Hindawi
Advances in Meteorology
Volume 2022, Article ID 1860460, 21 pages
https://doi.org/10.1155/2022/1860460

Research Article

Short-Term Daily Univariate Streamflow Forecasting Using Deep Learning Models

Eyob Betru Wegayehu and Fiseha Behulu Muluneh

School of Civil and Environmental Engineering, Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa, Ethiopia

Correspondence should be addressed to Eyob Betru Wegayehu; eyob.betru@aait.edu.et

Received 9 December 2021; Revised 25 January 2022; Accepted 27 January 2022; Published 10 February 2022

Academic Editor: Upaka Rathnayake

Hydrological forecasting is one of the key research areas in hydrology. Innovative forecasting tools will reform water resources management systems, flood early warning mechanisms, and agricultural and hydropower management schemes. Hence, in this study, we compared Stacked Long Short-Term Memory (S-LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), and Gated Recurrent Unit (GRU) with the classical Multilayer Perceptron (MLP) network for one-step daily streamflow forecasting. The analysis used daily time series data collected from Borkena (in Awash river basin) and Gummera (in Abay river basin) streamflow stations. All data sets passed through rigorous quality control processes, and null values were filled using linear interpolation. A partial autocorrelation was also applied to select the appropriate time lag for input series generation. Then, the data is split into training and testing datasets using a ratio of 80 : 20, respectively. Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2) were used to evaluate the performance of the proposed models. Finally, the findings are summarized in model variability, lag time variability, and time series characteristic themes. As a result, time series characteristics (climatic variability) had a more significant impact on streamflow forecasting performance than input lagged time steps and deep learning model architecture variations. Thus, Borkena's river catchment forecasting result is more accurate than Gummera's catchment forecasting result, with RMSE, MAE, MAPE, and R^2 values ranging between (0.81 to 1.53, 0.29 to 0.96, 0.16 to 1.72, 0.96 to 0.99) and (17.43 to 17.99, 7.76 to 10.54, 0.16 to 1.03, 0.89 to 0.90) for both catchments, respectively. Although the performance is dependent on lag time variations, MLP and GRU outperform S-LSTM and Bi-LSTM on a nearly equal basis.

1. Introduction

The science of streamflow forecasting is still one of the crucial research topics in hydrology. Accurate and reliable streamflow forecasting is vital for water resources planning, management, and disaster mitigation response authorities [1]. Streamflow forecasting can be classified into two. Firstly, short-term or real-time forecasting includes daily and hourly timestamps, widely applicable in flood management systems. Secondly, long-term forecasting usually contains weekly, monthly, and annual streamflow forecasting, crucial for reservoir operation, irrigation system management, and hydropower generation [2].

Generally, streamflow forecasting models can also be categorized into conceptual, physical-based, and data-driven models [3]. Conceptual models are lumped in nature and typically rely on empirical relationships among various hydrological variables. Due to the model's reliance on observed data, it is rarely applicable to data-scarce catchments. Hydrological processes can also be represented in physical models through mass, momentum, and energy conservation equations. These models may account for spatial variability, but since they are distributed in nature, they require a large amount of data on land use, slope, soil, and climate [4]. Lastly, data-driven models form a nonlinear input-output relationship without physical catchment information and minimum data requirements. Hence, the popularity of data-driven models is exploding these days with the advancement of computational capability and data set availability [5].

Zhang et al. [6] classified the data-driven approach as conventional and Artificial Intelligence (AI) based models. Conventional data-driven models such as Auto Regressive Moving Average with the exogenous term (ARMAX), Multiple Linear Regressive (MLR), and Auto-Regressive Integrated Moving Average (ARIMA) are easy to implement [6]. However, the nonlinearity of hydrological processes was left out of the equations. On the other hand, AI-based data-driven models can detect nonlinearity and perform better in streamflow forecasting. As a result, machine learning models have become a hot research topic these days.

AI-based data-driven streamflow forecasting models can be univariate when the model's input and output are designed with a single time series variable. Univariate forecasting models are straightforward to train using sparse data and provide ease of inference when evaluating forecast performance. Due to the complexity of agrometeorological data, it is simpler and more efficient to forecast the variables individually [7]. On the other hand, multivariate models are designed with multiple variables such as precipitation, temperature, evaporation, and other variables as input and a streamflow variable as output [6]. Thus, in data-scarce catchments with a limited amount of data, the application of univariate modelling is more feasible and has received wide attention in recent years [3, 6, 8, 9].

Wide variates of classical and deep learning models are present in the literature for time series forecasting, which includes Artificial Neural Network (ANN), Support Vector Machine (SVM), Fuzzy Logic (FL), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Genetic Programming (GP). However, because of the nonlinearity present in streamflow time series, the forecasting performance of these models is usually debatable [3, 10]. Under one-step and multistep-ahead forecast scenarios, Suradhaniwar et al. [7] compared the performance of Machine Learning (ML) and Deep Learning (DL) based time series forecasting algorithms. They also evaluated recursive one-step forward forecasts using walk-forward validation. Finally, Seasonal Auto-Regressive Integrated Moving Average (SARIMA) and Support Vector Regression (SVR) models outperform their DL-based counterparts: Neural Network (NN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) with fixed forecast horizons.

ANN (MLP) is the most widely used classical machine learning architecture in hydrology [11]. Cui et al. [12] demonstrated that when used for hourly river flow forecasting, the new generation of ANN or Emotional Neural Network (ENN) models outperformed the Multivariate Adaptive Regression Splines (Mars), Minimax Probability Machine Regression (MPMR), and Relevance Vector Machine (RVM) models. Yaseen et al. [2] also conducted a detailed review of literature from high impacted journals in the time frame of 2000–2015 on the state-of-the-art application of Artificial Neural Network (ANN), Support Vector Machine (SVM), Fuzzy Logic (FL), Evolutionary Computation (EC), and Wavelet-Artificial Intelligence (W-AI) for streamflow forecasting. The same study was concluded by stating that time series preprocessing, input variable selections, and time scale choices are the critical parameters for high-performing forecasting models.

RNN is the popular type of deep learning architecture that is optimized for time series analysis. However, it has drawbacks, such as vanishing and exploding gradients. Hochreiter and Schmidhuber [13] introduced the improved RNN version or LSTM, a popular time series model for longtime step forecasting. Recently, various fields of study have been experimenting with these models [14–20]. Moreover, Cho et al. [21] firstly introduced GRU as a simplified version of the LSTM model. GRU merges short-term and long-term memory cells into a single gate with reasonably good performance and fast running time [22]. Lara Benítez et al. [23] evaluated the accuracy and efficiency of seven popular deep learning architectures: Multilayer Perceptron (MLP), Elman Recurrent Neural Network (ERNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Echo State Network (ESN), Convolutional Neural Network (CNN), and Temporal Convolutional Network (TCN). Additionally, they constructed over 38000 distinct models to search for optimal architecture configuration and training hyperparameters, with LSTM achieving the lowest weighted absolute percentage error followed by GRU.

Even though we found multiple effective forecasting models with GRU and LSTM in different fields, specifically in hydrology, the accuracy of these models must be further fine-tuned with different data processing techniques and data input variations [24–27]. We can restructure the previous time step data series as a predictor input variable for univariate time series forecasting and the current or next step as the output variable. However, the number of input time steps selected is difficult to decide without prior knowledge. Hence, studying the effect of lagged time selection in streamflow forecasting is mandatory for to obtain accurate models. Lagged variables in univariate streamflow forecasting are significant factors that vary the model's performance positively or negatively and hold temporal dependency information as a predictor variable [28]. Tyralis and Papacharalampous [29] concluded that using a low number of recent predictor variables achieves higher accuracy for time series forecasting using Random Forest (RF) algorithm. It is vital to expand this finding to other popular deep learning models and to different climatic conditions.

Papacharalampous et al. [30] tested 20 one-step ahead univariate time series forecasting methods with extensive time series data. They concluded the study by suggesting that the most and least accurate approaches for one-step-ahead forecasting, such as the importance of time series length on the performance of various forecasting methods, are well addressed. Furthermore, the same study underlined that the machine learning model's optimization heavily relies on hyperparameter optimization and lagged variable selection. Torres et al. [31] also identified research gaps in various fields by analyzing the most successful deep learning architectures that predict time series effectively and highlighting MLP, RNN, GRU, and Bi-LSTM architectures in particular.

In the present study, we compared the different forms of LSTM architectures, S-LSTM, Bi-LSTM, and GRU, with the classical MLP network to forecast a single step streamflow amount with the available records of daily streamflow data. To the best of our knowledge, the LSTM has been mainly studied for monthly multivariate time series and not for daily univariate streamflow forecasting. Even though machine learning can model hydrological forecasting efficiently, researchers should further examine the impact of suitable input variables and model parameters selection on model accuracy very carefully [32].

2. Study Area and Data

Two river basin subcatchments in Ethiopia were selected for this study: (a) Gummera subcatchment in Abay River basin (Figure 1(a)), (b) Borkena subcatchment in Awash River basin (Figure 1(b)).

2.1. Borkena Catchment (Awash River Basin/Ethiopia). Borkena River originates at Kutaber woreda or the conjunction of Abay and Awash River basins (Ethiopia). The Awash River basin is usually classified into three main catchments, Lower Awash, Middle Awash, and Upper Awash. Borkena River is found in Lower Awash with its different tributaries, including Berbera River, Arawutie River, Abasharew/Wuranie River, Abba Abdela/Desso River, Worka River, and Leyole River. The total length of this river is estimated at around 165 km. Borkena River catchment hosts major cities, including Kombolcha, Dessie, and Kemissie. The study area streamflow outlet is at Kombolch station, and the catchment covers an area of 1709 km^2, bounded from 39° 30′E to 40° 0′E to 10° 15′N to 11° 30′N. Moreover, the area's elevation varies from 1775 m to 2638 m above sea level. The rainfall pattern of this catchment is unimodal, where 84% of the rainfall is happening from July to September [33].

2.2. Gummera Catchment (Abay River Basin/Ethiopia). The second case study area is the Gummara subbasin, one of the main tributaries of Lake Tana in the Abay River basin. The lake is located in the north-western highlands at 12° 00′N and 37° 15′E and collects runoff from more than 40 rivers. The lake receives water from several major rivers, including Gilgel Abay in the south, Megech River in the north, and Ribb and Gummara in the east. Small river streams from the lake's western side drain into the lake. Gummara River originates from the Guna mountains southeast of Debre Tabor at an altitude of approximately 3250 m.a.s.l. The Gummara catchment covers a total area of about 1592 km^2. Many small intermittent and perennial rivers and springs flow into the mainstream Gummara River. The catchment's topography is undulating, ranging from 1788 m.a.s.l. to 3750 m.a.s.l.

