizukami *et al.* 2019) (Equation (22)):

$$\mathrm{KGE} = 1 - \left(\left(\frac{\bar{x}_S}{\bar{x}_O} - 1 \right)^2 + \left(\frac{\mathrm{SD}_S}{\mathrm{SD}_O} - 1 \right)^2 + (r-1)^2 \right)^{0.5} \tag{22}$$

$\bar{x}_O$ and SD_O are average and standard deviations related to the observed, whereas $\bar{x}_S$ and SD_S correspond to simulated data. r is the coefficient of correlation.

4. RESULTS

This section includes results related to hyperparameter tuning, scatter plots, and streamflow projection in climate change framework using four SSPs.

4.1. Hyperparameter tuning

The grid search hyperparameter tuning was employed. Table S1 (refer to the appendix section) presents the parameters of six machine learning (ML) algorithms. LSTM, F-LSTM, C-LSTM, and FC-LSTM have an optimal learning rate of 0.001, whereas 0.0001 for the remaining. The ReLU activation function is optimal for CNN, LSTM, and C-LSTM, and it is LeakyReLU for F-CNN and F-LSTM. It is tan*h* in case of FC-LSTM. Pooling layers were optimal at a value of 3 in the case of CNN and F-CNN, whereas it is 2 in the case of C-LSTM and FC-LSTM. The optimal value of the kernels was reduced by 50% in the case of C-LSTM and FC-LSTM compared to CNN and F-CNN, which signifies that the hybrid algorithm and their fuzzy extensions require relatively less effort for capturing the patterns effectively for a given data. The dropout rate was optimal at 0.3 in the case of F-LSTM, whereas it is 0.2 in the remaining algorithms, indicating moderate noisy data. Batch sizes were found to be high in the case of C-LSTM and FC-LSTM, having a value of 64, whereas it is 32 in the case of CNN, LSTM, and F-LSTM. F-CNN is optimal at 10 with relatively lower batch sizes than the other algorithms. Larger batch sizes would accelerate the computational speed. However, assigning more than the optimal values may lead to erroneous results. The number of neurons was high in the case of LSTM and F-LSTM, having a value of 128. However, considering neurons greater than optimal values would lead to overfitting. The number of epochs is optimal at 200 for F-CNN and 100 for the remaining algorithms. The default membership function chosen was Gaussian in the fuzzy extensions.

4.2. Scatter and discharge line plots in training and testing records

Figure 3(a) and 3(b) shows scatter plots for 374 training and 94 testing records, whereas Figure 4(a) and 4(b) presents corresponding discharge line plots. Related results for CNN, F-CNN, LSTM, F-LSTM, C-LSTM, and FC-LSTM (in the same order) are as follows.

In training, change *in the simulated maximum streamflow is* 16.99, −8.54, 24.99, 1.62, −35.17, and −1.37% compared to the maximum observed and occurred in August 1990 for all the algorithms. Change is negative for C-LSTM, F-CNN, and FC-LSTM and positive for the remaining. *Average simulated* is changed by 17, 7.03, 11.86, 4.45, −13.48, and 5.24% compared to the average observed streamflow. Change is negative only for C-LSTM. Variability focuses on the relative spread of streamflow magnitudes. In the present study, the estimated v*ariability* is 1.67, 1.48, 1.80, 1.56, 1.54, and 1.62. LSTM has shown high variability, whereas it is low for F-CNN. KGE values obtained are 0.76, 0.909, 0.75, 0.901, 0.78, and 0.913. Here, fuzzy extensions are performing exceptionally well, as per KGE.

In testing, the *range of simulated maximum streamflow* compared to the maximum observed is −0.73 to 6.98% (occurred in August 2013). *Average simulated values* compared to the observed streamflow are −19.99, 8.19, −0.09, 8.13, −0.04, and 3.06% compared to the average observed streamflow. *Variability is* 1.77, 1.48, 1.49, 1.60, 1.59, and 1.62. KGE values are 0.72, 0.84, 0.74, 0.86, 0.74, and 0.88.

The key factors contributing to enhancing the performance of fuzzy deep learning are mainly due to incorporating the fuzzy inference layer. It resulted in a better peak flow simulation of F-CNN, F-LSTM, FC-LSTM, i.e., −8.54, 1.62, and −1.37% compared to the observed. These values in CNN, LSTM, and C-LSTM are 16.99, 24.99, and −35.17% compared to the observed. This demonstrates the adaptability of fuzzy deep learning in handling real-time data where uncertainty prevails. Thus, it strongly justifies the usage of classical ones in this context.

Furthermore, the fusion layer concatenated the outputs of the rule layer and membership layer using a linear transformation. This synergistic combination has shown greater potential in generating robust predictions. As a result, the 27 fuzzy rules, with the assistance of 19,683 fuzzy inferences, aided in effectively synthesizing information from these distinct modalities, leading to improved performance. Based on the KGE, FC-LSTM performed better. It is superior to the selected algorithms because of its greater spatiotemporal capturing ability due to hybridization, exceptional feature extraction capability, robust fusion layer, and improved generalization.

Consequently, the FC-LSTM algorithm is further explored for streamflow projections in a climate change framework for four-time segments of 20 years each, i.e., 2021–2040, 2041–2060, 2061–2080, and 2081–2099.

4.3. Streamflow projections for 2021–2040, 2041–2060, 2061–2080 and 2061–2080

The streamflow projections are compared with the help of minimum, average, maximum, and interquartile regions (IQR) to understand the streamflow fluctuations related to 2021–2040 (Table 1). Salient observations are as follows.

