Engineered Science
DOI: https://dx.doi.org/10.30919/es1047

Combining Long Short-Term Memory and Genetic Programming for Monthly Rainfall Downscaling in Southern Thailand's Thale Sap Songkhla River Basin

Sirimon Pinthong,[1] Pakorn Ditthakit,[1,*] Nureehan Salaeh,[1] Warit Wipulanusat,[2] Uruya Weesakul,[3] Ismail Elkhrachy,[4] Krishna Kumar Yadav[5, 6] and Nand Lal Kushwaha[7]

Abstract

This paper proposed a novel methodology that combines long short-term memory (LSTM) and genetic programming (GP) for downscaling monthly rainfall into watershed regions. The exploration of suitable downscaling models and the trend of monthly rainfall in the 2030s, 2060s, and 2080s in the Thale Sap Songkhla River basin (TSS) was investigated. The TSS is one of four major areas in Thailand's southern basin and has a tropical monsoon climate. The monthly rainfall observed by the Royal Irrigation Department (RID) from January 1993 to December 2018 (312 months) was available at three rainfall stations. Six machine learning techniques (*i.e.*, M5, RF, SVR, MLP, GP, and LSTM) were employed to downscale the monthly rainfall data from the general circulation models (GCMs) of CMIP5 (HadGEM2-ES and ACCESS1-3) and CMIP6 (HadGEM2-CGM31-LL and ACCESS-CM2) under the RCP4.5 (SSP245) and RCP8.5 (SSP585) scenarios. Since the TSS experiences significant differences in low and high rainfall for January–September and October–December, respectively, those data were analyzed separately in addition to using whole-year datasets. This study considered six common climate variables: precipitation (pr), maximum near-surface air temperature (tasmax), minimum near-surface air temperature (tasmin), relative humidity (hur), sea level pressure (psl), and near-surface wind speed (sfcWind). These variables were chosen based on the correlation between them and the observed rainfall data. The findings of this research indicate that when the LSTM and GP models are merged, they are the most efficient for downscaling monthly rainfall. The OI and r-value illustrate a highly robust relationship between the average values within the TSS watershed. These results offer valuable understanding of the clear strengths and limitations of every model category, which are influenced by factors such as the size of the data and the characteristics used in the model training process. Climate change is likely to have only a minor impact on rainfall patterns in the TSS in the near future, both in moderate and extreme emission scenarios. However, significant changes are expected in the later stages of this century (2060s and 2080s), particularly during the monsoon season, which experiences drastic shifts.

Keywords: Climate change; Genetic programming; Long short-term memory; Machine learning; Rainfall downscaling.
Received: 14 October 2023; Revised: 03 December 2023; Accepted: 04 December 2023.
Article type: Research article.

1. Introduction

Global climate change has been caused by the emission of greenhouse gases (GHGs) into the atmosphere over the past 150 years. Since the 19th century, the age of industrialization, many natural resources have been used in construction, and this is another critical factor contributing to global climate change.[1] Trends show the influence of humans on the global water cycle since 1950.[2] The Intergovernmental Panel on Climate Change (IPCC)[3] reported increasing precipitation in South America, Africa, East Asia, South Asia, Australia, and Southeast Asia. Eckstein *et al*.[4] showed that from 2000 to 2019 (20–year period), several Southeast Asian countries, including Myanmar, the Philippines, and Thailand, were the 2nd, 4th, and 9th most affected countries in the world, respectively.

[1] *Center of Excellence in Sustainable Disaster Management, School of Engineering and Technology, Walailak University, Nakhon Si Thammarat 80161, Thailand.*
[2] *Thammasat University Research Unit in Data Science and Digital Transformation, Department of Civil Engineering, Faculty of Engineering, Thammasat School of Engineering, Thammasat University, Pathumthani 12120, Thailand.*

Changes in global temperature, rainfall patterns, and solar radiation affect agricultural productivity.[5,6] Global food security has become one of the greatest problems facing humanity in the 21st century.[5,7] The Thale Sap Songkhla River basin (TSS), a major part of Thailand's southern basin, is located in the Nakhon Si Thammarat, Songkhla, and Phatthalung provinces. The TSS is an area with extremely high rainfall variability. Most of southern Thailand, except for forest areas, is primarily used for agriculture, which is the area's main economic activity. For the three provinces in the TSS area, approximately 50% of the province's land area is dedicated to agriculture, some of which is rain-dependent agriculture.[8] Seasonal patterns and the distribution of rainfall are important parameters, so their variation will affect the seasonal yield both quantitatively and qualitatively. The 40-year rainfall statistics for Thailand (from 1981 to 2020) show that the maximum and minimum rainfall have more alternating variability each year. This raises concerns about food insecurity in this area since the negative impact of climate change on agricultural productivity leads to many problems for the local people. Therefore, determining the precise trends in future rainfall increases or decreases is essential for water system management. However, no empirical study has been conducted to estimate future rainfall under the sixth phase of the Coupled Model Intercomparison Project (CMIP6) at the TSS.

General circulation models (GCMs) are the best tools for projecting future global climate change with the continued increase in GHGs in the atmosphere. Currently, the IPCC has updated the Fifth Assessment Report (AR5) to the Sixth Assessment Report (AR6), which is the latest in a series of reports on climate change. The IPCC demonstrated an improvement in the outcomes of future scenarios used to predict climate evolution under CMIP6, overcoming and improving the limitations identified in the outcomes under the Fifth phase of the Coupled Model Intercomparison Project (CMIP5). CMIP6 is reported with a new set of emission scenarios called shared socioeconomic pathways (SSPs), which replace CMIP5's older set of emission scenarios called representative concentration pathways (RCPs). These data were used to calculate the CO_2 equivalent of GHGs under different future emission scenarios, considering their individual forcing values and lifetimes in the atmosphere.[9-11] The SSPs and RCPs are aligned as follows: SSP126-RCP2.6, SSP245-RCP4.5, SSP460-RCP6.0, and SSP585-RCP8.5.[12,13] In general, GCMs divide the atmosphere and ocean into grids with a horizontal resolution of 2 to 48 degrees of latitude and 10 to 20 degrees of longitudinal layers in the vertical direction that cover an area of approximately 200 to 650 km at ground level, which is insufficient for local region or watershed studies.[9,14] To increase the resolution and match the local scale, these model outputs must be downscaled. Downscaling from GCMs to more granular data can be categorized into two methods: statistical downscaling (SD) and dynamic downscaling (DD). Although regional climate models (RCMs), which are DD methods, show detailed results up to 10-50 km, they are expensive and require intense data to run.[15,16] SD is the relationship between large-scale climate variables and past local climate variables. Compared with the DD method, the SD method is widely used because of its low cost and shorter runtime.[16-19] Anh and Taniguchi[20] downscaled six-hourly precipitation in the Red River Delta in Vietnam using an artificial neural network (ANN) and RCM. The outcomes of this study illustrate that ANN can produce RCM-like results ($r > 0.9$) at a fraction of the cost, with an 89% reduction in the required computational power.

In recent years, many scholars have increasingly used machine learning (ML) to downscale climate data, which has shown high efficiency. Han *et al.*[21] compared the performance of ML (multilayer perceptron: MLP and support vector machine: SVM) with that of the statistical downscaling model (SDSM) for downscaling daily precipitation in Shenzhen city, China. The results showed that ML outperformed SDSM. Jose *et al.*[22] showed that long short-term memory (LSTM) was more effective at downscaling rainfall than multiple linear regression (MLR), SVM, extra tree regressor (ETR), and random forest (RF) methods in a case study in the Netravati Basin, India. Tran Anh *et al.*[23] downscaled daily rainfall in the Vietnamese Mekong Delta using long short-term memory (LSTM) and feedforward neural networks (FNNs). The results showed that the LSTM and FNN models had r values > 0.9 and > 0.7, respectively. Dey *et al.*[24] used an ANN, a random forest (RF), and a support vector machine (SVM) versus the simple arithmetic mean for downscaling rainfall and maximum and minimum

[3] *Thammasat University Research Unit in Climate Change and Sustainability, Department of Civil Engineering, Faculty of Engineering, Thammasat School of Engineering, Thammasat University, Pathumthani 12120, Thailand.*

[4] *College of Engineering, Civil Engineering Department, Najran University, Najran 66291, Saudi Arabia.*

[5] *Faculty of Science and Technology, Madhyanchal Professional University, Ratibad, Bhopal, 462044, India.*

[6] *Environmental and Atmospheric Sciences Research Group, Scientific Research Center, Al-Ayen University, Thi-Qar, Nasiriyah, 64001, Iraq.*

[7] *Division of Agricultural Engineering, ICAR-Indian Agricultural Research Institute, New Delhi 110012, India.*

**Email:* dpakorn@mail.wu.ac.th (P. Ditthakit)

temperatures over the Damodar River basin, India. Their results showed that RF and SVM outperformed each other. Furthermore, existing models, including SVR,[11,25] RVM,[26] GP[19,27] and CNN[28], have been employed for downscaling climate data.