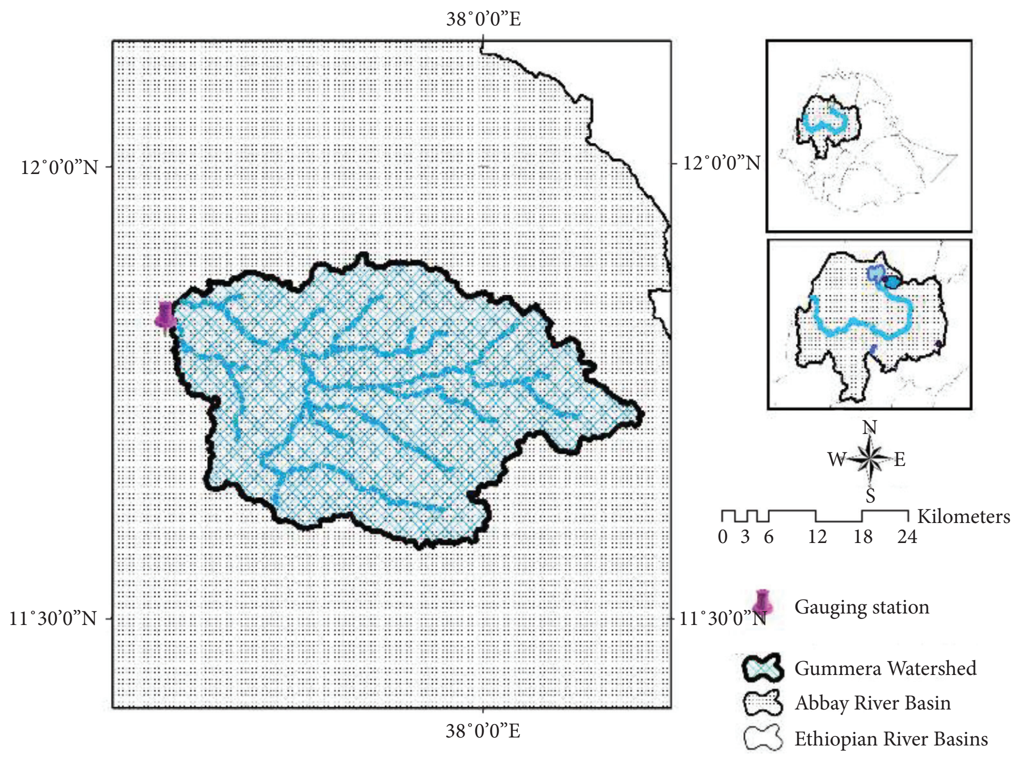
2.3. Data. Daily streamflow time series of two hydrometrological stations were collected from Ethiopia's Ministry of Water, Irrigation, and Energy (MoWIE). Then, these data were used to forecast single-step streamflow. At Borkena station, 6575 available streamflow data series were collected over the time window of January 1, 1972, to December 31, 1989. Similarly, at Gummera station, 9496 streamflow time series values from January 1, 1981, to December 31, 2006, were collected. A total of 866 null values are examined in the time series (i.e., 658 at Borkena and 208 at Gummara). The options for filling these gaps range from simple interpolation to complex statistical methods [34]. The method chosen depends on the length and season of missing data, availability of hydrometeorological data, climatic region, and length of previous observations. The sample mean or subgroup mean can be used to fill in missing values in daily streamflow data. However, replacing missing values with sample means may cause underestimating variance and incorrect subgroup identification [35]. Instead, the linear interpolation method is quick and straightforward to use, and it may be sufficient for data with a small gap [36]. Thus, we implemented linear interpolation in this study, and since the majority of the null values were stacked at low flows, the interpolation is acceptable [37–40].

After passing these rigorous quality control processes, the raw data were then split into training and testing datasets using a ratio of 80 : 20, respectively. Accordingly, different sizes of single overlapping step sliding windows were used to rebuild the input time series into a supervised learning format. The subsets were further standardized using a standard scalar approach. Figure 2 shows the descriptive statistics and the corresponding plots of split data for both catchments.

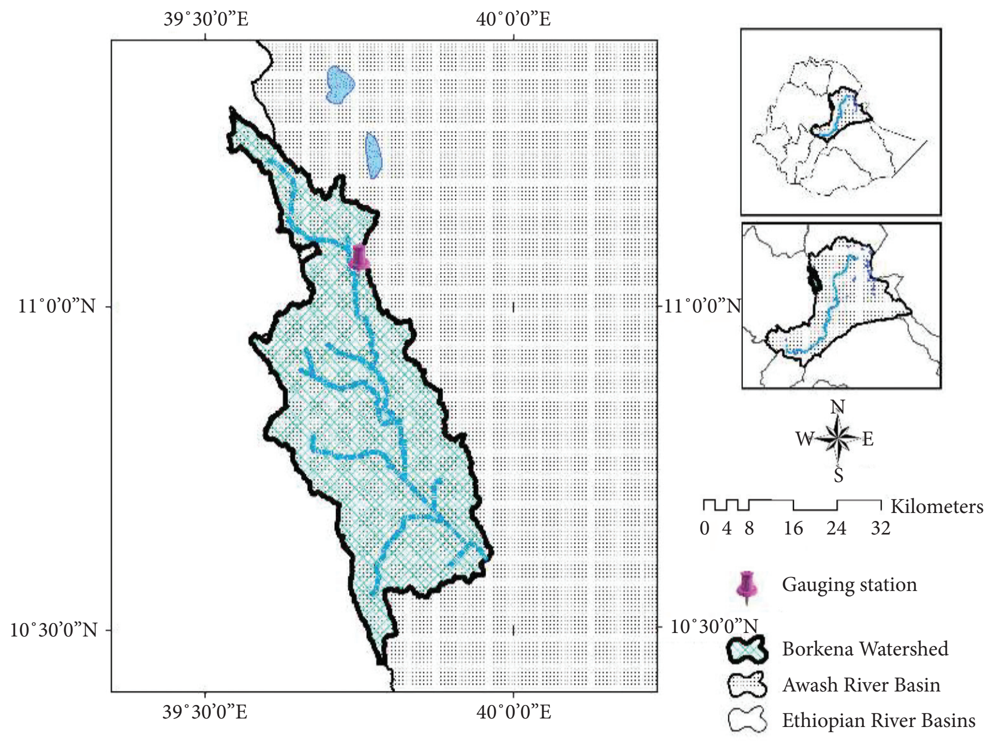
3. Methods

This study compared three types of recurrent network architectures (GRU, Bi-LSTM, and S-LSTM) with the classical neural network architecture (MLP).

3.1. Deep Learning Models. Deep learning models are usually distinguished from nondeep machine learning models by the depth of the network layer or by the number of stacked neuron layers and designed architectures. Nondeep learning models usually do not accurately learn the advanced nonlinearity present in the input and output variables [41]. In contrast, deep learning models are widely applied in different tasks, including processing, analyzing, designing, estimating, filtering, and detection tasks [42]. The popular deep learning models applied in different fields of studies are Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Radial Basis Function Networks (RBFN), and Generative Adversarial Network (GAN) [21, 25, 43–46]. The time series analysis models used in this study were specifically chosen, and they are briefly discussed in detail in the following sections.



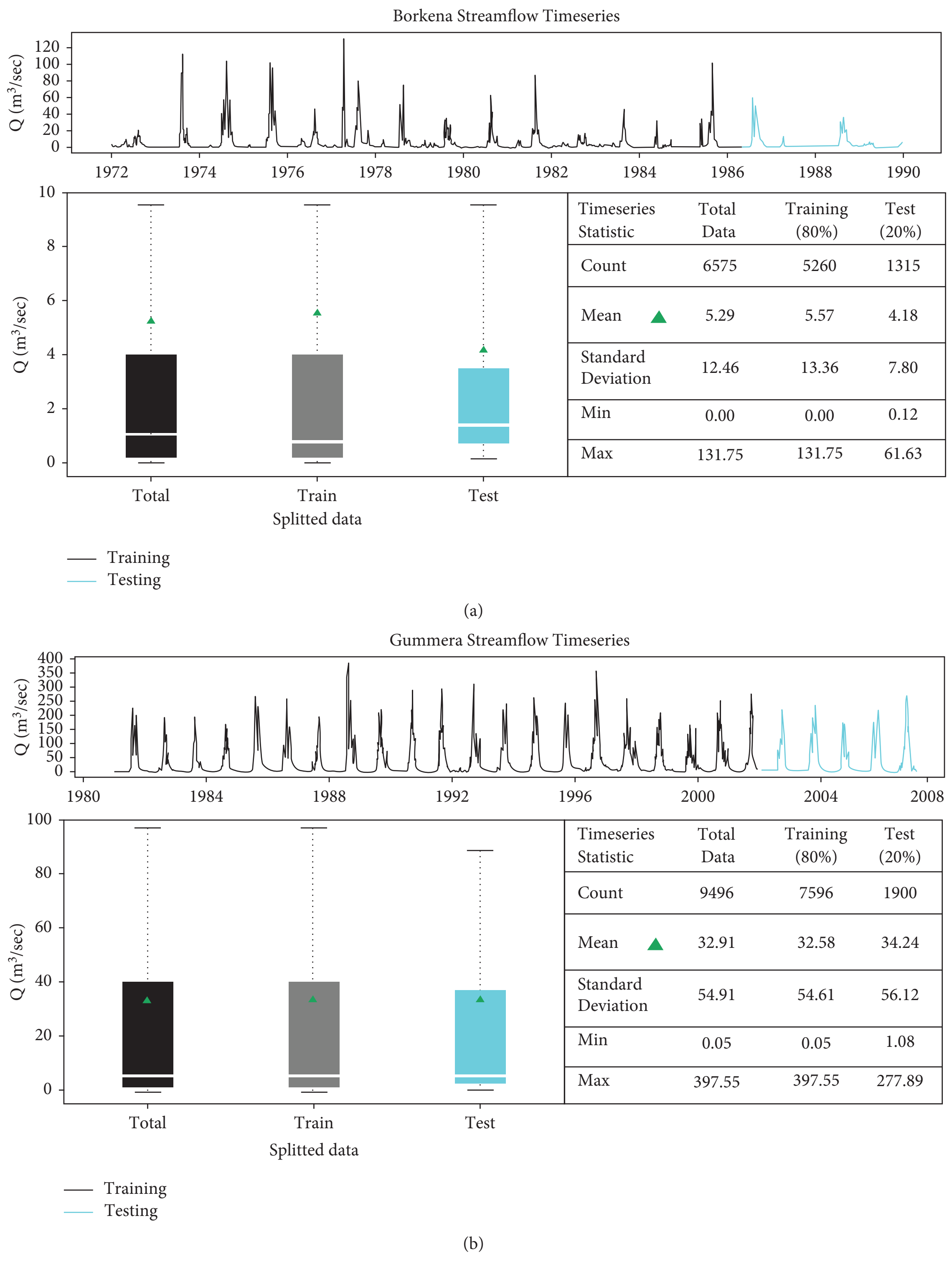
(a)



(b)

Figure 1: The location of the study areas. (a) Gummera, (b) Borkena.

3.1.1. Multilayer Perceptron (MLP). ANN or feed-forward multilayer perceptron (MLP) is an immensely used architecture in the hydrological literature [47]. Perceptron operates artificially, replicating our brain processing system and passing on different mathematical and probabilistic operations [48]. MLP contains three main layers: input layer, hidden layer, and output layer. The network becomes deep and can extract higher-order statistics by adding more



Timeseries Statistic	Total Data	Training (80%)	Test (20%)
Count	6575	5260	1315
Mean ▲	5.29	5.57	4.18
Standard Deviation	12.46	13.36	7.80
Min	0.00	0.00	0.12
Max	131.75	131.75	61.63

Timeseries Statistic	Total Data	Training (80%)	Test (20%)
Count	9496	7596	1900
Mean ▲	32.91	32.58	34.24
Standard Deviation	54.91	54.61	56.12
Min	0.05	0.05	1.08
Max	397.55	397.55	277.89

FIGURE 2: Descriptive statistics and the corresponding box plots of split data. (a) Borkena station, (b) Gummera station.

hidden layers [44]. Three-layered MLP is common in hydrological time series modelling.