Minimum streamflow: An increasing trend is observed in both SSP126 and SSP245 in all four-time segments. The lowest values are projected in SSP245 with magnitudes 378.31, 404.33, 458.14, and 477.55 m^3/s, respectively, whereas it has

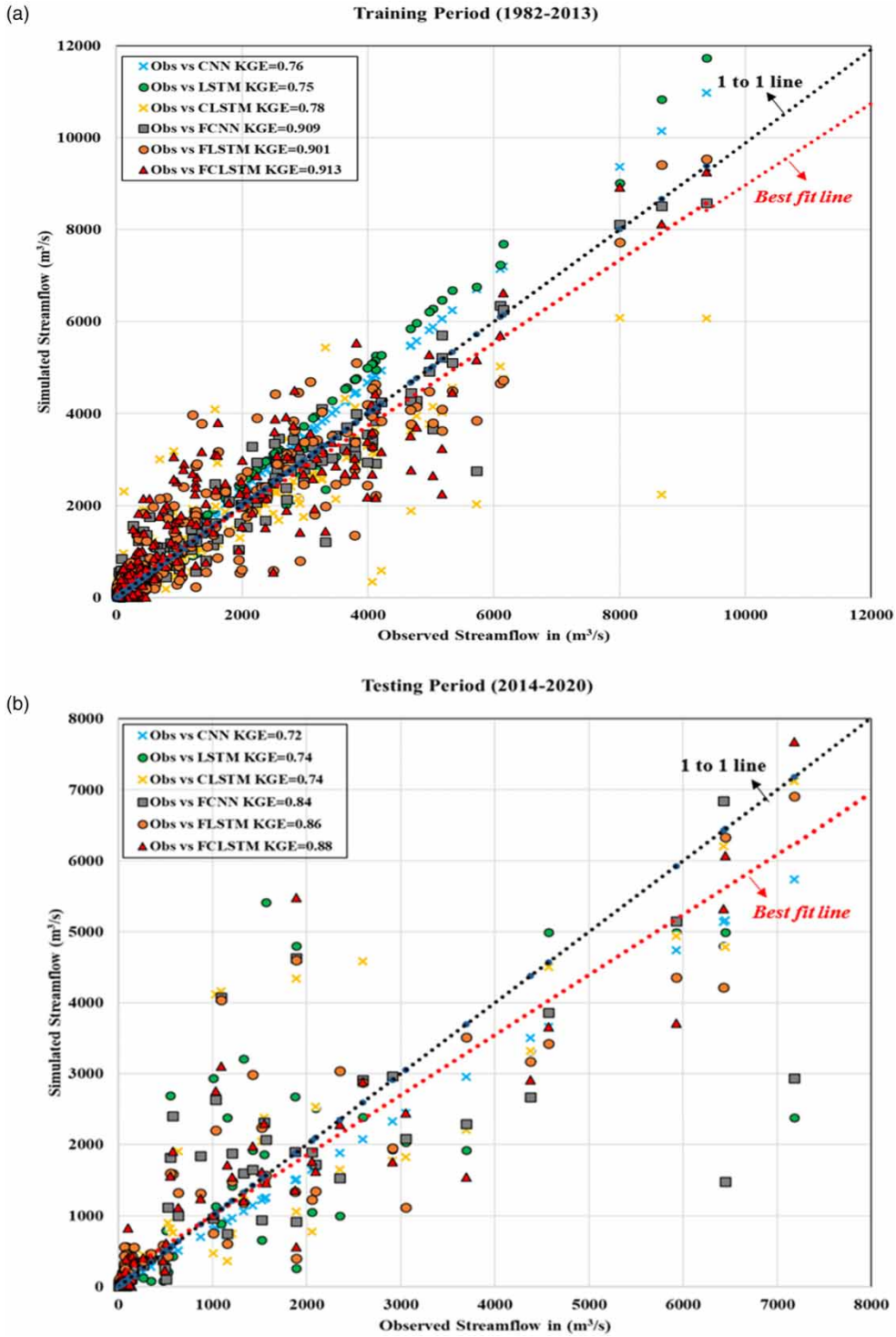


Figure 3 | Scatter plots in (a) training and (b) testing periods.

remained almost identical in SSP370 and SSP585 for all time segments. Higher values are projected in SSP370 in all four-time horizons with a magnitude of around 874 m^3/s.

Average streamflow: Similar inferences were drawn for average streamflow related to trends in SSP126 and SSP245 as observed for minimum streamflows. SSP370 and SSP585 are almost identical, with a marginal rise in streamflow values for time segments 2041–2060 in SSP370 and the same values in SSP585 for the 2041–2060, 2061–2080, and 2061–2080.