This study has two main goals: first, to explore suitable models for downscaling monthly rainfall and, second, to determine the trend of monthly rainfall in the 2030s, 2060s, and 2080s in the TSS. Machine learning techniques (*i.e.*, M5, RF, SVR, MLP, GP, and LSTM) were employed to downscale the monthly rainfall data from the GCM models of CMIP5 (HadGEM2-ES and ACCESS1-3) and CMIP6 (HadGEM2-CGM31-LL and ACCESS-CM2) under the RCP4.5 (SSP245) and RCP8.5 (SSP585) scenarios, achieving resolution at the watershed level studied in this study. To enhance the efficiency of the study, data were analyzed separately for the January-September (Jan-Sep) and October-December (Oct-Dec) periods, as the TSS experiences significant differences in low and high rainfall during these periods. However, selecting an appropriate dataset for ML model training was a challenge due to the limited data availability. Therefore, this study focused on identifying suitable models that can be utilized to address the limitations and complexities associated with investigating TSS.

2. Materials and methods

2.1 Description of the study area and data analysis

2.1.1 Study area and observed data

One of the four major river basins in southern Thailand, the Thale Sap Songkhla River basin (TSS), was selected for this research. It is situated between 6°45' and 8°00' north latitude and between 99°30' and 100°45' east longitude. This area is approximately 8.484.35 km^2. It is approximately 150 km long (from north to south) and 65 km wide (from east to west). The research area and the locations of the rainfall stations used in this research are indicated in Fig. 1. According to the Köppen climate classification, the climate of the TSS is tropical monsoon (Am).[29] The TSS has a year-round high-temperature atmospheric environment, with the highest temperatures occurring in April and the lowest occurring in October. There is extreme rainfall in the wet season that lasts 8 to 9 months, with only 2 to 3 months having rainfall less than 62 mm. The TSS is influenced by the southwest (S/W) and northeast (N/E) monsoons, which start from May to January and are wettest in November. The dry season begins in April and lasts until May. The annual rainfall is approximately 2,000 mm, of which more than 50% falls during the northeast monsoon (N/E) season.[30,31]

In this study, monthly rainfall data were collected from the Royal Irrigation Department (RID) for the period spanning from January 1993 to December 2018, encompassing 312 months. These data were sourced from three rainfall monitoring stations within the TSS. The monthly and annual

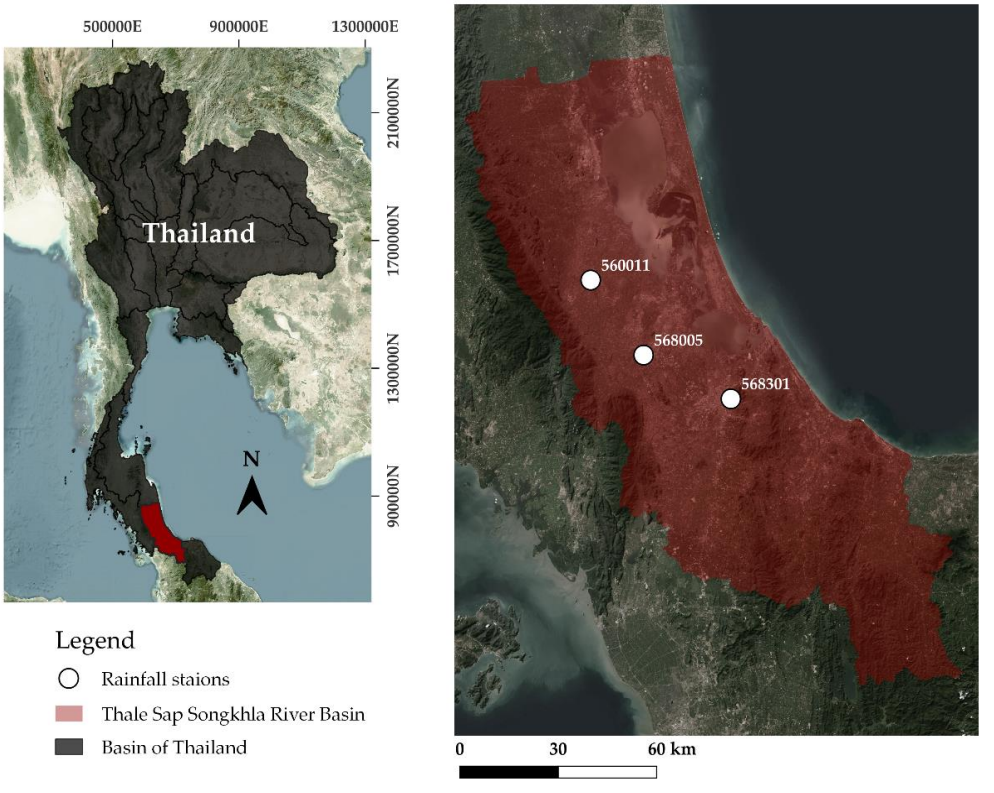

Fig. 1 Locations of the three rainfall stations in the Thale Sap Songkhla River basin.

statistics, including coordinates, are shown in Table 1 and Fig. 2. However, missing data were identified in the collected data. To obtain more complete data, the missing monthly rainfall for each station was imputed according to the imputation method suggested by Pinthong *et al*.[32], who conducted a study on a suitable method for imputing monthly rainfall in the TSS.

2.1.2 Climate projection

Variable selection that correlates with rainfall is essential for effectively downscaling data. Ghosh[33] stated that data downscaling should consider the reliability of the GCM, the availability of output data files from the GCM, and the strong correlation between variables and ground station data (observed rainfall). In this research, six commonly used climate variables were chosen, *i.e.*, precipitation (pr), maximum near-surface air temperature (tasmax), minimum near-surface air temperature (tasmin), relative humidity (hur), sea level pressure (psl), and near-surface wind speed (sfcWind).[34-38] The statistical values of historical data from CMIP5 and CMIP6 are shown in Fig. 3.

Selecting suitable GCMs for projecting meteorological data on a regional scale is challenging. "Which GCM is the most suitable?" remains an unanswered question.[39] Although the resolutions of GCMs differ, a higher-resolution GCM does not necessarily yield better predictions.[40] In this study, the historical and future projections of GCMs under CMIP5 and CMIP6 were used, *i.e.*, HadGEM2-ES and ACCESS1-3 under CMIP5 and HadGEM2-CGM31-LL and ACCESS-CM2 under CMIP6. These models have an atmospheric resolution of 1.875° x 1.25° horizontally, 38 levels vertically, and an atmospheric resolution of approximately 40 km. Climate data were projected with GCMs using the Earth System Grid Federation (ESGF), hosted by the Department of Energy's Lawrence Livermore National Laboratory (DOE/LLNL), which supports current CMIP5 activities (https://esgf-node.llnl.gov/projects/cmip5/). For CMIP6, projections were made through the ESGF (https://esgf-node.llnl.gov/projects/cmip6/) and the Copernicus Climate Change Service Climate Data Store (https://cds.climate.copernicus.eu/).

All climate variables were projected under two greenhouse gas concentration scenarios: RCP4.5 and RCP8.5 under CMIP5 and SSP245 and SSP585 under CMIP6. Two scenarios, which are famous for projecting climate data, were selected for this research.[25,41-44] They are the "medium" and "high" situations of future climate projections. The use of RCP2.6 (SSP126) and RCP6.0 (SSP460) was excluded. Because RCP2.6 is in a "low" situation, it has an insignificant effect on the change,[23] and RCP6.0 is in a "medium" situation. It is close to RCP4.5 and falls between RCP4.5 and RCP8.5.[45]

Table 1. Rainfall data characteristics and the geographic locations of the three rainfall stations.