A typical diagram of one node (j^{th}) ANN is displayed in Figure 3. Conditional to the layer location, the series of inputs form an input vector $X = (x_1, x_i, \ldots, x_n)$. The equivalent series of weights fit to each input form a weight vector $W_j = (w_{1j}, w_{ij}, \ldots, w_{nj})$. The output (Y_j) for the node j is calculated using the value of a function (f) with the inner product of the input vector (X) and weight vector (W_j) minus bias (b_j) [49]. The stated operation is displayed as follows:

$$Y_j = f\left(X \cdot W_j - b_j\right). \tag{1}$$

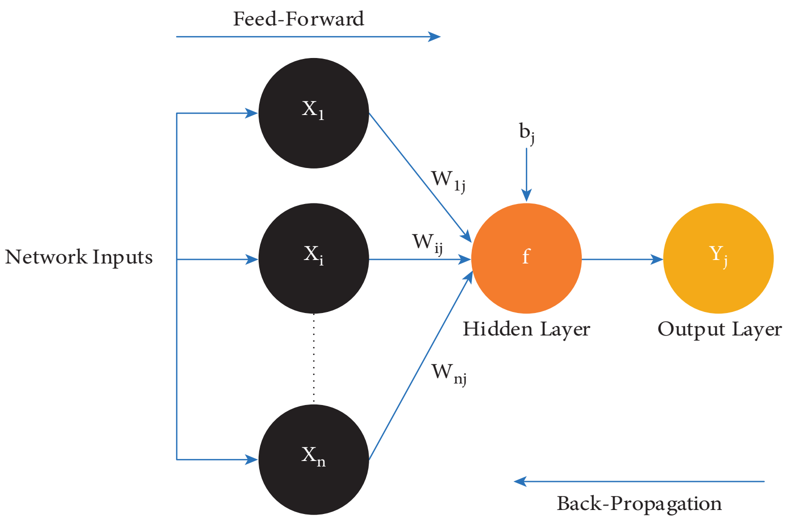


Figure 3: Schematic diagram of MLP for node *j*.

The activation function (*f*) helps decide a neuron activates a process or not, and these are the few commonly used activation functions: Rectified Linear Unit (ReLu), Leaky ReLu, Sigmoid, Hyperbolic Tangent (Tanh), and Softmax [50]. Even though ANN has been enormously applied in hydrological modelling for the past decades, its performance to capture extreme events is doubtful for complex problems such as rainfall-runoff processes [47].

3.1.2. Long Short-Term Memory (LSTM). Long Short-Term Memory (LSTM) networks are unique to MLP because the networks have recurrent connections; the information from previous long time-step memories is used to formulate forecasting. Overcoming vanishing and exploding gradients makes LSTM more popular in sequence and time series analysis than the traditional Recurrent Neural Network (RNN) [51]. LSTM networks have memory cells and three gates: input, forget, and output. These gates are responsible for the network to save, forget, pay attention, or pass the information to other cells [27]. Figure 4 displays the typical LSTM memory cell with three gated layers, and the detailed mathematical formulation for the network components is described as follows:

$$f_t = \sigma\left(u_f x_t + w_f h_{t-1} + b_f\right), \quad (2)$$

$$i_t = \sigma\left(u_i x_t + w_i h_{t-1} + b_i\right), \quad (3)$$

$$o_t = \sigma\left(u_o x_t + w_o h_{t-1} + b_o\right), \quad (4)$$

$$c_{et} = \tanh\left(w_c x_t + u_c h_{t-1} + b_c\right), \quad (5)$$

$$c_t = f_t * c_{t-1} + i_t * c_{et}, \quad (6)$$

$$h_t = \tanh\left(c_t\right) * o_t, \quad (7)$$

where f_t (equation (2)) is a forget gate that has a responsibility to choose the information to reject from the cell, i_t (equation (3)) is an input gate that can decide on the input values to update the memory state, and o_t (equation (4)) is an output gate that can decide the output value after analyzing the input and memory of the cell. The weight matrices w_i, w_f, w_o, and w_c correspond to the input gate, forget gate, output gate, and cell gate units, respectively. While u_i, u_f, u_o, and u_c weight matrices map the hidden layer output gates, b_i, b_f, b_o, and b_c are the bias vectors of the input gate, forget gate, output gate, and cell gate units, respectively. Moreover, c_t (equation (6)) and h_t (equation (7)) are a memory cell and hidden state [24].

3.1.3. Bidirectional LSTM (Bi-LSTM). Bi-LSTM is also the other option for getting the most out of RNN by stepping through the input time steps forward and backward. Although Bi-LSTMs were developed for speech recognition, the use of bidirectional input sequences is one of the principal options for sequence prediction nowadays. The hidden layer of the Bi-LSTM model needs to save two values, in which h_t involves the forward calculation and h'_t involves the backward calculation. The final output value Y_t is obtained by combining the outputs of the forward layers and backward layers' outputs [52].

Each point in the input sequence of the output layer is provided with the complete past and future contextual information. There is no information flow between the forward and backward hidden layers, ensuring that the expanded graph is acyclic [53]. Figure 5 displays the structure of bidirectional LSTM architecture.

3.1.4. Gated Recurrent Unit (GRU). GRU is the newer generation of LSTM that merges input and forget gates into the update gate. Hence, it has fewer parameters, faster running time, and debatable performance than LSTM [24, 26, 27, 40]. Update and reset gates are the two gates available in GRU. Update gate renews the current memory, which enables the memorization of valuable information; on the contrary, the reset gate clears the current memory to

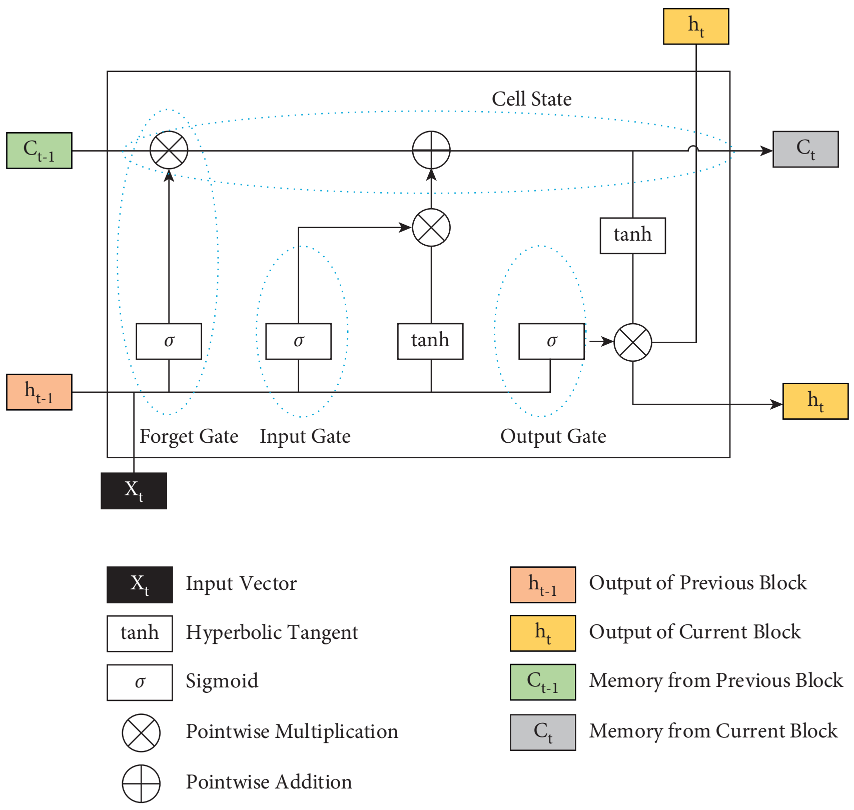


FIGURE 4: LSTM memory cell with three gated layers: forget gate f_t, input gate i_t, and output gate o_t, controlling the activation of cells c_{t-1} and c_t [38].

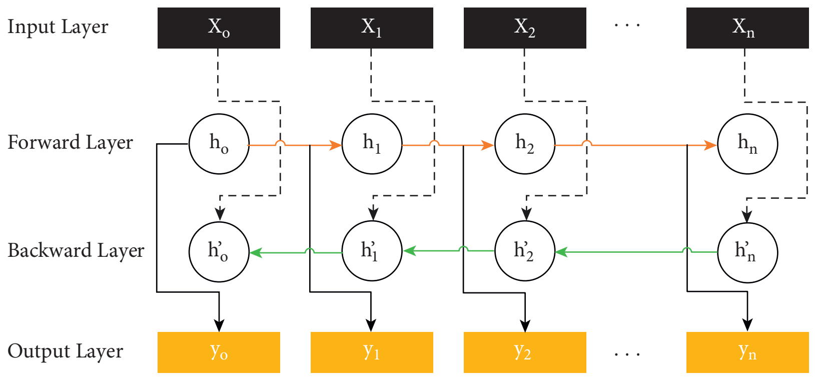


FIGURE 5: Bidirectional LSTM architecture [39].

forget invaluable information at any time step. Figure 6 shows the structure of the GRU network, and the detailed equations of the hidden units are described as follows:

$$Z_t = \sigma(W_z X_t + U_z h_{t-1} + b_z), \quad (8)$$

$$r_t = \sigma(W_r X_t + U_r h_{t-1} + b_r), \quad (9)$$

$$\widehat{h}_t = \tanh(W_h X_t + (r_t \odot h_{t-1}))U_h + b_h, \quad (10)$$

$$h_t = (1 - Z_t) \odot h_{t-1} + Z_t \odot \widehat{h}_t, \quad (11)$$

where X_t is the input vector, Z_t (equation (8)) is the update gate vector, r_t (equation (9)) is the reset gate vector, h_t (equation (11)) is the output vector, $\widehat{h}_t$ (equation (10)) is a candidate activation vector, W, U, and b are parameter matrices, and the sign $\odot$ denotes Hadamard product.

3.2. Main Model Development.

3.2. Main Model Development. Optimizing deep learning models require decisions on a combination of large hyperparameters, including the number of layers, number of units, batch size, epochs, and learning rate [54]. A random search can produce an infinite number of hyperparameter

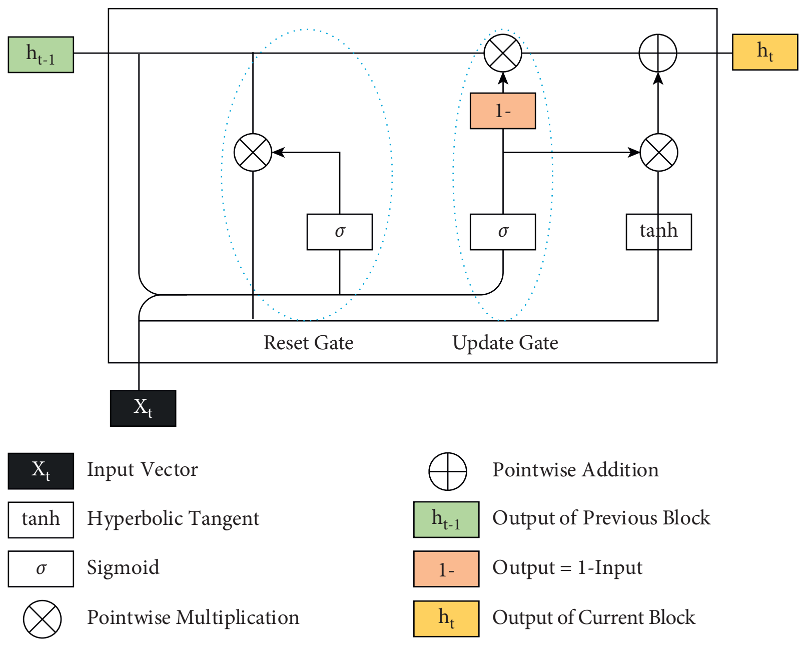


Figure 6: The structure of gated recurrent unit (GRU) network [38].

combinations with a median cost from the four major hyperparameter optimization techniques: trial-and-error, grid, random, and probabilistic approach [31]. Hence, in this study, we use a computationally efficient randomized search method called Keras Tuner developed by the Google team to search random combinations of parameters for optimized performance. The detailed flow chart of the proposed methodology is presented here in Figure 7.

The proposed models applied a double fully connected hidden layer. The minimum and the maximum numbers of neurons at each hidden layer were set with prior experience. Then, the output layer is linked to a dense layer with a single output neuron with a linear activation function. The network is compiled with Adam optimizer and mean squared error loss function, and the details of hyperparameter value ranges and choices are listed in Table 1. Moreover, the following paragraph discusses each hyperparameter optimized using Keras tuner.

3.2.1. Activation Function. In deep learning models, the activation function defines the output from the inputs each node receives. In our case, we applied Rectified Linear Units (ReLU) in all layers except the output layer.