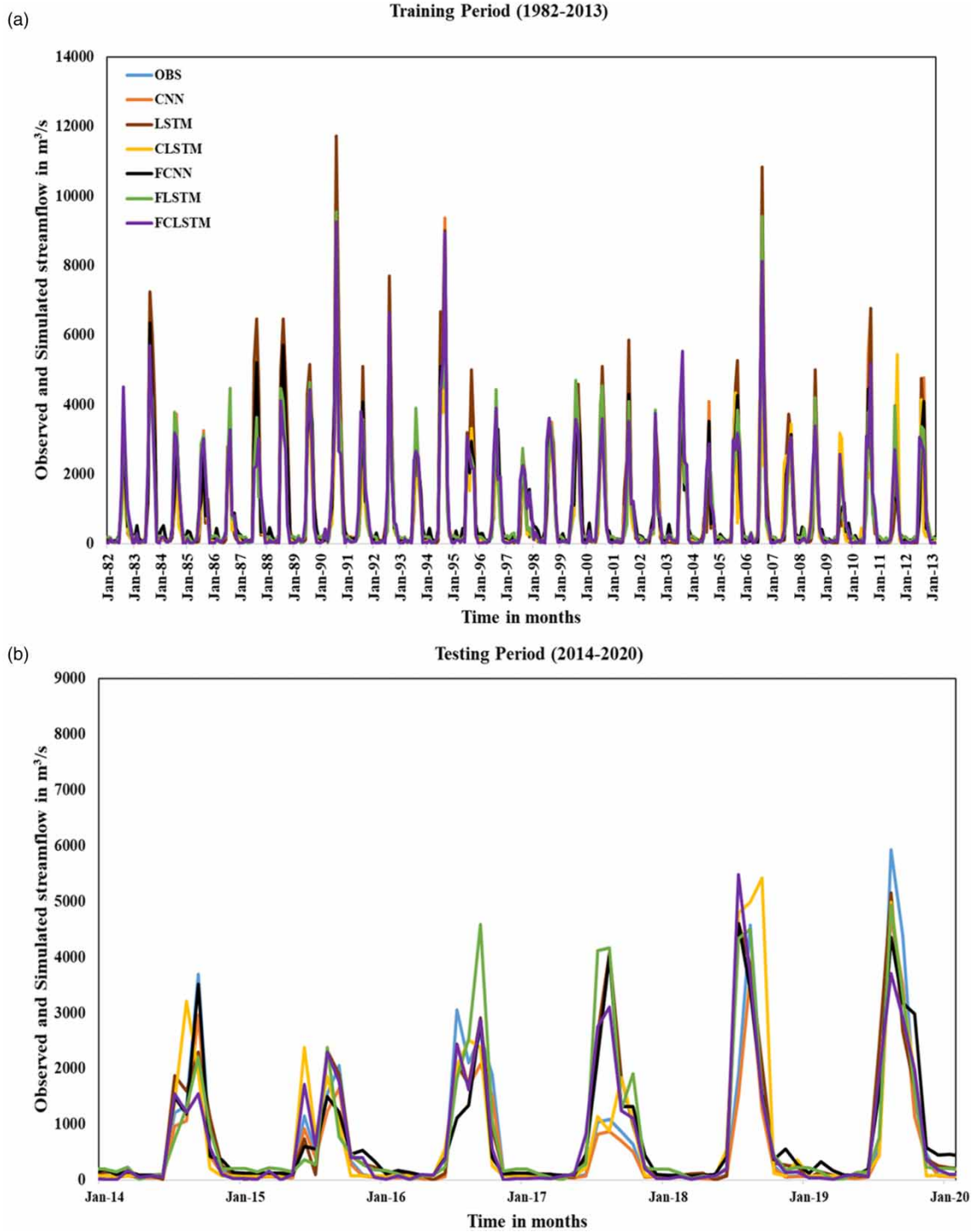


Figure 4 | Discharge line plots in (a) training and (b) testing periods.

Maximum streamflow: An increasing trend in all four-time segments in SSP126 is observed. However, the increasing trend is followed till 2061–2080 and remained identical in 2081–2099 for SSP245 and SSP370. SSP585 remained identical for 2021–2040 and 2041–2060, followed by an increase and decrease for 2061–2080 and 2081–2099. Overall, a maximum streamflow of 2,291.6 m^3/s is projected in SSP370 for 2041–2060.

Table 1 | Statistical information of projected streamflow in four SSPs with FC-LSTM

Time segment	SSP	Minimum (m^3/s)	Average (m^3/s)	Maximum (m^3/s)	IQR (m^3/s)
2021–2040	SSP126	482.79	682.85	850.35	161.13
	SSP245	378.31	669.67	917.51	307.10
	SSP370	873.55	877.58	912.82	3.91
	SSP585	865.98	873.72	884.34	9.03
2041–2060	SSP126	556.56	707.39	865.37	179.46
	SSP245	404.33	693.91	925.42	321.14
	SSP370	873.68	883.30	2,291.60	3.57
	SSP585	866.99	873.90	884.34	9.49
2061–2080	SSP126	570.52	736.82	874.23	176.84
	SSP245	458.14	725.12	927.97	309.03
	SSP370	874.13	877.47	880.01	3.71
	SSP585	866.99	879.88	2,281.05	9.41
2081–2099	SSP126	602.64	767.27	880.45	167.78
	SSP245	477.35	756.48	927.13	273.46
	SSP370	874.22	877.44	879.92	3.57
	SSP585	866.99	874.24	922.67	9.51

Figures 5 and 6(a)–6(d) display line and box plots (regarding *IQR*) for four SSPs corresponding to the projected discharge. SSP126 and SSP245 predicted an increase in streamflow for 2041–2060 compared to 2021–2040, followed by a decrease in 2061–2080 and a further decrease in 2081–2099. Lower IQR values of 3.91, 3.57, 3.71, and 3.57 m^3/s in SSP370 were observed in four-time segments, indicating a high probability that the basin will experience high streamflows for an extended period, posing challenges in basin planning and management.

Table 2 presents the Mann–Kendall and Pettitt test outputs for the projected streamflow of four-time segments and SSPs. Each 20-year segment, i.e., 2021–2040, 2041–2060, 2061–2080, comprises 240 records each, and 2081–2099 comprises 228, totaling 948 monthly projected streamflow records in each SSP.

Z_S in Mann–Kendall is observed to be positive in all the time segments, indicating an increasing trend. However, the value of p was less than 0.05 in SSP126 in the 2061–2080- and 2081–2099-time segments. This infers the existence of an increasing trend. In the case of Pettitt, all the change points were found in March in SSP126 and SSP245. The lowest and highest