No.	Code station	Coordinates		Monthly (mm)				Annual (mm)
		Latitude	Longitude	Max	Min	Avg	SD	
1	#560011	7.28	100.05	1282.70	0.00	180.79	199.18	2068.12
2	#568005	7.13	100.30	977.60	0.00	164.25	171.21	1971.06
3	#568301	7.02	100.50	964.70	0.00	186.39	176.88	2236.66

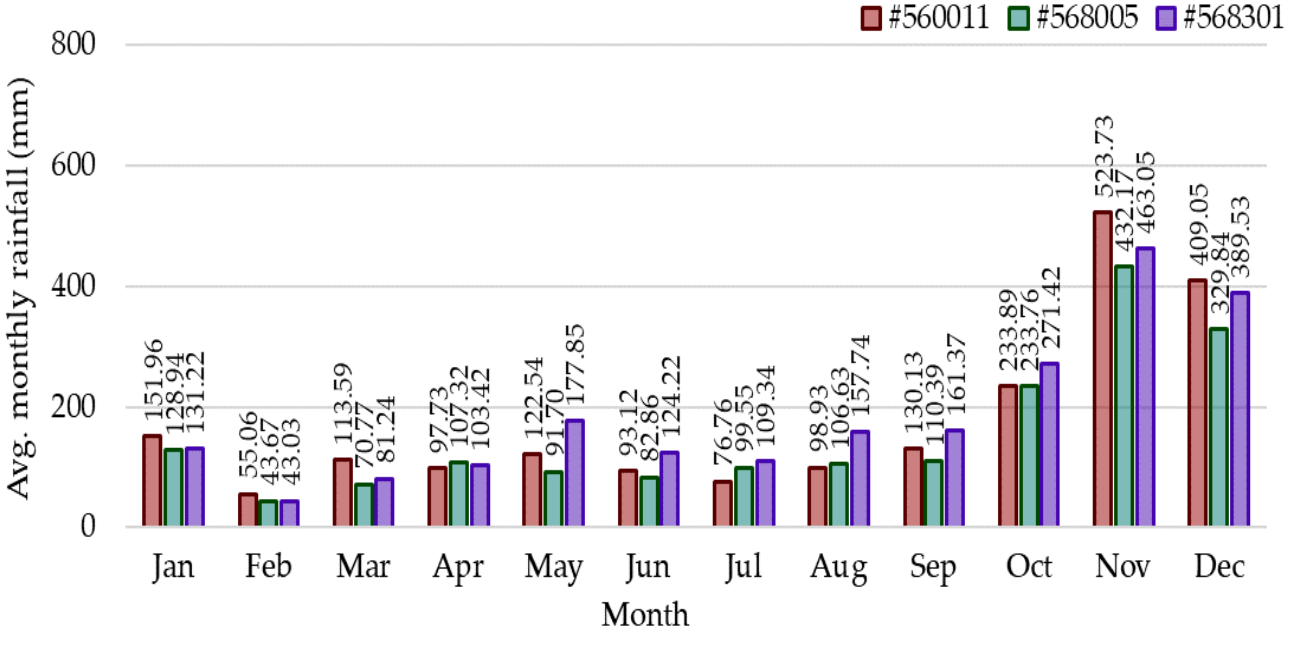

Fig. 2 Average monthly rainfall at three stations in the TSS.

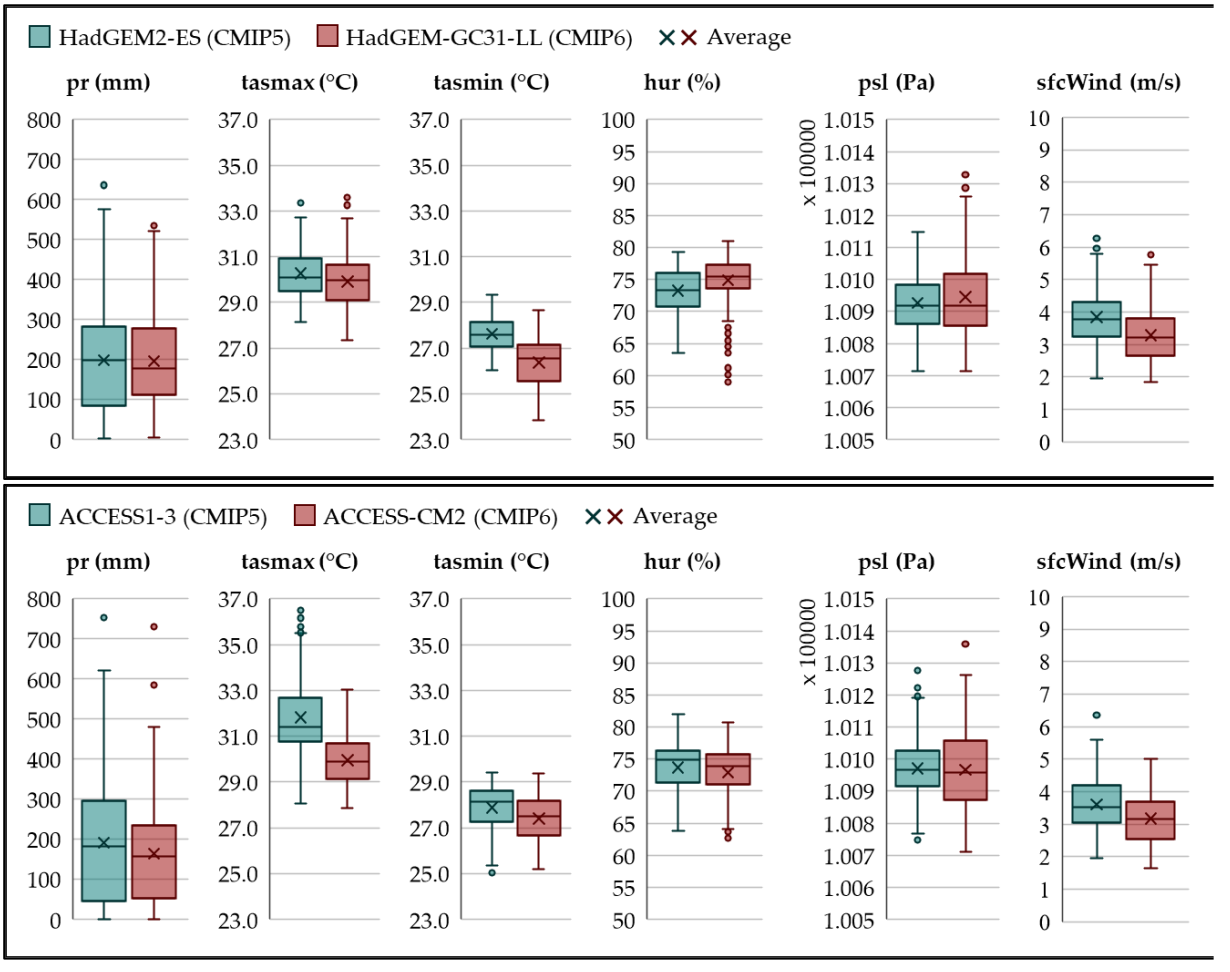

Fig. 3 The statistical values of the projected data from two GCMs (HadGEM2-ES and ACCESS1-3 under CMIP5 and HadGEM2-CGM31-LL and ACCESS-CM2 under CMIP6).

2.2 Models used for downscaling

2.2.1 M5 model tree (M5)

Quinlan[46] introduced M5 for the first time. The model builds a relationship between independent and dependent variables using a binary decision tree with linear regression functions at the terminal (leaf) nodes. Regression functions with few variables involved are simple to understand and straightforward.[47] Building a decision tree for the modeling steps is similar to creating a large tree-like structure. The optimal attribute is selected by using the standard deviation reduction (SDR) (Equation 1) for each attribute at a subinstance (which gives the highest SDR value).[48] To have a structure that fits properly and avoids overfitting, this model is similar to pruning an overgrown tree, and a linear regression function is used to replace the trimmed subtree.

$$\mathrm{SDR} = \mathrm{sd}(\mathrm{T}) - \sum \frac{|\mathrm{T_i}|}{|\mathrm{T}|} \mathrm{sd}(\mathrm{T_i}) \quad (1)$$

where T is a set of examples that reach the node, T_i is the subset of examples that have the i^{th} outcome of the potential set, and sd is the standard deviation.[49]

2.2.2 Random Forest (RF)

RF is a type of ensemble learning algorithm used for classification, regression, and other tasks. During training, multiple decision trees are created. The first version of this algorithm was created by Ho[50] and later developed by Breiman.[51] Ensemble learning models such as RF train multiple instances of a model on the same dataset, allowing each predictor model to learn as independently as possible. This is achieved by allowing each instance to learn from different data, which are selected randomly using a method called bootstrap aggregating (bagging) (shown in Fig. 4).

The RF model follows these steps. First, bagging is used to randomly select instances from a dataset to create different datasets based on the number of decision trees in the RF. Each instance chooses 2/3 of the data, and the remaining 1/3 of the dataset is used as an out-of-bag error (OOB) instance, which is intended to measure the prediction error. Then, a model decision tree is built for each instance. Finally, the results from each model are aggregated through bagging by either majority voting in classification or averaging in regression.