3.2.2. Learning Rate. In deep learning, the learning rate is one of the hyperparameters that decides the step size at each time the model progresses to a minimum loss function. Hence, it is crucial to optimize the learning rate properly; otherwise, the model may converge slowly with too small learning rate or diverge from the optimal error points with large learning rate values [55]. This study sets three values ($1e-2$, $1e-3$, or $1e-4$) to choose using Keras Tuner.

3.2.3. Number of Epochs. The other hyperparameter that decides how much time the deep learning algorithm will learn with the entire training samples. When we set more epochs, the weight in the model will get more chance to update itself. The loss curve goes through different stages such as underfitting, optimal state, or overfitting; even though there are no strict rules for these hyperparameter configurations, we set the minimum (10) and maximum (100) values for optimization using Keras Tuner.

3.2.4. Number of Batch Sizes. The batch size is a sample size cluster processed before the model updates itself.

3.2.5. Drop Out. The dropout layer is a hyperparameter that prevents overfitting and enhances training performance. Hence, at each iteration, the dropout freezes a fraction of the neuron from training, and it is defined on a range of 0 to 1.

Different open-source Python libraries were used for model development, such as Tensorflow, Keras, Scikit-Learn, Matplotlib for visualization, Statsmodels for performance evaluation. Moreover, the simulation was also conducted on a computer with Processor: Intel(R) Core (TM) i7-6500U CPU 2.50 GHz and RAM: 8 Gigabytes memory.

3.3. Input Time Lag Selection. To reprocess the highly correlated time series delay that was decomposed and used as a single input to a deep learning neural network, the autoregessive model using the partial autocorrelation function (pacf) was applied. Equation (12) shows the autoregressive model A.R. (p).

$$x_t = \varphi_1 x_{(t-1)} + \varphi_2 x_{(t-2)} + \cdots + \varphi_p x_{(t-p)} + \varepsilon_t, \qquad (12)$$

where φ is the autoregressive parameter, x is the observation at time t, and ε is the weighted noise at time t. The autoregressive model explores the correlation between current and past values [56]. As shown in Figure 8, the partial autocorrelation of daily streamflow time series with a 95% confidence interval, the time delay of 4 days, was considered for both case study areas in this study.

3.4. Performance Measures. The following performance measures were used to evaluate the accuracy of the model developed: coefficient of determination (R^2), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) [57].

(i) Coefficient of Determination (R^2)

$$R^2 = \frac{n\left(\sum Q_{\text{obs}} * Q_{\text{sim}}\right) - \left(\sum Q_{\text{obs}}\right) * \left(\sum Q_{\text{sim}}\right)}{\sqrt[1/2]{\left[n\left(\sum Q_{\text{obs}}^2\right) - \left(\sum Q_{\text{obs}}\right)^2\right] * \left[n\left(\sum Q_{\text{sim}}^2\right) - \left(\sum Q_{\text{sim}}\right)^2\right]}}. \tag{13}$$

(ii) Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{N}\left(Q_{\text{obs}}^t - Q_{\text{sim}}^t\right)^2}{N}}. \tag{14}$$

(iii) Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|Q_{\text{obs}}^t - Q_{\text{sim}}^t\right|. \tag{15}$$

(iv) Mean Absolute Percentage Error (MAE):

$$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{Q_{\text{obs}}^t - Q_{\text{sim}}^t}{Q_{\text{obs}}^t}\right|, \tag{16}$$

where Q_{obs} = discharge observed, Q_{sim} = discharge simulated, and n = number of observations. The range of R^2 lies between 0 and 1, representing, respectively, no correlation and a perfect correlation between observed and simulated values, whereas, for RMSE, MAE, and MAPE, the better performance is reached when we are close to 0. If $R^2 > 0.90$, the simulation is very acceptable; if $0.60 < R^2 < 0.90$, the simulation is acceptable; and if $R^2 < 0.60$, the simulation is unacceptable [58].

4. Results and Discussion

Previous analyses revealed a wide range of results, which varied according to the type of deep learning architecture used, the degree of climatic variability, and the timescale used. The GRU model outperforms both the extreme learning machine (ELM) and the least-squares support vector machine (LSSVM) on monthly univariate streamflow data from the Shangjingyou and Fenhe reservoir stations in the upper reaches of the Fenhe River [59]. Sahoo et al. [19] also demonstrated the superiority of LSTM over RNN on univariate daily discharge data from the Basantapur gauging station in India's Mahanadi River basin. Suradhaniwar et al. [7] demonstrated that SARIMA and SVR models outperform NN, LSTM, and RNN models when hourly averaged univariate time series data is used. Even though the best model generalization is complex, case-based analysis is the most effective method for determining which model best fits a given situation [60]. For the first time in this study, we attempted to organize the findings around three distinct themes: model variability, lag time variability, and time series characteristics (climatic variability).

The following section discusses the performance of four selected models under four different input time lag scenarios; in total, 16 experimental results are presented. Tables 2 and 3 show the optimized hyperparameter values for both stations and all scenarios using Keras Tuner. Additionally, Tables 4 and 5 illustrate performance metrics in terms of RMSE, MAE, MAPE, R^2, and TTPE (sec). In the testing phase, single-step streamflow forecasting results demonstrated very acceptable performance for both case study areas.

4.1. Model Variability. We used four evaluation matrices to investigate the effect of model variability on the accuracy of single-step streamflow forecasting. Then, using box plots, we could visualize the spread of prediction error (m^3/s). Additionally, we plotted a bar chart (Figure 9) with different prediction error classes to identify the class limit with the highest error concentration. As a result, as shown in Table 4, GRU's model has a slight performance advantage over MLP and S-LSTM for Borkena station's lag time ($T+2$ and $T+3$). Whereas, for the Gummera catchment (Table 5), MLP outperformed GRU and S-LSTM in terms of performance increment in lag time ($T+1$ and $T+3$). Prediction error box plots and bar chart graphs (Figures 10 and 11) were used to investigate these high-performing architectures further. Hence, the prediction error of GRU is typically concentrated in small ranges (0 to 0.5 m^3/s) for Borkena and (0 to 2.5 m^3/s) for the Gummera catchment. Moreover, as shown in Tables 4 and 5, when considering computational speed, MLP demonstrated the quickest training time per epoch, followed by S-LSTM, GRU, and Bi-LSTM.

4.2. Time Series Characteristics (Climatic Variability). The other major issue affecting deep learning model performance is time series characteristics. As a result, in this study, the four-evaluation metrics displayed Borkena's river catchment forecasting result is more accurate than Gummera's catchment with RMSE, MAE, MAPE, R^2, ranging between (0.81 to 1.53, 0.29 to 0.96, 0.16 to 1.72, 0.96 to 0.99) and (17.43 to 17.99, 7.76 to 10.54, 0.16 to 1.03, 0.89 to 0.90) for both catchments, respectively. The possible cause for this performance variation between the two catchments is the significant natural streamflow time series variability in the Borkena catchment. Furthermore, the spread of prediction error (m^3/sec) in Figures 10 and 11 shows that the error class limit for most cases is smaller in the Borkena station than in the Gummera station.

4.3. Lag Time Variability. In univariate streamflow forecasting, lagged variables are the other significant factors that hold temporal information and affect model performance. Taylor diagram in Figure 12 shows the forecasting ability of

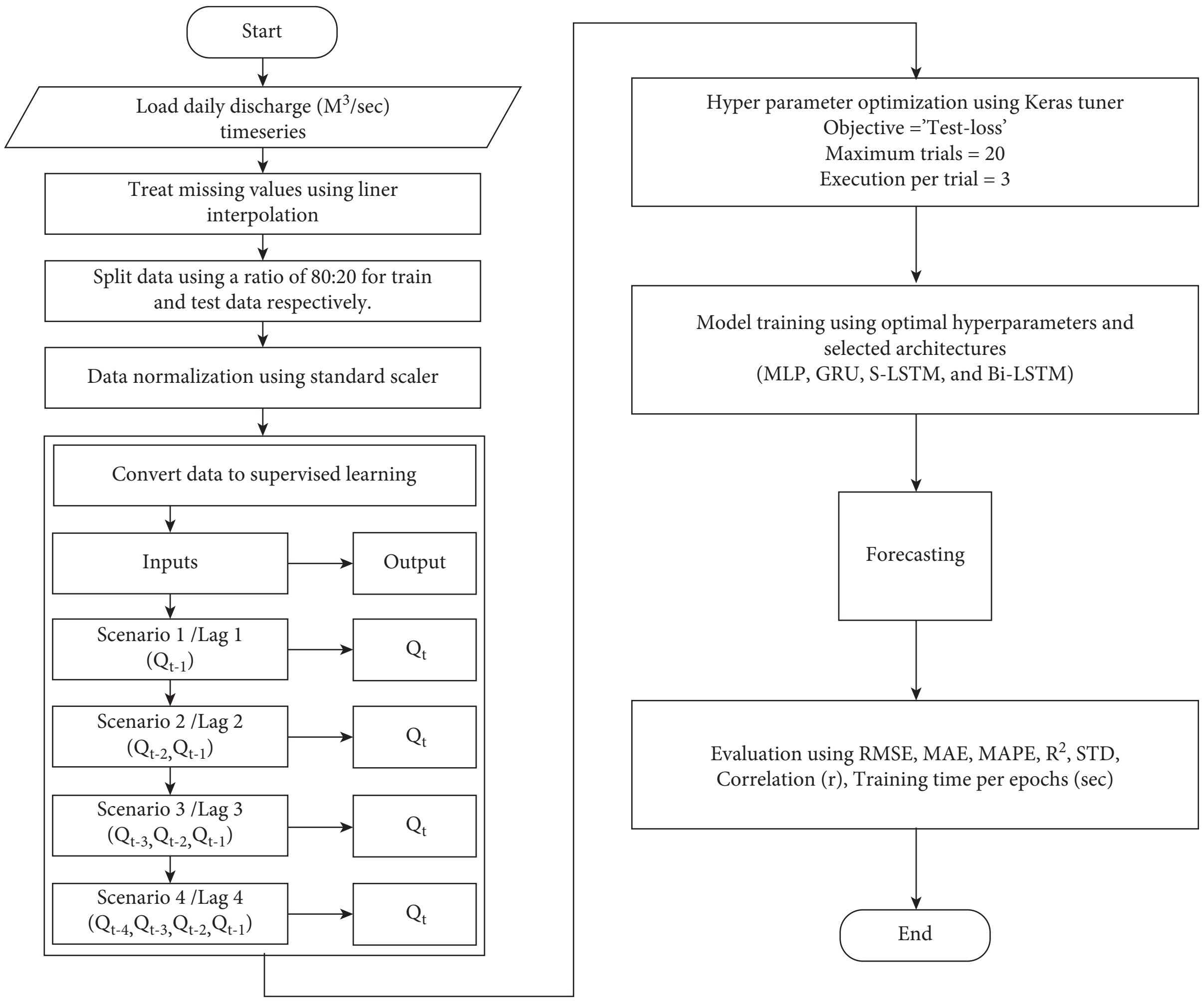

Start
Load daily discharge (M3/sec) timeseries
Treat missing values using liner interpolation
Split data using a ratio of 80:20 for train and test data respectively.
Data normalization using standard scaler
Convert data to supervised learning
Inputs
Output
Scenario 1 /Lag 1 (Qt-1)
Qt
Scenario 2 /Lag 2 (Qt-2,Qt-1)
Qt
Scenario 3 /Lag 3 (Qt-3,Qt-2,Qt-1)
Qt
Scenario 4 /Lag 4 (Qt-4,Qt-3,Qt-2,Qt-1)
Qt
Hyper parameter optimization using Keras tuner
Objective ='Test-loss'
Maximum trials = 20
Execution per trial = 3
Model training using optimal hyperparameters and selected architectures (MLP, GRU, S-LSTM, and Bi-LSTM)
Forecasting
Evaluation using RMSE, MAE, MAPE, R2, STD, Correlation (r), Training time per epochs (sec)
End

FIGURE 7: Data analysis flow chart proposed for the study.