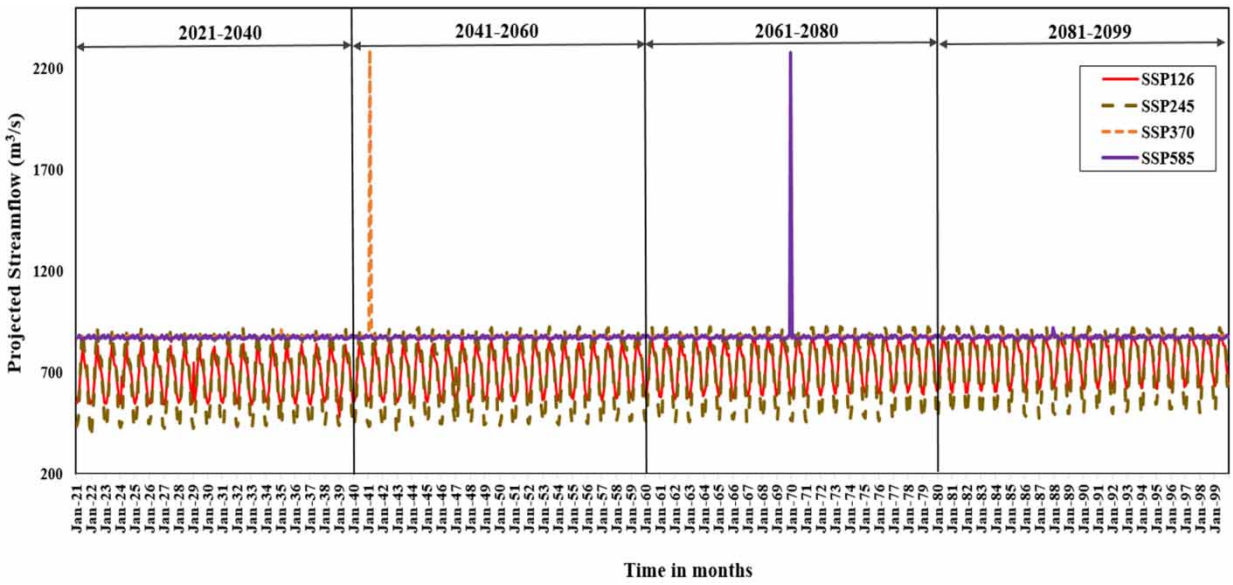


Figure 5 | Line plot of projected streamflow in 2021–2099.

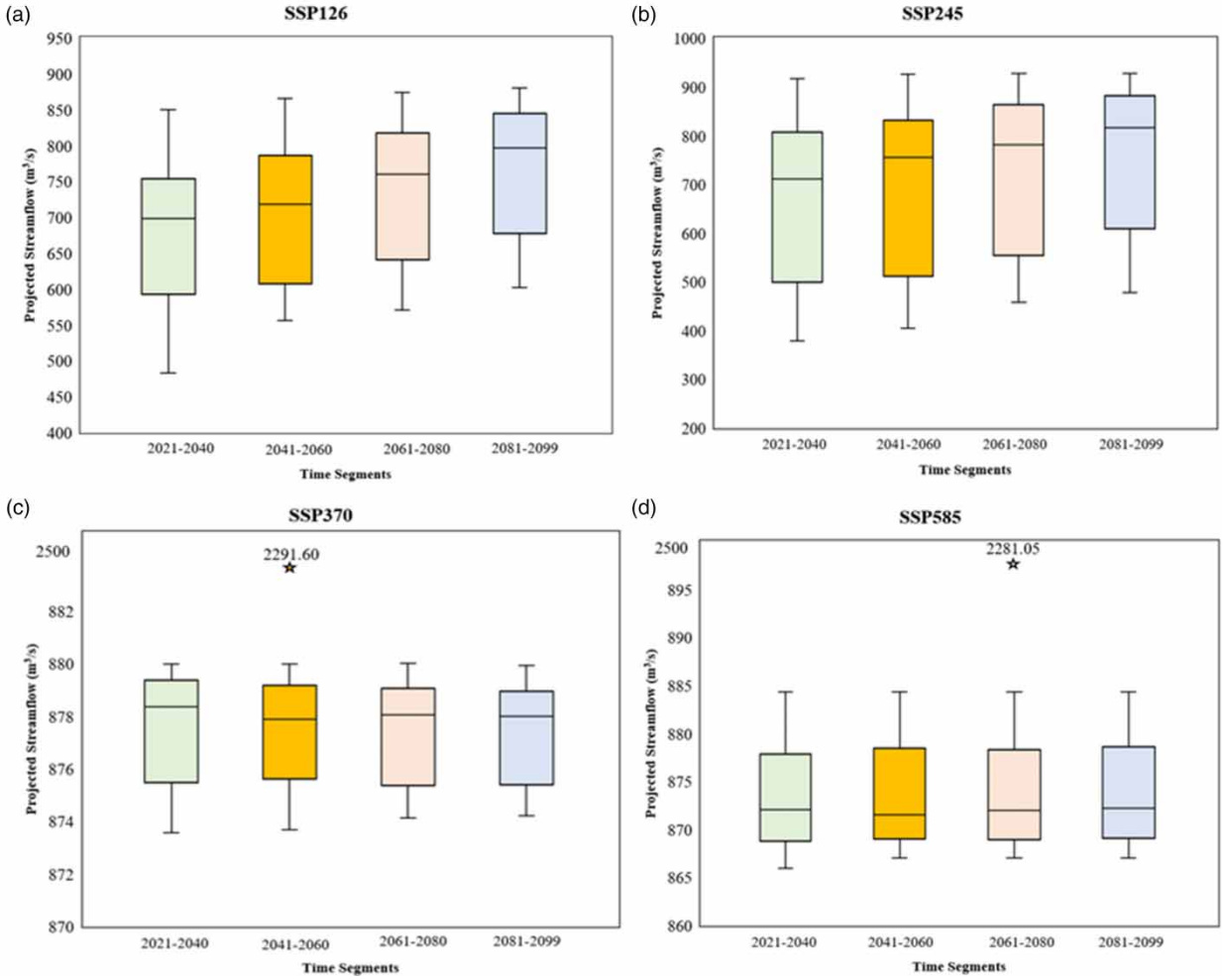


Figure 6 | Box plots of projected streamflow in (a) SSP126, (b) SSP245, (c) SSP370, and (d) SSP585.