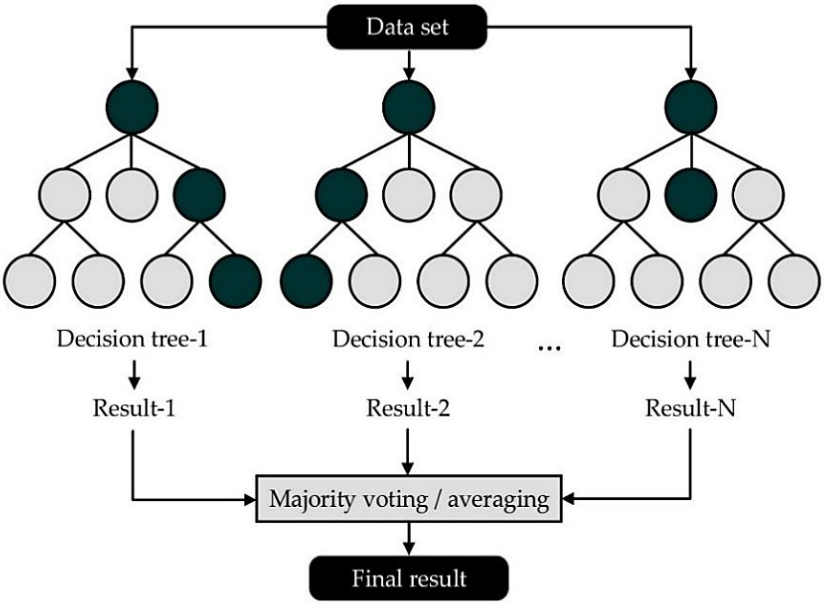

Fig. 4 Schematic of the random forest. Reproduced with permission from Ref. [52].

2.2.3 Support Vector Regression (SVR)

Support vector regression (SVR) is a supervised learning model used for regression analysis and was developed by Vapnik *et al*.[53]. SVR is a data-driven model that utilizes associated learning algorithms to analyze data. SVR is based on the support vector machine (SVM) concept and is a popular ML model used for classification problems where the data are not linearly separable. The goal of SVR is to predict a linear relationship between the input vector ($x \in Rn$) and the output variable ($y \in R$). During training, SVR calculates the weights and bias based on the input and output data. Then, this information is used to estimate the output results from the input data during testing. SVR performs two primary functions: first, it estimates the prediction errors during training, and second, it calculates the output values using the weight, bias, and input data.[26]

To predict the output values based on the input vectors, the SVR technique uses the ε-insensitive loss function, which is a type of loss function that creates an epsilon tube. This loss function was introduced by Vapnik[54] and is shown in Fig. 5. The SVR model is then formulated as the following convex optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} W^T \cdot W + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \quad (2)$$

$$\text{Subject to } \begin{matrix} y_i - [W^T \cdot x_i + b] \le \varepsilon - \xi_i & ; \xi_i, \xi_i^*, 0, i = 1,2,\ldots,l \\ [W^T \cdot x_i + b] - y_i \le \varepsilon - \xi_i^* & ; \xi_i, \xi_i^*, 0, i = 1,2,\ldots,l \end{matrix} \quad (3)$$

where ξ_i, ξ_i^* are the slack variables, C is a fixed integer that is the cost when an error occurs, ε is the error of the training dataset, and l is the number of support vectors. A nonlinear regression using the Kernel function can be written as

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) k(x_i, x_j) + b \quad (4)$$

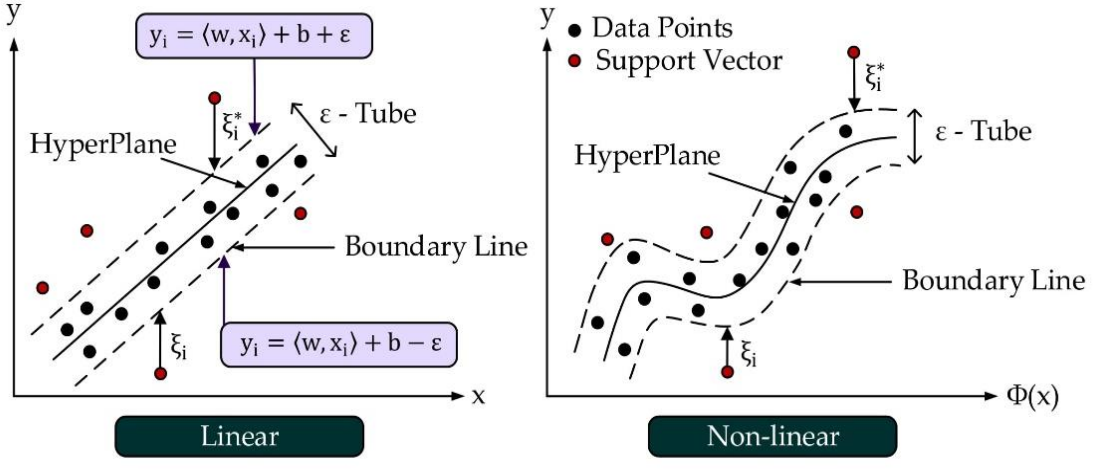

Fig. 5 Schematic of the support vector regression model. Reproduced with permission from Refs. [55] and [56].

where b is the bias, α_i, α_i^* are Lagrange multipliers, and $k(x_i, x)$ is the Kernel function. The popular Kernel functions are:

Polynomial kernel

$$k(x_i, x_j) = (1 + x_i \cdot x_j)^d \quad (5)$$

Radial Basis Function kernel (RBF kernel)

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (6)$$

2.2.4 Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is a type of feedforward artificial neural network that falls under the branch of artificial intelligence known as an ANN. ANNs have been around since the work of McCulloch and Pitts,[57] with the first era focusing on the development of artificial neurons until limitations were identified by Minsky and Papert.[58] The second era began with the discovery and dissemination of backpropagation algorithms for training, as noted by Dawson and Wilby,[59] McClelland, *et al.*[60]

ANNs have the same processing structure and function as the living brain, adapting to inputs. An MLP is a well-known type of ANN consisting of multiple layers of nodes with connections between the input and hidden layers and between the hidden layer and output (shown in Fig. 6).[34] It is particularly effective in handling complexity and works with supervised learning systems and backpropagation to train the return process. This process involves a forward pass and backward pass, with the sums of the input neurons computed, the activation function used to compute the output neurons, and the error signal propagated backward through the network while adjusting the weights to minimize the overall error. These steps are repeated until the overall error is sufficiently small.[25,61,62]

2.2.5 Genetic Programming (GP)

Genetic programming (GP) is a programming technique created by Koza and Rice,[63] which is based on Darwin's evolutionary theory. GP emulates the biological principle of natural selection, where the fittest organisms are better equipped to survive. In GPs, living organisms are modified to create more suitable genetic operators by modifying chromosomes into better genes. To determine the optimal solution, GP uses chromosomes to represent the answer and improves individual solutions via various evolutionary operations. The structure of a GP is a tree consisting of a set of functions and a set of terminals. The terminal is represented by a function that has no arguments or attributes.

To calculate the fitness solution, GP follows these steps:

I. In each generation, genes are randomly assigned to the initial chromosome. Suitable genes are selected for transfer to the next generation based on their fitness value, which is determined through a fitness function.

II. After selecting suitable genes, pairs of genes are used to create the next generation through either mutation or crossover. Since good-quality genes from the previous generation are selected, theoretically, better-quality genes can be obtained in the next generation.

III. Steps I and II are repeated until the best result is obtained and the specified conditions are met.

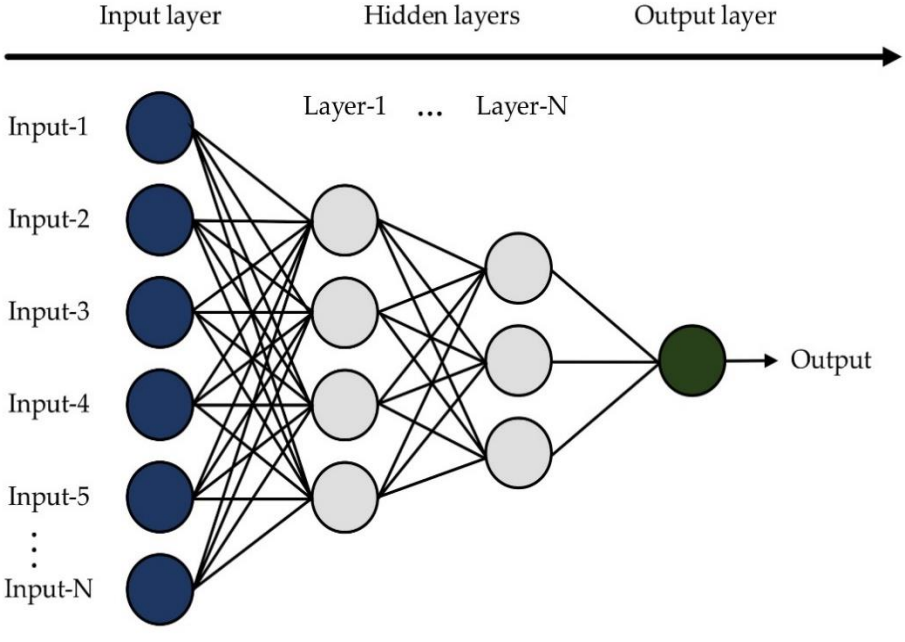

Fig. 6 A multilayer perceptron with two hidden layers. Reproduced with permission from [56].