TABLE 1: Model hyperparameter choices or value ranges for optimization by keras tuner.

N°	Hyperparameters	Value Ranges**			Choices	Default
		Min	Max	Step		
1	MLP, LSTM, Bi-GRU+LSTM, or GRU layer 1 units	5	40	5	*	*
2	Dropout 1	0.0	0.3	0.1	*	0.2
3	MLP, LSTM, Bi-LSTM, or GRU layer 2 units	5	40	5	*	*
4	Learning rate	*	*	*	$1e-2$, $1e-3$ or $1e-4$	*
5	Number of epochs	10	100	10	*	*
6	Number of batch sizes	10	100	10	*	*

**Value ranges or choices for optimization by keras tunner: (objective = "test loss," max trials = 20, executions per trial = 3). *Not applicable.

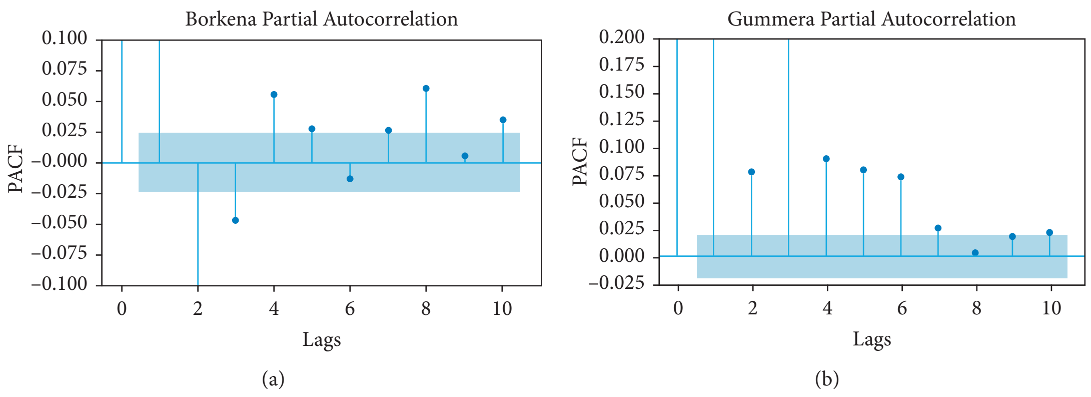


Figure 8: Partial autocorrelation of the daily time series for both catchments (lag units are in days, and the blue shadow lines indicate 95% confidence intervals). (a) Borkena, (b) Gummera.

Table 2: Keras tuner optimized hyperparameter values for Borkena station with its MSE score.

Hyperparameters	MLP				S-LSTM				Bi-LSTM				GRU			
	$T+1$	$T+2$	$T+3$	$T+4$	$T+1$	$T+2$	$T+3$	$T+$	$T+1$	$T+2$	$T+3$	$T+4$	$T+1$	$T+2$	$T+3$	$T+4$
Layer_1_units	35	10	40	35	25	35	40	40	35	40	25	20	25	30	15	10
Dropout_1	0.0	0.0	0.0	0.1	0.1	0.0	0.2	0.2	0.1	0.2	0.0	0.1	0.1	0.0	0.1	0.1
Layer_2_units	10	20	25	15	35	25	30	10	20	30	20	30	20	25	40	25
Learning rate	0.0001	0.0001	0.0001	0.001	0.0001	0.0001	0.001	0.001	0.0001	0.001	0.001	0.001	0.0001	0.001	0.0001	0.001
Number of epochs	90	100	100	40	60	80	20	30	70	20	50	70	50	60	40	50
Number of batch sizes	20	20	30	30	30	40	20	10	40	20	20	30	50	30	10	90
Score (MSE)	0.099	0.099	0.096	0.097	0.099	0.099	0.098	0.096	0.099	0.099	0.097	0.096	0.099	0.099	0.099	0.097

Table 3: Keras tuner optimized hyperparameter values for Gummera station with its MSE score.

Hyperparameters	MLP				S-LSTM				Bi-LSTM				GRU			
	$T+1$	$T+2$	$T+3$	$T+4$	$T+1$	$T+2$	$T+3$	$T+4$	$T+1$	$T+2$	$T+3$	$T+4$	$T+1$	$T+2$	$T+3$	$T+4$
Layer_1_units	25	25	10	25	20	30	15	25	10	25	15	15	5	40	25	15
Dropout_1	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.2
Layer_2_units	15	40	30	25	30	10	35	10	5	20	10	20	30	20	25	15
Learning rate	0.0001	0.0001	0.0001	0.001	0.001	0.001	0.01	0.001	0.001	0.001	0.01	0.001	0.01	0.01	0.0001	0.001
Number of epochs	70	90	80	50	60	50	50	80	60	90	50	30	80	40	50	30
Number of batch sizes	60	60	70	60	80	40	30	50	50	40	20	10	20	30	50	40
Score (MSE)	0.019	0.011	0.012	0.015	0.019	0.011	0.013	0.015	0.018	0.011	0.013	0.013	0.018	0.012	0.013	0.016

the proposed models with the observed test data for both case study areas. The diagram is designed on a two-dimensional scale: the standard deviation on the polar axis, root mean square error, and correlation coefficient on the radial axis. It shows that, irrespective of deep learning models, forecasting with a lag time of four gives us a time series closest to the standard deviation of the actual test observations. Moreover, Figures 13 and 14 also display the

Table 4: Performance comparison of the proposed models for different input time lags in Borkena station.

Borkena	$T+1$					$T+2$					$T+3$					$T+4$				
	RMSE	MAE	MAPE	R^2	TTPE * (sec)	RMSE	MAE	MAPE	R^2	TTPE* (sec)	RMSE	MAE	MAPE	R^2	TTPE * (sec)	RMSE	MAE	MAPE	R^2	TTPE* (sec)
MLP	1.09	0.47	0.85	0.98	0.24	0.89	0.38	0.76	0.99	0.16	1.06	0.35	0.19	0.98	0.12	**0.81**	**0.29**	**0.34**	**0.99**	0.29
GRU	1.15	0.64	0.84	0.98	1.29	**0.85**	**0.31**	**0.16**	**0.99**	0.79	**0.91**	**0.35**	**0.18**	**0.98**	0.99	1.35	0.69	1.41	0.97	1.31
S-LSTM	1.07	**0.38**	**0.29**	**0.98**	0.30	0.87	0.36	0.26	0.99	0.54	1.02	0.64	0.99	0.98	0.81	1.53	0.96	1.72	0.96	0.58
Bi-LSTM	1.06	0.39	0.31	0.98	0.88	0.90	0.46	0.76	0.99	1.55	1.12	0.51	0.95	0.98	1.55	0.98	0.44	0.53	0.98	5.3

*TTPE(training time per epochs). The bold values indicate the best performance score for each time lag.

Table 5: Performance comparison of the proposed models for different input time lags in Gummera station.

Gummera	T + 1					T + 2					T + 3					T + 4				
	RMSE	MAE	MAPE	R^2	TTPE * (sec)	RMSE	MAE	MAPE	R^2	TTPE* (sec)	RMSE	MAE	MAPE	R^2	TTPE * (sec)	RMSE	MAE	MAPE	R^2	TTPE* (sec)
MLP	17.67	**7.84**	**0.18**	**0.90**	0.88	17.68	7.87	0.19	0.90	0.75	**17.43**	**7.76**	**0.18**	**0.90**	**0.61**	17.56	9.73	0.78	0.90	0.95
GRU	17.66	8.39	0.41	0.90	0.51	17.76	**7.82**	**0.16**	**0.90**	1.43	17.47	8.29	0.35	0.90	1.24	17.99	10.54	1.03	0.89	0.53
S-LSTM	17.66	8.29	0.39	0.90	1.54	17.71	8.04	0.22	0.90	0.75	17.69	8.51	0.32	0.90	2.99	17.69	**8.41**	**0.35**	**0.90**	3.31
Bi-LSTM	17.63	7.98	0.27	0.90	0.92	17.93	8.39	0.29	0.89	2.34	17.83	9.18	0.56	0.89	2.77	17.56	8.56	0.39	0.90	1.84

*TTPE(training time per epochs).The bold values indicate the best performance score for each time lag.

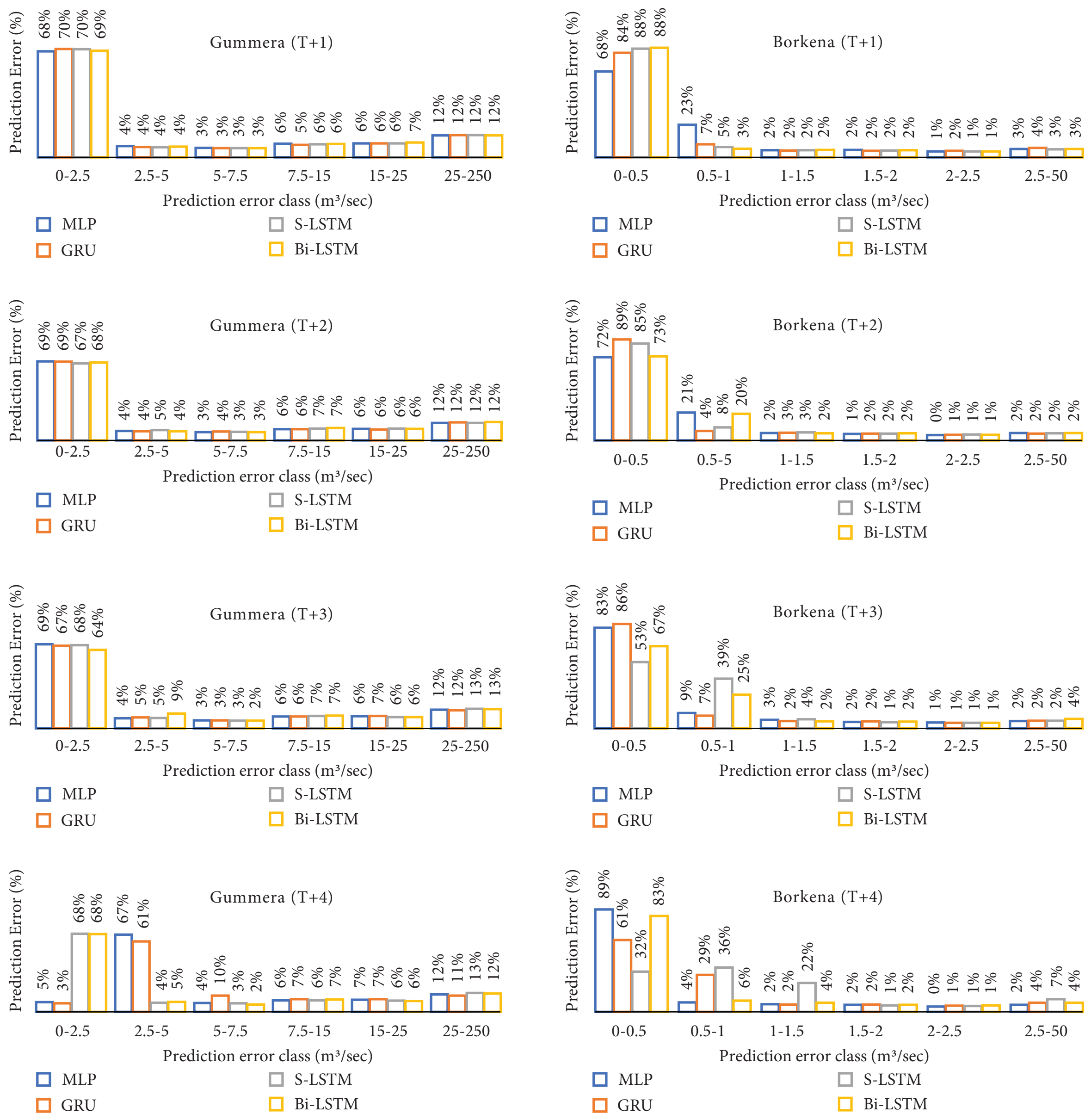


FIGURE 9: Bar chart showing the frequency (%) of absolute prediction errors |PE| (m^3/sec) with different class limits for the proposed four models and input lagged time variables in both catchments.