Table 2 | Mann–Kendall and Pettitt test information for projected streamflow

		Mann–Kendall test			Pettitt test		
SSP	**Time segment**	**Z_S**	***p*-value**	**Trend**	**Change point (cp)**	**Statistic (*U*)**	***p*-value**
126	2021–2040	1.17	0.24	No trend	2028-March	1,471	0.68
	2041–2060	1.91	0.056	No trend	2052-March	2,135	0.24
	2061–2080	2.39	0.017	*Increasing*	2069-March	2,673	0.076
	2081–2099	2.15	0.032	*Increasing*	2091-March	2,245	0.13
245	2021–2040	0.86	0.39	No trend	2032-March	1,445	0.70
	2041–2060	1.43	0.154	No trend	2052-March	1,803	0.43
	2061–2080	1.43	0.154	No trend	2071-March	1,869	0.39
	2081–2099	1.085	0.278	No trend	2088-March	1,457	0.60
370	2021–2040	0.33	0.74	No trend	2032-April	602	1.0
	2041–2060	0.023	0.98	No trend	2041-December	560	1.0
	2061–2080	0.151	0.88	No trend	2066-December	732	0.99
	2081–2099	0.812	0.417	No trend	2087-June	1,126	0.87
585	2021–2040	0.03	0.98	No trend	2033-June	590	0.99
	2041–2060	0.34	0.73	No trend	2047-March	811	0.99
	2061–2080	0.29	0.77	No trend	2069-August	824	0.99
	2081–2099	0.23	0.82	No trend	2091-September	568	0.99

U-statistic values were observed in SSP370 and SSP126 for time segments 2041–2060 and 2061–2080, respectively. All the *p*-values identified were more significant than 0.05, indicating no abrupt shifts in the time series data.

5. DISCUSSION

The input data have the greatest impact on simulation accuracy, succeeded by training size, pre-processing techniques, and choice of algorithm, according to Moosavi *et al.* (2022). Among the four, the impact of hybridization and training size on algorithm efficacy are two notable factors that resonate with the present study. Marginal improvement of C-LSTM with a KGE value of 0.78 during training is observed compared to CNN and LSTM. Several studies (Ghimire *et al.* 2021; Deng *et al.* 2022; Wang *et al.* 2023) have drawn a similar conclusion that hybrid C-LSTM edges out standalone CNN and LSTM in streamflow simulation.

Boulmaiz *et al.* (2020) studied the impact of varying training sizes from 3 to 15 years on streamflow simulation across 21 catchments of the USA using feed forward neural network and LSTM. LSTM was effective even with 9 years of training and 3 years of testing data. Contrarily, Tounsi *et al.* (2023) found that a 55-year training size yielded an exceptional KGE value of 0.97 compared to a 25-year training size of 0.86 using LSTM while simulating streamflow across 438 USA watersheds. However, the classical deep learning algorithms have displayed KGE values greater than 0.75 in training and a satisfactory performance ranging from 0.5 to 0.75 in the testing period, even with 32-year streamflow records (Towner *et al.* 2019). Fuzzy extensions improved the KGE performance by 13–15% compared to classical ones. The influence of the train-test split ratio also impacts the efficacy to a certain extent.

The selection of hyperparameters is another crucial aspect influencing the efficacy. Deng *et al.* (2022) utilized CNN, LSTM, and C-LSTM to simulate runoff across the Feilaixia catchment in southeastern China and identified C-LSTM as superior. The similarity resonating with the present study was the grid search method. The optimal values identified for LSTM matched the epochs, batch size, activation function, and learning rate. However, it deviates in the case of the dropout rate and the other two algorithms. Ghimire *et al.* (2021) found C-LSTM to simulate streamflow better in the Brisbane River and Tewah Creek catchments of Australia. The two similarities matching the present study were the grid search method and ReLU activation function, which were identified as optimal for all the chosen algorithms. However, the study differs in the optimal values for batch sizes, epochs, and dropout rates. Batch sizes were optimal at 400, 400, and 500 for CNN, LSTM, and C-LSTM, respectively, which are more than 10 times that of the present study. However, optimal batch size values are identical in both CNN and LSTM. Epochs were optimal at 300, precisely three times that of the present study.

Interestingly, all the chosen algorithms were optimal at the same value in the present and previous studies. The dropout rate was optimal at half of the optimal value identified in the present paper. Kareem *et al.* (2021) found LSTM to be the most suitable for simulating the streamflow of the Piney watershed of the USA. The similarities identified from the study were the usage of the grid search method, similar optimal values of 100 for the epochs, and the ReLU. The dissimilarities were noticed in the optimal number of neurons (64) and dropout rate (0.1), which are 50% of that identified from the present study. Batch size was optimal at 128, whereas it is 100 in the present study.

Streamflow projections are vital in policy-making. However, few studies reported using deep learning algorithms in the CMIP6 framework. The present study projects the streamflow using FC-LSTM employing an ensemble of 13 GCMs and 4 SSPs. Several studies preferred using multiple GCMs and SSPs, similar to the present study, i.e., an ensemble of 11 GCMs for Yeongsan Basin, South Korea (Song *et al.* 2022), and 6 GCMs for Sutlej Basin, India (Singh *et al.* 2023) using SSP245 and SSP585.

Arathy *et al.* (2023) used an ensemble of three GCMs using three scenarios for the Greater Pamba River Basin, India, in the CMIP6 framework. Similar conclusions were drawn from Song *et al.* (2022), reporting a decrease in streamflow by 13.6% using LSTM for the 2050 and 2080s for most of the South Korean catchments, as observed in the present paper.