Overall, GP is an evolutionary algorithm that simulates natural selection to find an optimal solution. It achieves this by iteratively creating better generations of chromosomes through selection, mutation, and crossover processes until the desired result is achieved.

2.2.6 Long short-term memory

Long short-term memory (LSTM) is a new type of network derived from RNNs. Although RNNs can handle historical data for a brief period, they struggle to learn patterns from long-term dependencies. As a result, backpropagation may become problematic, and the vanishing gradient problem may occur. The LSTM technique was developed specifically to address the shortcomings of traditional RNNs in learning long-term dependencies. The functional part of the memory cell state in LSTM has a "gate" unit that controls the data entering each node. This gate is composed of a forget gate, an input gate, and an output gate. The forget gate determines whether the information coming into the cell state should be stored or discarded, while the input gate receives new information and decides whether to update or maintain the existing value. Finally, the output gate determines whether to reveal or conceal the information. As a result, LSTM can learn from sequential data, retaining or discarding information as needed.[64] The LSTM structure is shown in Fig. 7, and the formulas are as follows:

Forget gate

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f) \quad (7)$$

Input gate

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i) \quad (8)$$

Cell state candidates

$$\tilde{C}_t = tanh(W_c \cdot [x_t, h_{t-1}] + b_c) \quad (9)$$

Cell state

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (10)$$

Output Gate

$$O_t = (W_o \cdot [x, h_{t-1}] + b_o) \quad (11)$$

Hidden state

$$h_t = O_t * tanh\,(C_t) \quad (12)$$

where W and b are the weight matrices and bias, respectively, and σ and tanh are the activation functions of the logistic sigmoid function. The internal operations of the LSTM can be adjusted to suit the needs of each specific task.[55] The neural network receives three inputs: the input at the current time step (x_t), the output from the previous LSTM unit (h_{t-1}), and the memory of the previous unit (C_{t-1}). The current network outputs two values: the output at the current time step (h_t) and the memory of the current unit (C_t).

Sigmoid (σ), tanh, and Relu are the most commonly used activation functions in ANNs and deep learning. The sigmoid is an S-curve function that suffers from the vanishing gradient problem.[65] reported that tanh has the same S-curve as sigmoid but with output values between -1 and 1, making the derivatives more durable, although the vanishing gradient problem still occurs in deeper networks. ReLU is a linear function that expands the range from 0 to ∞ and overcomes the vanishing gradient problem, allowing faster model training.[66] However, it cannot update negative weights.

2.3 Performance criteria metrics

a. Pearson's correlation coefficient (r) describes the similarity between the measured and predicted values. The r value is between -1 and 1. It is an index of the degree of the linear relationship between the data. There is no linear connection if r = 0. A perfect positive or negative linear connection exists if r = 1 or -1. In other words, the smaller the error is. In general, r values greater than 0.5 are often regarded as acceptable.[67-69]

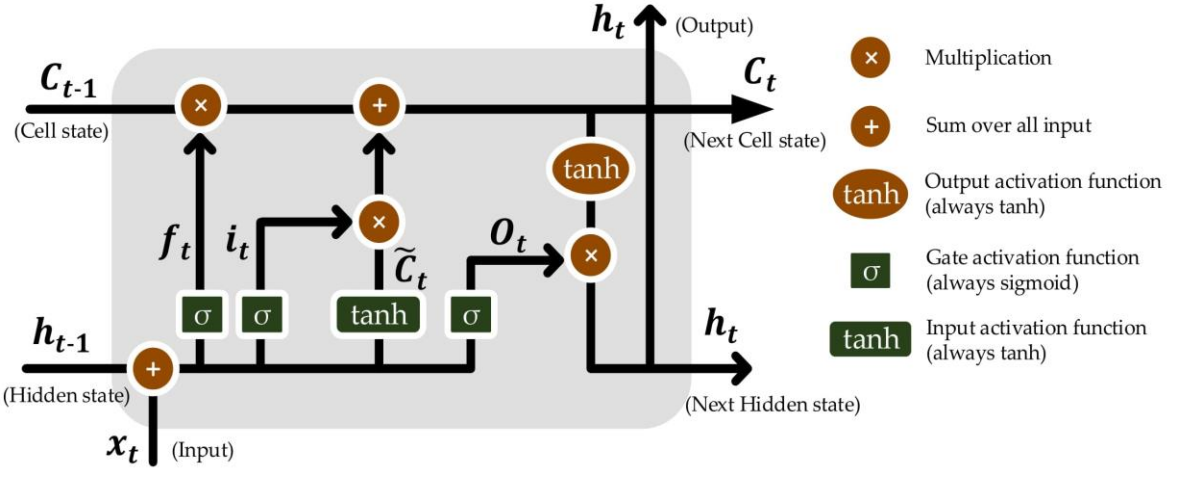

Fig. 7 Long short-term memory (LSTM) neural network structure. Reproduced with permission from [22].

b. The root mean square error (RMSE), the standard deviation of the residuals (prediction errors), represents the difference between the actual and calculated values and has the same unit as the dataset.[70] The RMSE is a standard benchmark for model performance in meteorology, air quality, and climate research studies.[71] A lower RMSE indicates excellent model performance. However, the RMSE is always nonnegative. Therefore, although zero is the best value for the RMSE, it is practically impossible.[72,73]

c. The overall index (OI), which is used to monitor the effectiveness of the model, combines the RMSE and model efficiency (ME) indicators.[74] The model's OI value refers to how well the observed and predicted data fit together. It is between -∞ and 1. A greater OI closer to 1 indicates that the model performs optimally.

d. Relative peak error (PE) is a measure used to estimate the peak data-specific error.[75] PE represents a good prediction when the value is low. In terms of rainfall data, the peak is vital because rainfall has a high variation or dispersion of a set of values; therefore, the maximum and minimum are quite different. PE indicates that the model is good at predicting high-range data and is valuable for implementing a trained model.

$$r = \frac{\sum_{i=1}^{n}(X_{obs}-\bar{X}_{obs})(X_{cal}-\bar{X}_{cal})}{\sqrt{\sum_{i=1}^{n}(X_{obs}-\bar{X}_{obs})^2}\cdot\sqrt{\sum_{i=1}^{n}(X_{cal}-\bar{X}_{cal})^2}} \quad (13)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs}-X_{cal})^2}{n}} \quad (14)$$

$$OI = \frac{1}{2}\left[2 - \frac{RMSE}{X_{obs,max}-X_{obs,min}} - \frac{\sum_{i=1}^{n}(X_{cal}-X_{obs})^2}{\sum_{i=1}^{n}(X_{obs}-\bar{X}_{obs})^2}\right] \quad (15)$$

$$PE = \frac{\sum_{i=1}^{n}|X_{obs,peak}-X_{cal,peak}|}{\sum_{i=1}^{n}X_{obs,peak}} \times 100 \quad (16)$$

Where X_{obs} is the observed data value, X_{cal} is the calculated data value, and n is the data point number.

3. Experimental design

The experimental design for this study consisted of three primary stages: projecting climate data, downscaling rainfall data, and estimating future rainfall data (shown in Fig. 9).

3.1 Projecting climate data

In the first stage, climate data were predicted, as detailed in Section 2.1.2. As a data preparation step, the projected data were extracted, and the coordinates of the rainfall stations were utilized to identify the GCM grids. A 12-month lag for climate variables (t-1 to t-12) was established, resulting in 300 data points for CMIP5 (1994 to 2018) and 252 data points for CMIP6 (1994 to 2014). The reason for the smaller data range in CMIP6 is that the historical data predicted by the GCMs only extend to 2014.

3.2 Downscaling rainfall data

This study employed various machine learning algorithms, namely, M5, RF, SVR-poly, SVR-rbf, MLP, GP, and LSTM, to downscale the monthly rainfall data. The modeling process was conducted using the software tools WEKA 3.9.5, GPdotNET, and ANNdotNET (available at https://github.com/bhrnjica). The downscaling stage was divided into four substeps:

Step I: Selecting the right inputs is a crucial step in developing machine learning models. Choosing the optimal input variables helps prevent overburdening the model with unnecessary data. Conducting a preliminary analysis of the dataset is essential for excluding redundant, inconsistent, or irrelevant data from the model.[55] The accuracy of the model, learning time, and sample size all depend on the selection of input variables.[76]

In this study, the correlation coefficient (r-value) is used as a criterion for selecting the input variables. It is a simple method that allows the identification of variables most closely associated with the output variable (observed rainfall). The GCM projection in this study includes six variables, including lagged variables correlated with the output variable. Typically, there are no exact established criteria for selecting inputs based on the r value, but an r value greater than 0.35 or less than -0.35 is used to determine the input variables. These criteria are selected based on the characteristics of the available data. Variables with a correlation greater than the specified r value are included as input data for the ML model's training. However, when training LSTM models, input selection is not necessary. LSTM models are complex and intelligent models capable of handling input data autonomously. Therefore, the input selection step can be skipped when training LSTM models.