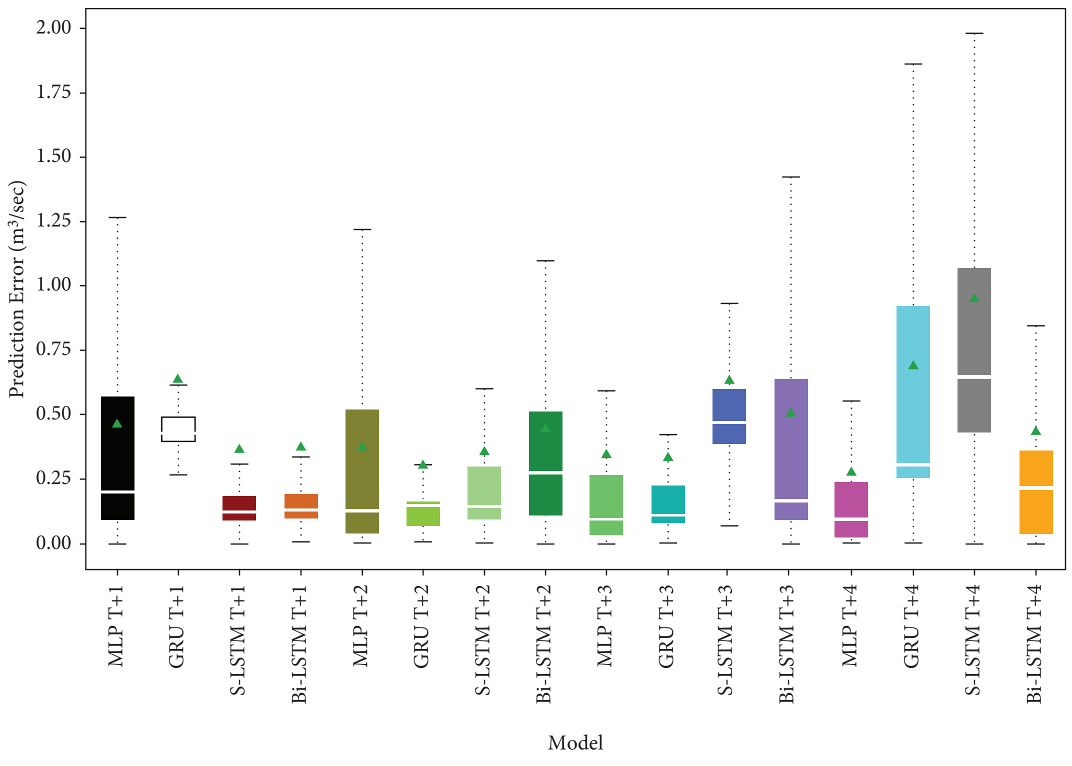


FIGURE 10: Spread of prediction error (m^3/s) or box plot for the proposed models during the test period (Borkena station).

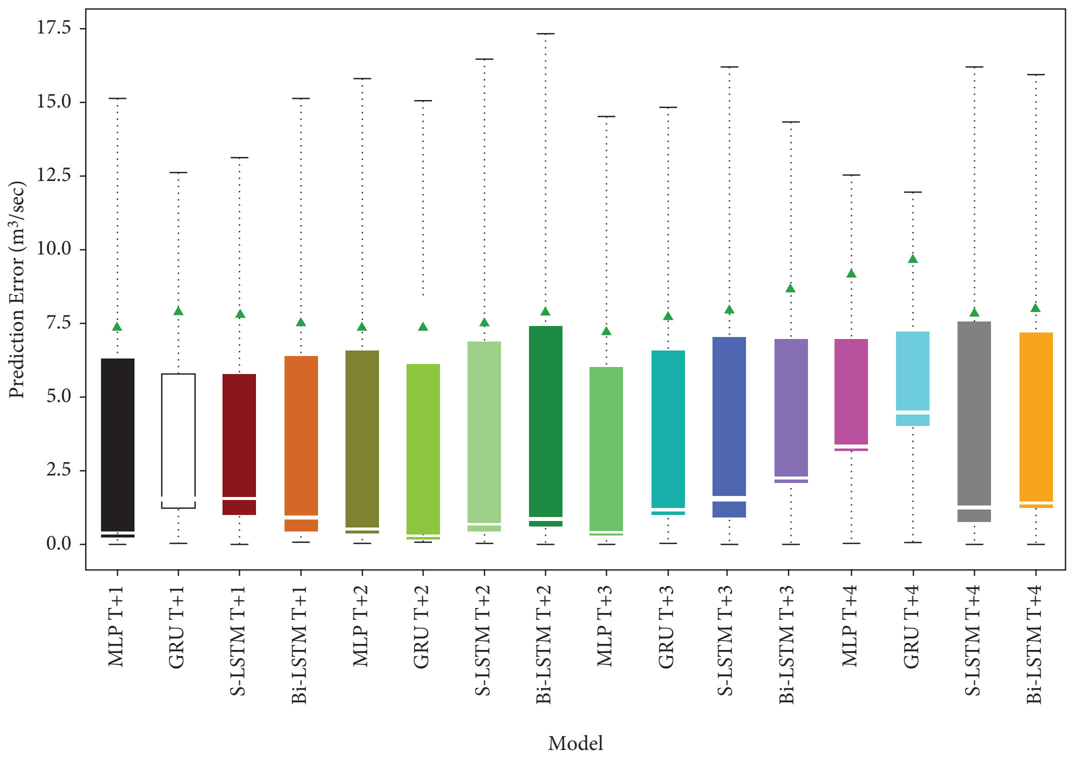
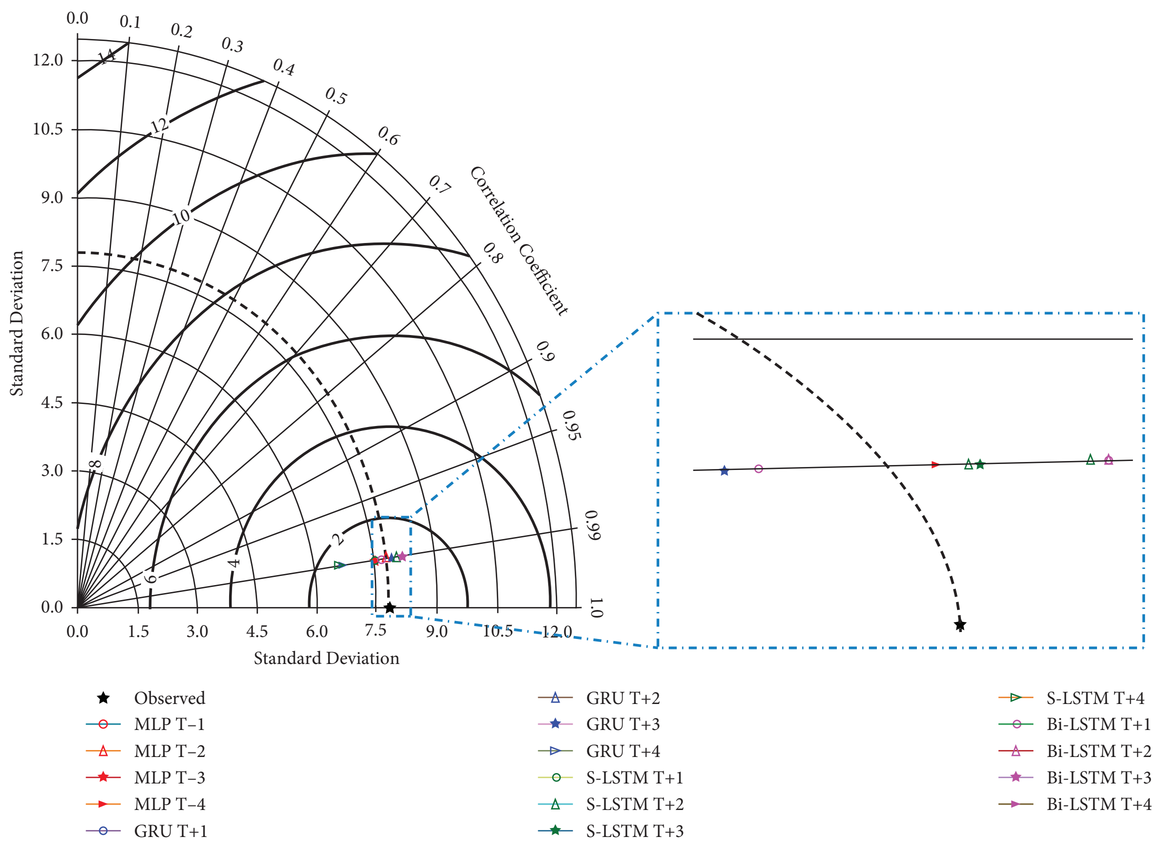
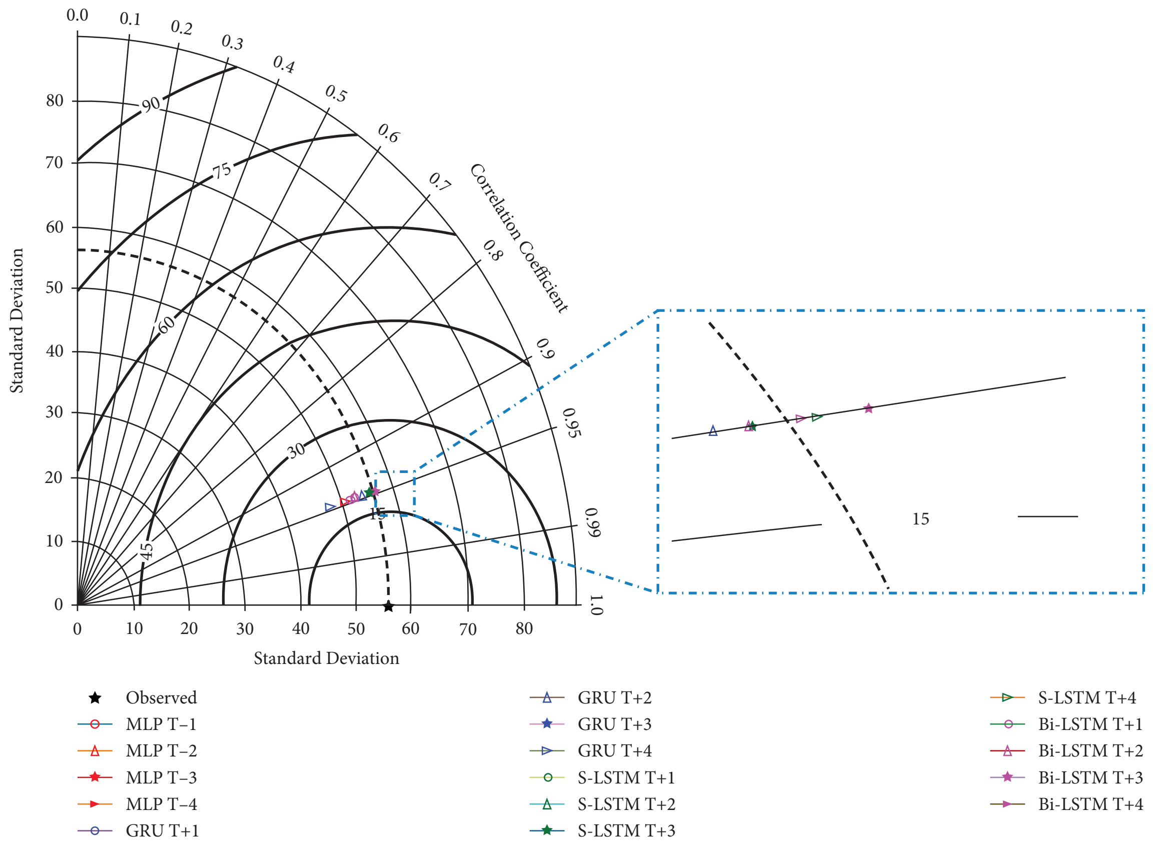


FIGURE 11: Spread of prediction error (m^3/s) or box plot for the proposed models during the test period (Gummera station).