6. SUMMARY AND CONCLUSIONS

The present paper discusses the intricacies of exploring fuzzy-based deep learning algorithms F-CNN, F-LSTM, and FC-LSTM along with CNN, LSTM, and C-LSTM to simulate streamflows of Lower Godavari Basin, India. The meteorological and streamflow information is obtained from 1982 to 2020, i.e., 468 months. Among these, the training and testing percentage ratio considered was 80:20. Grid search technique was used for tuning the hyperparameters of six algorithms.

It is concluded from the study that incorporation of the fuzzy inference layer has strengthened the deep learning algorithms in terms of dealing with uncertainty, incorporating expert knowledge, and producing interpretable outputs, and significantly contributed to the improved performance of F-CNN (0.909, 0.84), F-LSTM (0.901,0.86), and FC-LSTM (0.913,0.88) over their classical counterparts CNN (0.76, 0.72), LSTM (0.75, 0.74), and C-LSTM (0.78, 0.74) in training and testing periods, respectively. FC-LSTM has shown superior simulation ability among the six algorithms and was used to project streamflows in future time segments. The projections showed that SSP370 had reported the highest maximum streamflow and average values of 2,291.6 and 883.30 m^3/s, respectively, in 2041–2060. SSP245 projects the lowest value in minimum streamflow of 378.31 m^3/s in 2021–2040. Interestingly, SSP370 has projected the lowest interquartile ranges across all four-time segments, indicating a high likelihood of prolonged streamflows, posing challenges to the planning and management of the basin.

Efforts were made to improve model efficacy by incorporating a fuzzy logic-based framework. The development of fuzzy-based CNN, LSTM, and C-LSTM can be applied to any field of engineering and management. However, there are some limitations in the present work and possible paths that can be explored by researchers, which are as follows.

We could not utilize K-fold cross-validation due to substantial computational demands, mainly when working with complex fuzzy hybridized algorithms. However, we consider this to be further work to understand the intricacies of K-fold with standard validation procedures. The present study explored the grid search algorithm alone for hyperparameter tuning of algorithms as an initiation, and further scope exists to employ metaheuristic algorithms. The study is limited to hybridizing CNN and LSTM in the fuzzy framework. Efforts can be made to hybridize more algorithms to understand their suitability.

In summary, we put a balanced emphasis on the model, which extensively focuses on applicability and replicability to hydrology and climate change, keeping sustainable development in view.

ACKNOWLEDGEMENTS

This research work is sponsored by CSIR, New Delhi, through Project No. 22(0782)/19/EMR-II dated 24.7.19. The authors thank the officials for their support.

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

Anshuman, A., Kunnath-Poovakka, A. & Eldho, T. I. 2021 Performance evaluation of conceptual rainfall-runoff models GR4J and AWBM. *ISH J. Hydraul. Eng.* **27** (4), 365–374.

Arathy, N. G. R. N., Shamsudeen, S. D., Mohan, M. G. & Sankaran, A. 2023 Basin-scale streamflow projections for greater Pamba River Basin, India integrating GCM ensemble modelling and flow accumulation-weighted LULC overlay in deep learning environment. *Sustainability* **15** (19), 14148.

Arsenault, R., Martel, J. L., Brunet, F., Brissette, F. & Mai, J. 2023 Continuous streamflow prediction in ungauged basins: Long short-term memory neural networks clearly outperform traditional hydrological models. *Hydrol. Earth Syst. Sci.* **27** (1), 139–157.

Bakhshi, O. F., Moradi, S., Asadi, A., Moghaddam Nia, A. & Taheri, S. 2023 Performance improvement of LSTM-based deep learning model for streamflow forecasting using Kalman filtering. *Water Resour. Manage.* **37** (8), 3111–3127.

Bao, R., Zhou, Y. & Jiang, W. 2022 FL-CNN-LSTM: Indoor air quality prediction using fuzzy logic and CNN-LSTM model. In: *2nd International Conference on Electrical Engineering and Control Science (IC2ECS)*, pp. 986–989. doi:10.1109/IC2ECS57645.2022.10088050.

Boulmaiz, T., Guermoui, M. & Boutaghane, H. 2020 Impact of training data size on the LSTM performances for rainfall–runoff modeling. *Model. Earth Syst. Environ.* **6**, 2153–2164.

Cheng, M., Fang, F., Kinouchi, T., Navon, I. M. & Pain, C. C. 2020 Long lead-time daily and monthly streamflow forecasting using machine learning methods. *J. Hydrol.* **590**, 125376.

de Campos Souza, P. V. 2020 Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Appl. Soft Comput.* **92**, 106275.

Deng, H., Chen, W. & Huang, G. 2022 Deep insight into daily runoff forecasting based on a CNN-LSTM model. *Nat. Hazards* **113**, 1675–1696.

Forghanparast, F. & Mohammadi, G. 2022 Using deep learning algorithms for intermittent streamflow prediction in the headwaters of the Colorado River, Texas. *Water* **14** (19), 2972.

Fu, G., Jin, Y., Sun, S., Yuan, Z. & Butler, D. 2022 The role of deep learning in urban water management: A critical review. *Water Res.* **223**, 118973.

Ghimire, S., Yaseen, Z. M., Farooque, A. A., Deo, R. C., Zhang, J. & Tao, X. 2021 Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks. *Sci. Rep.* **11** (1), 17497.

Ghobadi, F. & Kang, D. 2023 Application of machine learning in water resources management: A systematic literature review. *Water* **15** (4), 620.

Girihagama, L., Naveed Khaliq, M., Lamontagne, P., Perdikaris, J., Roy, R., Sushama, L. & Elshorbagy, A. 2022 Streamflow modelling and forecasting for Canadian watersheds using LSTM networks with attention mechanism. *Neural Comput. Appl.* **34** (22), 19995–20015.

Hsu, M. J., Chien, Y. H., Wang, W. Y. & Hsu, C. C. 2020 A convolutional fuzzy neural network architecture for object classification with small training database. *Int. J. Fuzzy Syst.* **22** (1), 1–10. https://doi.org/10.1162/neco.1997.9.8.1735.