Step II: Tuning the hyperparameters is a crucial step in training machine learning models. Hyperparameters are adjustable parameters that are set before training and play a critical role in regulating the learning process, determining the algorithm's functionality, and shaping the model's structure. By customizing these parameters to suit the specific machine learning problem of each dataset, researchers can improve the accuracy, robustness, and generalizability of their models. Consequently, suitable hyperparameters can differ across datasets, and thus, it is essential to tune them for each problem to achieve optimal results[77].

Step III: The dataset selected for ML training was divided into two rainfall periods: January-September (Jan-Sep) and October-December (Oct-Dec). ML models trained under small

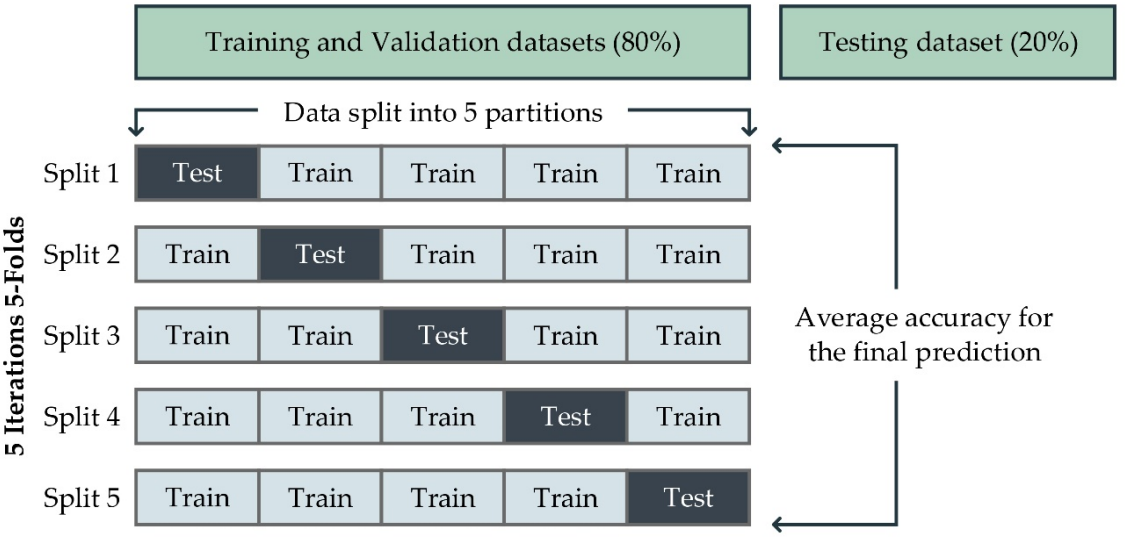

Fig. 8 The concept of machine learning cross-validation methods.

data constraints during the 3-month period of Oct-Dec had limitations. This raised the question of whether completed or split datasets were more effective for training. In this study, one sample station from the CMIP5 dataset, which is the largest dataset, was used as a test case. Model training was divided into two approaches: 1) using the completed dataset to train all the data and evaluate performance separately for Jan-Sep and Oct-Dec and 2) using the split dataset to divide the data into Jan-Sep and Oct-Dec before model training. Since the model training involved a different number of data points, 5-fold cross-validation was used to ensure greater reliability of the model's performance (as shown in Fig. 8). For the testing stage, 20% of the data (60 data points) were selected, resulting in 240 data points for the model training stage with the completed dataset. When training the model with the split dataset, there were 180 data points for Jan-Sep and 60 data points for Oct-Dec.

Step IV: The model training step involved training a suitable model to downscale the monthly rainfall data in the TSS. To accomplish this, a completed dataset was used, and a hold-out data method was employed, where 70% of the initial data were allocated for the training stage and the remaining 30% were allocated for the testing stage. In the case of CMIP5, the training stage comprised 210 data points, while there were 90 data points for the testing stage. For CMIP6, there were 176 data points for the training stage and 76 data points for the testing stage. The performance of the trained model was evaluated separately for the Jan-Sep and Oct-Dec periods to identify the most efficient model for each period using the performance criteria r, RMSE, OI, and PE. Subsequently, the best model for each period was combined to estimate the future monthly rainfall.

3.3 Estimating future rainfall data

In the final stage of the study, suitable trained models were employed to perform a downscaling of future monthly rainfall. This allowed the general trend of average monthly rainfall, including year-round, Jan-Sep, and Oct-Dec periods, to be ascertained. This study encompassed the near-future (2030s), middle-future (2060s), and far-future (2080s) periods under the RCP4.5 (SSP245) and RCP8.5 (SSP585) scenarios.

4. Results and discussion

4.1 Downscaling model training

4.1.1 Input selection

Figure 10 illustrates the relationship between the output variable (observed rainfall) and lagged variables in the projected CMIP5 dataset for sample station #560011. A strong positive correlation between the observed rainfall and lagged variables under CMIP5 and pr at t to t-2 and t-12 was revealed by the analysis. Additionally, a strong negative correlation with t-8 and t-9 was found, which aligns with the data predicted under CMIP6. However, a discrepancy in pr between HadGEM2-ES and HadGEM-GC31-LL at t-3 and t-4, indicating opposite directions, was observed. In terms of tasmax and tasmin, a robust positive correlation at t-5 to t-8 was observed, while a strong negative correlation was found between t-10 and t-12. For hur, a strong positive correlation was identified at t-2, while t-7 and t-9 exhibited a strong negative correlation. With respect to psl, ACCESS1-3 under CMIP5 exhibited a poor correlation with no lagged variables selected, whereas HadGEM2-ES showed a strong positive

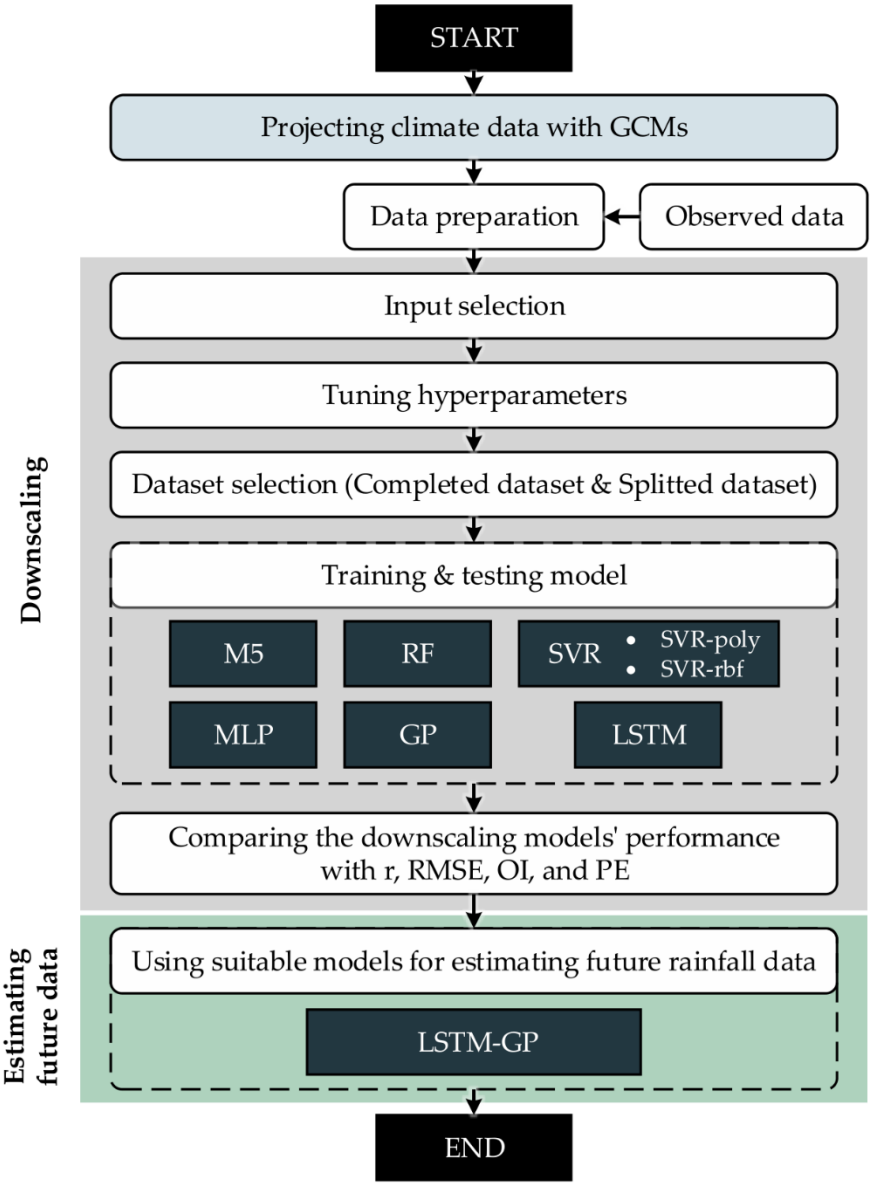

Fig. 9 The framework for downscaling monthly rainfall and estimating future rainfall data.

correlation at t-1 and a strong negative correlation with t-4 and t-5. In contrast, no lagged variables were used as input variables in CMIP6 because the correlation was insufficient. Regarding sfcWind, only ACCESS1-3 at t-11 was selected as the input variable under CMIP5. In contrast, under CMIP6, a robust positive correlation at t-4, t-5, and t-10 and a strong negative correlation at t-7 and t-8 were observed. The correlation trend between the output variable (observed rainfall) and lagged variables in #568005 and #568301 is similar to that of sample station #560011.