(a)



(b)

Figure 12: Taylor diagram displays the standard deviations, root mean square error, and correlation coefficient between observed and predicted streamflow for the proposed four models and lagged time variables (Q m^3/s). (a) Borkena, (b) Gummera.

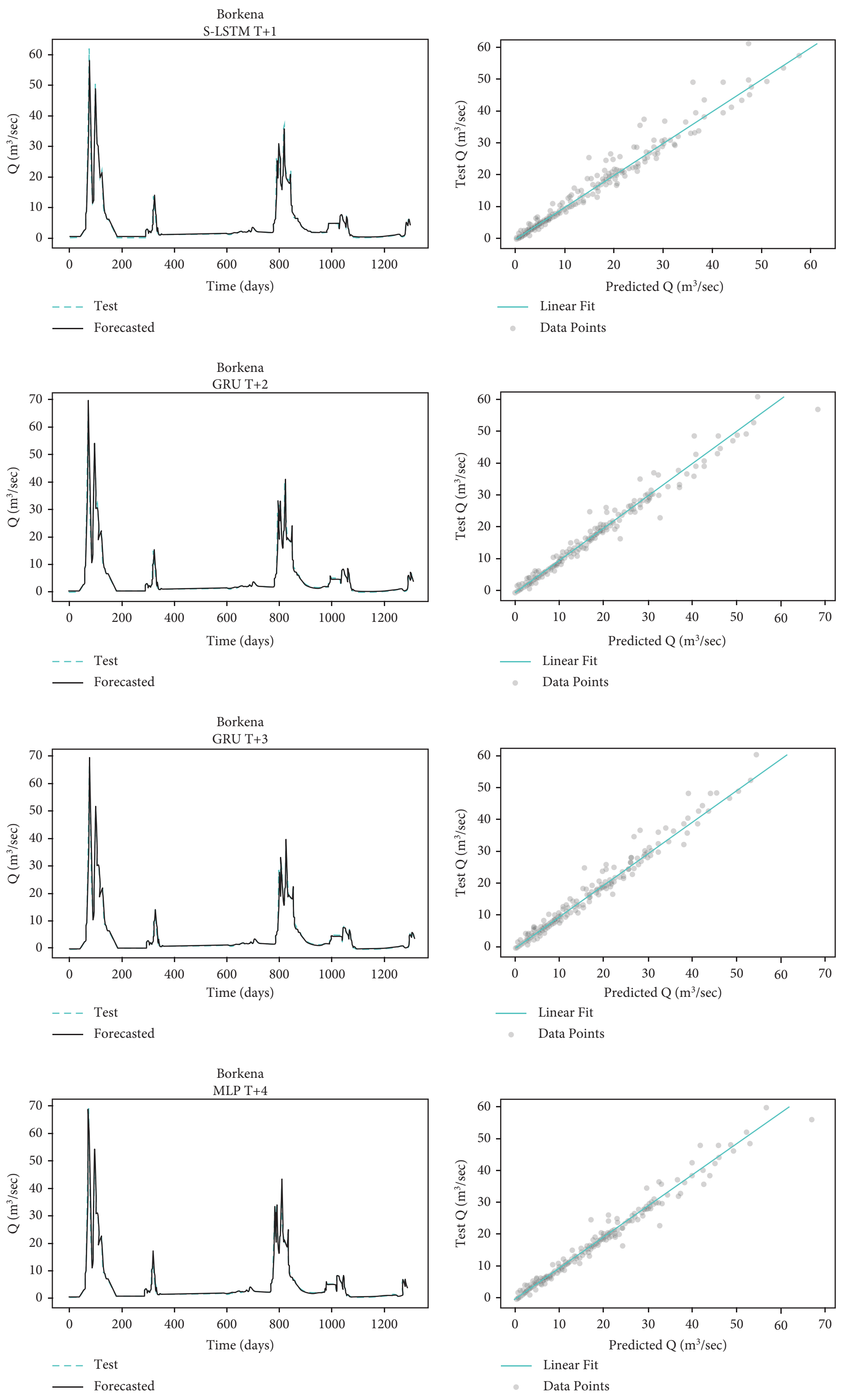


Figure 13: Comparison of true values and predicted values of the optimized high score deep learning model for each time lag (Borkena station).

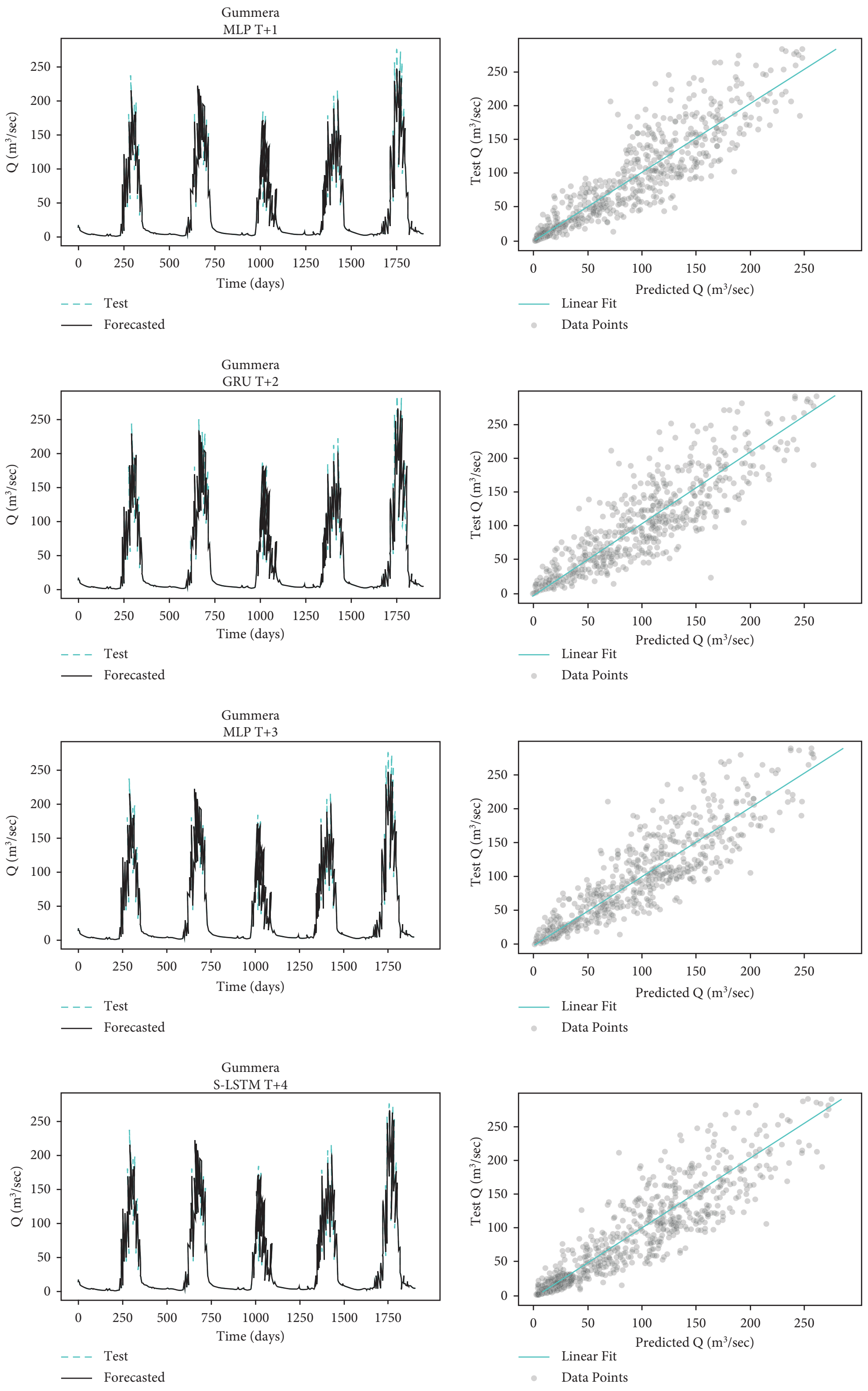


Figure 14: Comparison of true values and predicted values of the optimized high score deep learning model for each time lag (Gummera station).

actual values and predicted values of the optimized high score deep learning model for each time lag and both case study catchments.

5. Conclusions

This study showed a comparative analysis of different deep learning algorithms for one-step daily streamflow forecasting at two subcatchment stream flow outlets. MLP, S-LSTM, Bi-LSTM, and GRU are the four algorithms used in this study. The study clearly showed the impacts of climatic (time series characteristics) and lagged time variability on the performance of different proposed deep learning models. The following key points will elaborate on the outcome of this research.

(i) Deep learning models have excellent potential in forecasting short-term daily streamflow in different time series characteristics.

(ii) The performance of deep learning models for short-term streamflow forecasting varies with time series characteristics and input time lag variations. Moreover, the Borkena station has more significant natural streamflow variability than the Gummera station, which is also reflected in the model results. Hence, this study showed that catchment response variability impacts deep learning model performance.

(iii) MLP and GRU outperform S-LSTM and Bi-LSTM on a nearly equal basis for single-step short-term streamflow forecasting in both case study areas. However, the performance is relative to the lagged time variations.

(iv) Catchment characteristics had a high impact on the performance of streamflow forecasting than deep learning model architectures and lagged time variations.

(v) The study also showed that classical MLP could almost equally perform with S-LSTM and GRU deep learning networks on a small amount of streamflow time series data.

Future research may further expand this study's findings to other climatic regions, hybrid deep learning model architectures, hyperparameter tuning, and lagged time selection methods. We must also investigate the effects of large input variability on deep learning models for univariate streamflow forecasting in all its implications. As part of our future work, we plan to implement an ensemble learning approach to simulate streamflow from remote sensing-derived data products (precipitation and vegetation indexes) using a combination of neural networks, decision trees, and boosting algorithms.

Data Availability

The corresponding author's raw metrological and hydrological data sets used for the Borkena and Gummera watershed are available upon request. However, authorization letters are required from the Ethiopian National Metrological Agency (NMA) and Ethiopian Ministry of Water, and Energy (MoWE) (https://mowe.gov.et/)

Conflicts of Interest

The authors declare that they have no conflicts of interest for this article.

Acknowledgments

The authors acknowledge the Ethiopian Ministry of Water, and Energy (MoWE) for hydrological data and the Ethiopian National Metrological Agency (NMA) for metrological data.

References

[1] P. Sharma and D. Machiwal, "Streamflow forecasting," in *Advances in Streamflow Forecasting*, pp. 1–50, Elsevier, Amsterdam, Netherlands, 2021.

[2] Z. M. Yaseen, A. El-shafie, O. Jaafar, H. A. Afan, and K. N. Sayl, "Artificial intelligence based models for streamflow forecasting: 2000-2015," *Journal of Hydrology*, vol. 530, pp. 829–844, 2015.

[3] Z. M. Yaseen, W. H. M. W. Mohtar, A. M. S. Ameen et al., "Implementation of univariate paradigm for streamflow simulation using hybrid data-driven model: case study in tropical region," *IEEE Access*, vol. 7, pp. 74471–74481, 2019.

[4] R. K. Jaiswal, S. Ali, and B. Bharti, "Comparative evaluation of conceptual and physical rainfall-runoff models," *Applied Water Science*, vol. 10, no. 1, Article ID 48, 2020.

[5] M. Ghaith, A. Siam, Z. Li, and W. El-Dakhakhni, "Hybrid hydrological data-driven approach for daily streamflow forecasting," *Journal of Hydrologic Engineering*, vol. 25, no. 2, Article ID 04019063, 2020.