Hochreiter, S. & Schmidhuber, J. 1997 Long short-term memory. *Neural Comput.* **9** (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Jhajharia, D., Gupta, S., Mirabbasi, R., Kumar, R. & Patle, G. T. 2021 Pan evaporative changes in transboundary Godavari River basin, India. *Theor. Appl. Climatol.* **145** (3–4), 1503–1520.

Kambalimath, S. & Deka, P. C. 2020 A basic review of fuzzy logic applications in hydrology and water resources. *Appl. Water Sci.* **10** (8), 1–14.

Kareem, K. Y., Seong, Y. & Jung, Y. 2021 LSTM prediction of streamflow during peak rainfall of Piney River. *J. Korean Soc. Dis. Sec.* **14** (4), 17–27.

Khosravi, K., Golkarian, A. & Tiefenbacher, J. P. 2022 Using optimized deep learning to predict daily streamflow: A comparison to common machine learning algorithms. *Water Resour. Manage.* **36** (2), 699–716.

Kim, C. & Kim, C. S. 2021 Comparison of the performance of a hydrologic model and a deep learning technique for rainfall-runoff analysis. *Trop. Cyclone Res. Rev.* **10** (4), 215–222.

Kumar, V., Kedam, N., Sharma, K. V., Mehta, D. J. & Caloiero, T. 2023 Advanced machine learning techniques to improve hydrological prediction: A comparative analysis of streamflow prediction models. *Water* **15** (14), 2572.

Langeroudi, M. K., Yamaghani, M. R. & Khodaparast, S. 2022 FD-LSTM: A fuzzy LSTM model for chaotic time-series prediction. *IEEE Intell. Syst.* **37** (4), 70–78.

Lapides, D. A., Sytsma, A. & Thompson, S. 2021 Implications of distinct methodological interpretations and runoff coefficient usage for rational method predictions. *J. Am. Water Resour. Assoc.* **57** (6), 859–874.

Li, J., Tan, S., Wei, Z., Chen, F. & Feng, P. 2014 A new method of change point detection using variable fuzzy sets under environmental change. *Water Resour. Manage.* **28**, 5125–5138.

Li, R., Hu, Y. & Liang, Q. 2020 T2F-LSTM method for long-term traffic volume prediction. *Int. J. Fuzzy Syst.* **28** (12), 3256–3264.

Li, J., Qian, K., Liu, Y., Yan, W., Yang, X., Luo, G. & Ma, X. 2022a LSTM-based model for predicting inland river runoff in arid region: A case study on Yarkant River, Northwest China. *Water* **14** (11), 1745.

Li, X., Xu, W., Ren, M., Jiang, Y. & Fu, G. 2022b Hybrid CNN-LSTM models for river flow prediction. *Water Supply* **22** (5), 4902–4919.

Liang, W., Chen, Y., Fang, G. & Kaldybayev, A. 2023 Machine learning method is an alternative for the hydrological model in an alpine catchment in the Tianshan region, Central Asia. *J. Hydrol. Reg. Stud.* **49**, 101492.

Lin, C. J. & Jhang, J. Y. 2022 Intelligent traffic-monitoring system based on YOLO and convolutional fuzzy neural networks. *IEEE Access* **10**, 14120–14133.

Manikanta, V. & Vema, V. K. 2022 Formulation of wavelet based multi-scale multi-objective performance evaluation (WMMPE) metric for improved calibration of hydrological models. *Water Resour. Res.* **58** (7), e2020WR029355.

Meher, J. K. & Das, L. 2019 Gridded data as a source of missing data replacement in station records. *J. Earth Syst. Sci.* **128**, 1–14.

Mishra, V., Bhatia, U. & Tiwari, A. D. 2020 Bias-corrected climate projections for South Asia from coupled model intercomparison project-6. *Sci. Data* **7** (1), 338.

Mizukami, N., Rakovec, O., Newman, A. J., Clark, M. P., Wood, A. W., Gupta, H. V. & Kumar, R. 2019 On the choice of calibration metrics for 'high-flow' estimation using hydrologic models. *Hydrol. Earth Syst. Sci.* **23** (6), 2601–2614.

Moosavi, V., Fard, Z. G. & Vafakhah, M. 2022 Which one is more important in daily runoff forecasting using data driven models: Input data, model type, preprocessing or data length? *J. Hydrol.* **606**, 127429.

Ng, K. W., Huang, Y. F., Koo, C. H., Chong, K. L., El-Shafie, A. & Ahmed, A. N. 2023 A review of hybrid deep learning applications for streamflow forecasting. *J. Hydrol.* **625** (B), 130141.

Nguyen, Q., Shrestha, S., Ghimire, S., Sundaram, S. M., Xue, W., Virdis, S. G. & Maharjan, M. 2023 Application of machine learning models in assessing the hydrological changes under climate change in the transboundary 3S River Basin. *J. Water Clim. Chang.* **14** (8), 2902–2918.

Ni, L., Wang, D., Singh, V. P., Wu, J., Wang, Y., Tao, Y. & Zhang, J. 2020 Streamflow and rainfall forecasting by two long short-term memory-based models. *J. Hydrol.* **583**, 124296.

Parisouj, P., Mohebzadeh, H. & Lee, T. 2020 Employing machine learning algorithms for streamflow prediction: A case study of four river basins with different climatic zones in the United States. *Water Resour. Manage.* **34**, 4113–4131.

Polavaram Project Authority Annual Report 2020–21 Ministry of Jal Shakti, Department of Water Resources, River Development & Ganga Rejuvenation, Government of India.