Regarding input variable selection, the numbers of variables selected for stations #560011, #568005, and #568301 under CMIP5 were 36, 34, and 36, respectively. For CMIP6, the corresponding numbers of input variables were 39, 39, and 29, respectively.

After conducting the observations, it was found that lagged climatic variables exhibited a relationship with rainfall during different months. Notably, most of the strong correlations were observed during the mid-year period, ranging from t-5 to t-8, as well as toward the end of the year, between t-11 and t-12. A positive correlation within the t to t-2 range was identified, which gradually transitioned into a negative correlation around t-6 to t-8 before re-entering the positive correlation phase during t-11 and t-12.

These findings regarding the relationships between climate variables and rainfall have implications, suggesting that a lengthy lag time of 12 months should be considered. Although this may be challenging in some studies because of factors such as the number of climatic variables, the intensity of the method, the test situation, or the correlation method, it is

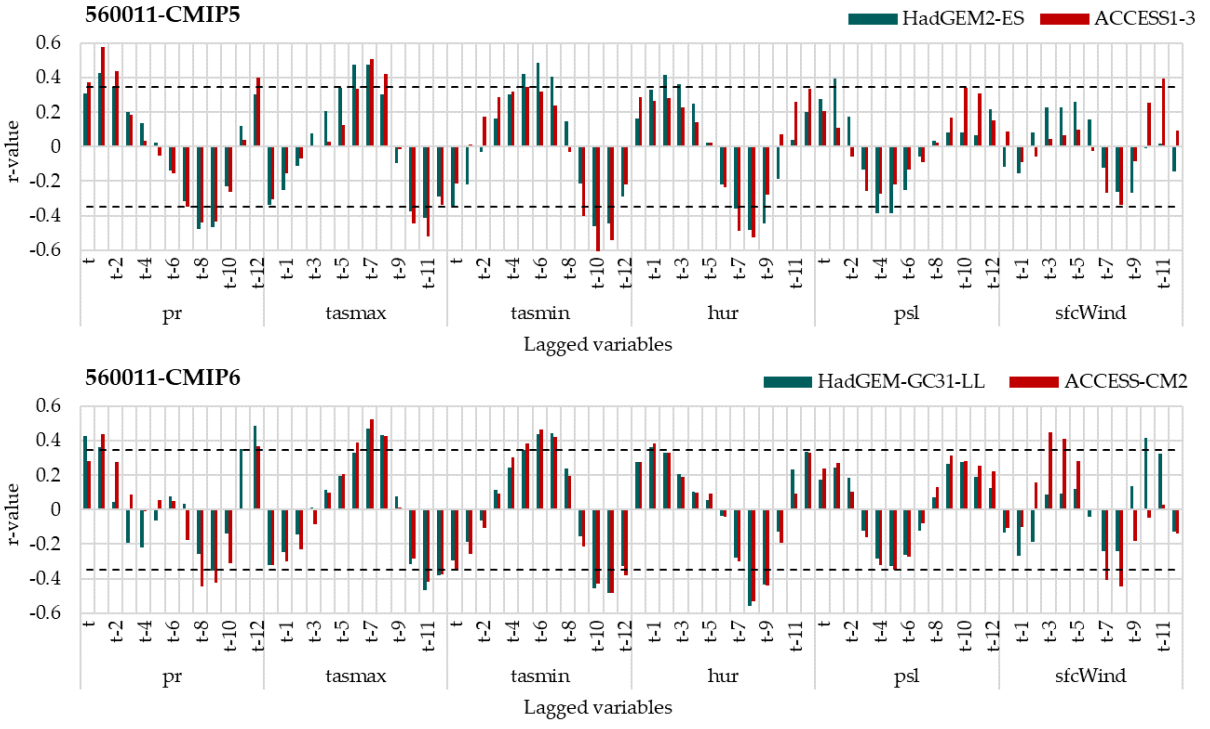

Fig. 10 The correlation (r-value) between the observed rainfall and six lagged variables for #560011.

recommended that the one-year loop relationships be examined to avoid overlooking any lagged climate variables that may have a stronger correlation with the rainfall data. Notably, having a more robustly correlated lagged variable in the ML model training stage leads to superior study performance. This is particularly crucial in traditional ML models, as they require a substantial amount of input data for optimal performance, particularly one with a strong correlation.

In contrast, more complex models such as LSTM automatically select attributes and classify data that learn how to handle them. One key feature of LSTM networks is their ability to selectively retain or discard data over time using gates. Therefore, the gate in an LSTM network automatically learns which input data are relevant to the task at hand and can be safely ignored.[78] Therefore, an explicit input selection procedure is not needed. This is an advantage of deep learning over traditional ML models. Salaeh *et al.*[55] showed that LSTM can exhibit better performance than traditional ML models even without input selection.

4.1.2 Tuning hyperparameters

The WEKA Experimenter, a companion tool in the WEKA software, facilitates the tuning of hyperparameters for the M5, RF, SVR-poly, SVR-rbf, and MLP models, leveraging the root relative squared error value (RRSE) as the performance metric. For the GP and LSTM models, GPdotNET and ANNdotNET software were utilized, respectively, with the lowest RMSE used as the criterion for selecting suitable parameters for each dataset. The summary hyperparameter values and sensitivities, as shown in Table 2, can be explained as follows:

1) M5 model tree (M5)

M5 has been subjected to tuning for two of its hyperparameters, namely, minNumInstances and numDecimalPlaces. By default, minNumInstances is set to 4, serving as the minimum allowable number of instances at a leaf node. According to the findings, the minimum number of instances sensitive across experiments ranged from 4 to 30. Moreover, numDecimalPlaces determines the number of decimal points to be output in the model, and given its insensitivity, it is set to the default value of 4.

2) Random Forest (RF)

During the RF tuning process, three parameters underwent optimization: numIteration, numExecutionSlots, and bagSizePercent. numIteration denotes the number of trees in the forest, and generally, higher values bolster model performance at the expense of increased runtime. The default value for this parameter is 100. In this study, the values ranged from 10 to 150. Meanwhile, numExecutionSlots determines the number of threads or execution slots required for ensemble construction, and since it is nonsensitive, it was set at 1.00 for the experiments. bagSizePercent, on the other hand, relates to the percentage of the total number of samples in the original dataset used to create a bootstrap sample with replacement for each decision tree. It was determined that this parameter exhibits sensitivity; thus, values of 30 and 100 were

established for experimental purposes.

3) Support Vector Regression (SVR)

Two kernel functions for support vector regression (SVR) were evaluated: a polynomial kernel (SVR-poly) and a radial basis function kernel (SVR-rbf). In general, there are two parameters for SVR tuning: the complexity parameter (C) and the epsilon parameter, which determine the margin of tolerance for controlling the trade-off between minimizing the training error and avoiding overfitting. C and epsilonParameter have default values of 1.0 and 1.00E-06, respectively.

C, the epsilon parameter and the exponent were tuned for SVR-poly. Analysis revealed that C and the epsilon parameter were interdependent and highly sensitive, with values ranging from 0.10 to 1.00 and from 1.00E-04 to 1.00E-01, respectively. The exponent, on the other hand, was nonsensitive and retained its default value of 1.00. For SVR-rbf, the tuning of C, epsilonParameter, and gamma, which govern the extent of influence of individual training examples, was performed. All three parameters were sensitive, with C ranging from 0.10 to 10.00, epsilon parameters ranging from 1.00E-06 to 1.00E-01, and gamma ranging from 0.01 to 0.10. The gamma parameter's value had a significant impact on the model's performance, as low values signified a more substantial influence range, while high values indicated closer influence.

4) Multilayer Perceptron (MLP)

The tuning process of the MLP model involved adjusting four parameters. First, the number of hidden layers, which determines the number of hidden layers in the neural network, was set to two in this study. To determine the optimal number of nodes and layers for each test, a trial-and-error approach was used, employing inputs, outputs, inputs+outputs, and (inputs+output)/2. The second parameter, the learning rate, is a critical factor in neural network training, controlling the model's rate of adaptation to the problem.[79] The default learning rate was 0.30, but it was fine-tuned within the range of 0.10 to 0.30. The third parameter, momentum, modifies the weight updates and influences how quickly the neural network weights change during training. The default momentum value is 0.20, and during the tuning process, it was adjusted within the range of 0.10 to 0.60. Finally, the trainingTime parameter specifies the number of epochs used for training, with a default value of 500 and a tuning range of 100 to 1000.