[6] Z. Zhang, Q. Zhang, and V. P. Singh, "Univariate streamflow forecasting using commonly used data-driven models: literature review and case study," *Hydrological Sciences Journal*, vol. 63, no. 7, pp. 1091–1111, 2018.

[7] S. Suradhaniwar, S. Kar, S. S. Durbha, and A. Jagarlapudi, "Time series forecasting of univariate agrometeorological data: a comparative performance evaluation via one-step and multi-step ahead forecasting strategies," *Sensors*, vol. 21, no. 7, Article ID 2430, 2021.

[8] A. Danandeh Mehr and M. J. S. Safari, "Genetic programming for streamflow forecasting," in *Advances in Streamflow Forecasting*, pp. 193–214, Elsevier, Amsterdam, Netherlands, 2021.

[9] A. Essien and C. Giannetti, "A deep learning framework for univariate time series prediction using convolutional LSTM stacked autoencoders," in *Proceedings of the IEEE International Symposium on Inovations in Intelligent Systems and Applications (INISTA)*, pp. 1–6, Sofia, Bulgaria, July 2019.

[10] Z. M. Yaseen, S. O. Sulaiman, R. C. Deo, and K.-W. Chau, "An enhanced extreme learning machine model for river flow forecasting: state-of-the-art, practical applications in water resource engineering area and future research direction," *Journal of Hydrology*, vol. 569, pp. 387–408, 2019.

[11] O. Oyebode and D. Stretch, "Neural network modeling of hydrological systems: a review of implementation techniques," *Natural Resource Modeling*, vol. 32, no. 1, Article ID 12189, 2019.

[12] F. Cui, S. Q. Salih, B. Choubin, S. K. Bhagat, P. Samui, and Z. M. Yaseen, "Newly explored machine learning model for river flow time series forecasting at Mary River, Australia," *Environmental Monitoring and Assessment*, vol. 192, no. 12, Article ID 761, 2020.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Y. Bai, N. Bezak, B. Zeng, C. Li, K. Sapač, and J. Zhang, "Daily runoff forecasting using a cascade long short-term memory model that considers different variables," *Water Resources Management*, vol. 35, no. 4, pp. 1167–1181, 2021.

[15] D. Couta, Y. Zhang, and Y. Li, "River flow forecasting using long short-term memory," *DEStech Transactions on Computer Science and Engineering*, vol. 16, 2019.

[16] A. Z. U. Din, Y. Ayaz, M. Hasan, J. Khan, and M. Salman, "Bivariate short-term electric power forecasting using LSTM network," in *Proceedings of the 2019 International Conference on Robotics and Automation in Industry (ICRAI)*, pp. 1–8, Rawalpindi, Pakistan, Octomber 2019.

[17] M. Jain, S. Manandhar, Y. H. Lee, S. Winkler, and S. Dev, "Forecasting precipitable water vapor using LSTMs," in *Proceedings of the 2020 IEEE USNC-CNC-URSI North American Radio Science Meeting (Joint with AP-S Symposium)*, Montreal, QC, Canada, July 2020.

[18] M. Rahimzad, A. Moghaddam Nia, H. Zolfonoon, J. Soltani, A. Danandeh Mehr, and H.-H. Kwon, "Performance comparison of an lstm-based deep learning model versus conventional machine learning algorithms for streamflow forecasting," *Water Resources Management*, vol. 35, 2021.

[19] B. B. Sahoo, R. Jha, A. Singh, and D. Kumar, "Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting," *Acta Geophysica*, vol. 67, no. 5, pp. 1471–1481, 2019.

[20] L. Yan, J. Feng, and T. Hang, "Small watershed stream-flow forecasting based on LSTM," *Advances in Intelligent Systems and Computing*, vol. 935, pp. 1006–1014, 2019.

[21] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[22] M. M. Hussain, S. H. Bari, I. Mahmud, and M. I. H. Siddiquee, "Application of different artificial neural network for streamflow forecasting," in *Advances in Streamflow Forecasting*, pp. 149–170, Elsevier, Amsterdam, Netherlands, 2021.

[23] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems*, vol. 31, no. 3, Article ID 2130001, 2021.

[24] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, "Stock market forecast using multivariate analysis with bidirectional and stacked (LSTM, GRU)," in *Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC)*, pp. 1–7, Riyadh, Saudi Arabia, April 2018.

[25] S. Kumar, L. Hussain, S. Banarjee, and M. Reza, "Energy load forecasting using deep learning approach LSTM and GRU in spark cluster," in *Proceedings of the 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pp. 1–4, Kolkata, India, January 2018.

[26] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons, and Fractals*, vol. 140, Article ID e110212, 2020.

[27] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between ARIMA, LSTM, and GRU for time series forecasting," in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 49–55, New York, NY, USA, December 2019.

[28] G. Papacharalampous, H. Tyralis, and D. Koutsoyiannis, "Univariate time series forecasting of temperature and precipitation with a focus on machine learning algorithms: a multiple-case study from Greece," *Water Resources Management*, vol. 32, no. 15, pp. 5207–5239, 2018.

[29] H. Tyralis and G. Papacharalampous, "Variable selection in time series forecasting using random forests," *Algorithms*, vol. 10, no. 4, Article ID 114, 2017.

[30] G. Papacharalampous, H. Tyralis, and D. Koutsoyiannis, "One-step ahead forecasting of geophysical processes within a purely statistical framework," *Geoscience Letters*, vol. 5, no. 1, Article ID 12, 2018.

[31] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.

[32] W. Niu and Z. Feng, "Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management," *Sustainable Cities and Society*, vol. 64, Article ID 102562, 2021.

[33] F. F. Abera, S. Arega, and B. H. Gedamu, "Climate change induced precipitation and temperature effects on water resources: the case of Borkena watershed in the highlands of Wollo, central Ethiopia," *Water Conserv Sci Eng*, vol. 5, no. 1, pp. 53–66, 2020.

[34] F. D. Mwale, A. J. Adeloye, and R. Rustum, "Infilling of missing rainfall and streamflow data in the Shire River basin, Malawi-a self organizing map approach," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 50-52, pp. 34–43, 2012.

[35] B. Mesta, O. B. Akgun, and E. Kentel, "Alternative solutions for long missing streamflow data for sustainable water resources management," *International Journal of Water Resources Development*, vol. 37, no. 5, pp. 882–905, 2021.

[36] F. B. Hamzah, F. Mohd Hamzah, S. F. Mohd Razali, and H. Samad, "A comparison of multiple imputation methods for recovering missing data in hydrological studies," *Civil Engineering Journal*, vol. 7, no. 9, pp. 1608–1619, 2021.

[37] A. Gnauck, "Interpolation and approximation of water quality time series and process identification," *Analytical and Bioanalytical Chemistry*, vol. 380, no. 3, pp. 484–492, 2004.

[38] D. K. Yawson, V. M. Kongo, and R. K. Kachroo, "Application of linear and nonlinear techniques in river flow forecasting in the Kilombero River basin, Tanzania," *Hydrological Sciences Journal*, vol. 50, no. 5, Article ID 796, 2005.

[39] A. K. Fleig, L. M. Tallaksen, H. Hisdal, and D. M. Hannah, "Regional hydrological drought in north-western Europe: linking a new regional drought area index with weather types," *Hydrological Processes*, vol. 25, no. 7, pp. 1163–1179, 2011.

[40] S. Gao, Y. Huang, S. Zhang et al., "Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation," *Journal of Hydrology*, vol. 589, Article ID 125188, 2020.

[41] C. Shen, "A transdisciplinary review of deep learning research and its relevance for water resources scientists," *Water Resources Research*, vol. 54, no. 11, pp. 8558–8593, 2018.

[42] S. ardabili Faizollahzadeh, A. Mosavi, and M. Dehghani, "Deep learning and machine learning in hydrological processes, climate change and earth systems: a systematic review," in *Engineering for Sustainable Future*Springer, Berlin, Germany, 2019.

[43] N. Hayatbini, B. Kong, K.-L. Hsu et al., "Conditional generative adversarial networks (cGANs) for near real-time precipitation estimation from multispectral GOES-16 satellite imageries-PERSIANN-cGAN," *Remote Sensing*, vol. 11, no. 19, 2019.

[44] S. Nacar, M. A. Hınıs, and M. Kankal, "Forecasting daily streamflow discharges using various neural network models and training algorithms," *KSCE Journal of Civil Engineering*, vol. 22, no. 9, pp. 3676–3685, 2018.

[45] A. Sahoo, S. Samantaray, and D. K. Ghose, "Stream flow forecasting in mahanadi river basin using artificial neural networks," *Procedia Computer Science*, vol. 157, pp. 168–174, 2019.

[46] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: a systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, Article ID 106181, 2020.

[47] W. Wang, P. H. A. J. M. V. Gelder, J. K. Vrijling, and J. Ma, "Forecasting daily streamflow using hybrid ANN models," *Journal of Hydrology*, vol. 324, no. 1, pp. 383–399, 2006.

[48] R. K. Mishra, G. Y. S. Reddy, and H. Pathak, "The understanding of deep learning: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2021, Article ID e5548884, 15 pages, 2021.

[49] ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, "Artificial neural networks in hydrology I: preliminary concepts," *Journal of Hydrologic Engineering*, vol. 5, no. 2, pp. 115–123, 2000.

[50] O. Sharma, "A new activation function for deep neural network," in *Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 84–86, Faridabad, India, February 2019.

[51] E. B. Wegayehu and F. B. Muluneh, "Multivariate streamflow simulation using hybrid deep learning models," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID e5172658, 16 pages, 2021.

[52] J. Yin, Z. Deng, A. V. M. Ines, J. Wu, and R. Eeswaran, "Forecast of short-term daily reference evapotranspiration under limited meteorological variables using a hybrid bi-directional long short-term memory model (Bi-LSTM)," *Agricultural Water Management*, vol. 242, Article ID 106386, 2020.

[53] Q. Zou, Q. Xiong, Q. Li, H. Yi, Y. Yu, and C. Wu, "A water quality prediction method based on the multi-time scale bidirectional long short-term memory network," *Environmental Science and Pollution Research*, vol. 27, no. 14, pp. 16853–16864, 2020.

[54] J. F. Torres, D. Gutiérrez-Avilés, A. Troncoso, and F. Martínez-Álvarez, "Random hyper-parameter search-based deep neural network for power consumption forecasting," *Advances in Computational Intelligence*, pp. 259–269, Springer, Berlin, Germany, 2019.

[55] J. Konar, P. Khandelwal, and R. Tripathi, "Comparison of various learning rate scheduling techniques on convolutional neural network," in *Proceedings of the 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pp. 1–5, Bhopal, India, February 2020.

[56] S. Aljahdali, A. Sheta, and H. Turabieh, "river flow forecasting: a comparison between feedforward and layered recurrent neural network," in *Innovation in Information Systems and Technologies to Support Learning Research*, pp. 523–532, Springer, Berlin, Germany, 2020.

[57] S. Kumar, T. Roshni, and D. Himayoun, "A comparison of emotional neural network (ENN) and artificial neural network (ANN) approach for rainfall-runoff modelling," *Civil Engineering Journal*, vol. 5, no. 10, pp. 2120–2130, 2019.

[58] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, "Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting," *Water*, vol. 12, no. 5, Article ID 1500, 2020.

[59] X. Zhao, H. Lv, S. Lv, Y. Sang, Y. Wei, and X. Zhu, "Enhancing robustness of monthly streamflow forecasting model using gated recurrent unit based on improved grey wolf optimizer," *Journal of Hydrology*, vol. 601, Article ID 126607, 2021.

[60] M. B. Gunathilake, C. Karunanayake, A. S. Gunathilake et al., "Hydrological models and artificial neural networks (ANNs) to simulate streamflow in a tropical catchment of Sri Lanka," *Applied Computational Intelligence and Soft Computing*, vol. 2021, Article ID e6683389, 9 pages, 2021.