Rosli, M. H., Malik, N. K. A., Jamil, N. R., Kamarudin, M. K. A. & Maulud, K. N. A. 2022 Performance evaluation of spatial lumped model and spatial distributed travel time model using event-based rainfall for hydrological simulation. *Arabian J. Geosci.* **15** (24), 1765.

Roushangar, K. & Abdelzad, S. 2023 River flow modeling in semi-arid and humid regions using an integrated method based on LARS-WG and LSTM models. *Water Resour. Manage.* **37**, 3813–3831.

Sabitha, N. M., Thampi, S. G. & Kumar, D. S. 2023 Application of a distributed hydrologic model to assess the impact of climate and land-use change on surface runoff from a small urbanizing watershed. *Water Resour. Manage.* **37** (6–7), 2347–2368.

Sabzipour, B., Arsenault, R., Troin, M., Martel, J. L. & Brissette, F. 2023 Sensitivity analysis of the hyperparameters of an ensemble Kalman filter application on a semi-distributed hydrological model for streamflow forecasting. *J. Hydrol.* **626**, 130251.

Sarkar, S. 2022 Drought and flood dynamics of Godavari basin, India: A geospatial perspective. *Arab. J. Geosci.* **15** (8), 772.

Shekar, P. R., Mathew, A., Pandey, A. & Bhosale, A. 2023 A comparison of the performance of SWAT and artificial intelligence models for monthly rainfall–runoff analysis in the Peddavagu River Basin, India. *Aqua Water Infrastruct. Ecosyst. Soc.* **72** (9), 1707–1730.

Shen, C. 2018 A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resour. Res.* **54** (11), 8558–8593.

Sheng, Z., Wen, S., Feng, Z. K., Gong, J., Shi, K., Guo, Z. & Huang, T. 2023 A survey on data-driven runoff forecasting models based on neural networks. *IEEE Trans. Emerg. Topics Comput.* **7** (4), 1083–1097.

Shu, X., Peng, Y., Ding, W., Wang, Z. & Wu, J. 2022 Multi-step-ahead monthly streamflow forecasting using convolutional neural networks. *Water Resour. Manage.* **36** (11), 3949–3964.

Sidle, R. C. 2021 Strategies for smarter catchment hydrology models: Incorporating scaling and better process representation. *Geosci. Lett.* **8** (1), 24.

Singh, D., Vardhan, M., Sahu, R., Chatterjee, D., Chauhan, P. & Liu, S. 2023 Machine-learning-and deep-learning-based streamflow prediction in a hilly catchment for future scenarios using CMIP6 GCM data. *Hydrol. Earth Syst. Sci.* **27** (5), 1047–1075.

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y. & Demir, I. 2020 A comprehensive review of deep learning applications in hydrology and water resources. *Water Sci. Technol.* **82** (12), 2635–2670.

Sivakumar, M. & Uyyala, S. R. 2021 Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic. *Int. J. Data Sci. Anal.* **12** (4), 355–367.

Song, C. M. 2022 Data construction methodology for convolution neural network based daily runoff prediction and assessment of its applicability. *J. Hydrol.* **605** (7), 127324.

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. *Sci. Total Environ.* **838** (3), 156162.

Syed, S., Syed, Z., Mahmood, P., Haider, S., Khan, F., Syed, M. T. & Syed, S. 2023 Application of coupling machine learning techniques and linear bias scaling for optimizing 10-daily flow simulations in the Swat River basin. *Water Pract. Technol.* **18** (6), 1343–1356.

Tounsi, A., Abdelkader, M. & Temimi, M. 2023 Assessing the simulation of streamflow with the LSTM model across the continental United States using the MOPEX dataset. *Neural Comput. Appl.* **35**, 22469–22486.

Towner, J., Cloke, H. L., Zsoter, E., Flamig, Z., Hoch, J. M., Bazo, J. & Stephens, E. M. 2019 Assessing the performance of global hydrological models for capturing peak river flows in the Amazon basin. *Hydrol. Earth Syst. Sci.* **23** (7), 3057–3080.

Tripathy, K. P. & Mishra, A. K. 2024 Deep learning in hydrology and water resources disciplines: Concepts, methods, applications, and research directions. *J. Hydrol.* **628**, 130458.

van Vuuren, D. P., Riahi, K., Calvin, K., Dellink, R., Emmerling, J., Fujimori, S. & O'Neill, B. 2017 The shared socioeconomic pathways: Trajectories for human development and global environmental change. *Global Environ. Change* **42**, 148–152.

Vatanchi, S. M., Etemadfard, H., Maghrebi, M. F. & Shad, R. 2023 A comparative study on forecasting of long-term daily streamflow using ANN, ANFIS, BiLSTM and CNN-GRU-LSTM. *Water Resour. Manage.* **37** (12), 4769–4785.

Wang, F., Shao, W., Yu, H., Kan, G., He, X., Zhang, D. & Wang, G. 2020 Re-evaluation of the power of the Mann Kendall test for detecting monotonic trends in hydrometeorological time series. *Front. Earth Sci.* **8**, 14.

Wang, Y., Liu, J., Xu, L., Yu, F. & Zhang, S. 2023 Streamflow simulation with high-resolution WRF input variables based on the CNN-LSTM hybrid model and gamma test. *Water* **15** (7), 1422.

Wegayehu, E. B. & Muluneh, F. B. 2022 Short-term daily univariate streamflow forecasting using deep learning models. *Adv. Meteorol.* **2022**, 1860460.

Xu, T. & Liang, F. 2021 Machine learning for hydrologic sciences: An introductory overview. *Wiley Interdiscip. Rev.: Water* **8** (5), e1533.

Xu, W., Jiang, Y., Zhang, X., Li, Y., Zhang, R. & Fu, G. 2020 Using long short-term memory networks for river flow prediction. *Hydrol. Res.* **51** (6), 1358–1376.

First received 2 October 2023; accepted in revised form 28 January 2024. Available online 10 February 2024