5) Genetic Programming (GP)

In GP, the tuning process involves adjusting seven parameters. The first parameter is the population size in each generation, which determines the initial population and is set to a default value of 500. During tuning, this parameter was varied within the range of 300 to 500. The second, third, and fourth parameters are the crossover probability, mutation probability, and reproduction probability, respectively, and represent the likelihood of these actions occurring in each generation. The tuning process revealed a significant crossover probability ranging from 0.80 to 0.90, indicating a high chance of the next generation being generated by crossover. The mutation probability ranged from 0.05 to 0.10, and the reproduction probability ranged from 0.20 to 0.30. The fifth and sixth parameters are the maximum initial tree level and maximum operation tree level, respectively. The maximum initial tree level is a sensitive parameter and retained its default value of 5, while the maximum operation tree level is a nonsensitive parameter with a default value of 6. The seventh parameter, generation number, determines the number of cycles before termination and has a default value of 500, with a tuning range of 300 to 1200.

6) Long Short-Term Memory (LSTM)

The LSTM network has two key sets of parameters: 1) the visual network designer, which enables the creation of various deep network architectures, and 2) the learning and training parameters. The visual network designer comprises an LSTM layer, dense layers, dropout layers, and an output layer. The LSTM layer is a specialized version of the recurrent network layer that allows for peephole and self-stabilization.[55] In this study, it was found that one layer of LSTM with dimensions between 50 and 110 cells is suitable. The dense layer, a classic neural network layer with an activation function, is defined by two layers with values between 30 and 60. Two layers of dropout with a value of 10 for each are employed to prevent overfitting of the model.[80] The output layer, which has only one dimension, has an ReLU activation function. The suitable activation functions for the LSTM layer, dense layer 1, and dense layer 2 are tanh, tanh, and Relu, respectively.

The learning and training parameters include the learning rate, momentum, and number of epochs. The learning rate determines how quickly the model adapts to the problem and how much the weights of the model are updated based on the error. In this study, it is 0.90. The momentum determines how quickly or slowly the weights of the neural network change during training and has a default value of 1.00, which was retained since it is a nonsensitive parameter. Finally, the number of epochs for training tuning ranges from 100 to 500. The results of hyperparameter tuning in this study indicate that default values of some parameters can produce satisfactory model efficiency without the need for tuning. However, it is not always advisable to rely on default values, as models may not always run satisfactorily without hyperparameter tuning.

Table 2. Summary of the hyperparameter values and sensitivities of the machine learning models.

Models	Hyperparameters	Sensitivity	Start	End
M5	minNumInstances	YES	4	30
	numDecimalPlaces	NO	4	4
RF	bagSizePercent	YES	30	100
	numIteration	YES	10	150
	numExecutionSlots	NO	1	1
SVR-poly	C	YES	0.10	1.00
	epsilonParameter	YES	1.00E-04	1.00E-01
	exponent	NO	1	1
SVR-rbf	C	YES	0.10	10.0
	epsilonParameter	YES	1.00E-06	1.00E-01
	gramma	YES	0.01	0.10
MLP	hiddenLayer1	YES	1	39
	hiddenLayer2	YES	-	39
	learningRate	YES	0.10	0.30
	momentum	YES	0.10	0.60
	trainingTime	YES	100	1000
GP	population size	YES	300	500
	crossover probability	YES	0.80	0.90
	mutation probability	YES	0.05	0.10
	reproduction probability	YES	0.20	0.30
	max initial tree level	YES	5	5
	max operation tree level	NO	6	6
	generation number	YES	300	1200
LSTM	Learning rate	YES	0.90	0.90
	Momentum	NO	1.00	1.00
	Epoch	YES	100	500
	LSTM layer (tanH)	YES	50	110
	Dense layer1 (tanH)	YES	30	60
	Dense layer2 (Relu)	YES	30	60
	Dropout layer1	YES	10	10
	Dropout layer2	YES	10	10
	Output layer (Relu)	YES	1	1

Additionally, it is important to avoid setting parameters to excessively high values, such as training time in the MLP, generation number in the GP, and number of epochs in the LSTM, as this can lead to overfitting during the training stage and result in extreme performance.

In regard to parameter suitability, hyperparameter tuning using the WEKA Experimenter often involves manual tuning or trial-and-error methods.[77,81] This is a common technique that is 100% manually driven; therefore, "suitable" parameter values may not always be "optimal" for a given dataset.[32] Defining parameter ranges and adjusting values accordingly within these ranges can result in gaps that are not thoroughly tested. Nonetheless, this method remains popular among researchers due to its simplicity and reliance on practical experience and knowledge to produce satisfactory results. In line with this study, the trial-and-error method was used to determine suitable parameters for several models.[82,83]

4.1.3 Dataset selection

The dataset selection process for this research was prompted by the desire to examine the data separately for the periods of Jan-Sep and Oct-Dec. Data analysis revealed that October, November, and December were marked by high rainfall, which might be classified as outliers or extreme events.

Consequently, these data points were overlooked during the training stage of the ML models, leading to an underestimation of the model's performance. An underestimation of the training data due to outliers is not uncommon. This critical issue should not be disregarded when dealing with heavy rainfall data. To address this concern, all the data checks conducted for the two periods, Jan-Sep and Oct-Dec, are thoroughly discussed in Section 4.1.4 (Model performance comparison).

Table 3 summarizes the performance metrics of OI, r, and RMSE for the model training, validation, and testing stages, highlighting the top-performing models during both Jan-Sep and Oct-Dec testing, selected from model training using both completed and split datasets. In the Jan-Sep period, the LSTM was the best-performing model with the completed dataset, exhibiting OI, r, and RMSE values of 0.464, 0.462, and 92.30 mm, respectively. In contrast, the SVR-Rbf model trained with the split dataset yielded the best results, with OI, r, and RMSE values in the validation stage of 0.411, 0.314, and 81.89 mm, respectively. In the Oct-Dec period, the GP model was found to be the most effective in modeling rainfall data, with both the completed and split datasets exhibiting similar results. In the validation stage, the GP model achieved OI, r, and RMSE values of 0.768, 0.862, and 97.15 mm, respectively, for the completed dataset and 0.592, 0.675, and 135.59 mm, respectively, for the split dataset. These findings indicate that the GP model can closely replicate the observed rainfall data during the Oct-Dec period.

When utilizing the year-round model, the predicted data from the models used in Jan-Sep (LSTM model) and Oct-Dec (GP model) are combined to provide a comprehensive examination of the data throughout the year. Table 3 presents the efficiency values in the testing stage columns, while Fig. 11 shows the time series graph plotted during the testing stage to compare the predicted and observed rainfall. Overall, the predicted data from training with the completed dataset better simulated rainfall, as evidenced by the OI, r, and RMSE values of 0.739, 0.774, and 115.44 mm, respectively. Furthermore, the green line on the graph that represents the completed dataset line is closer to the observed data during both light and heavy rainfall periods.

Previous research has not provided a clear understanding of how much data are needed for accurate ML simulations. This is due to the different training styles and complexities of the models, which affect the amount of data needed.[84] Deep learning, in general, requires a large amount of data for optimal training.[85] Gauch *et al*.[84] showed that input scaling has a significant impact on LSTM and that it works better when larger datasets are used. Nevertheless, the performance of LSTM relies on several factors, such as the signal in the data, the number of input variables, the correlation and complexity of the input variables, and the model's configuration.[85,86] This is the same for other MLs. Although the SVR-rbf model exhibited the best performance with training using the split dataset from Jan-Sep, it still resulted in unsatisfactory performance.[67,87]

Nonetheless, in regard to the GP results from Oct-Dec, a distinctive trend is observed with both the split and completed datasets. Despite being trained on a smaller dataset than Jan-Sep, outstanding performance is displayed by the GP. This result is surprising and differs from what was expected when training models with limited data. However, it demonstrates the GP's ability to excel in such situations.

Our analysis of training with split datasets indicates that the lengths of these datasets are inadequate for ML. While training the model with cross-validation can prevent

Table 3. The top-model performances for the training, validation, and testing stages in the dataset selection process for sample station #568301.

Case/Stage		Training-Validation			Testing		
		OI	r	RMSE (mm)	OI	r	RMSE (mm)
Completed dataset							
Jan-Sep (LSTM)	Training	0.573	0.697	64.95	0.739	0.774	115.44
	Validation	0.464	0.462	92.30			
Oct-Dec (GP)	Training	0.640	0.686	152.61			
	Validation	0.768	0.862	97.15			
Splitted dataset							
Jan-Sep (SVR-rbf)	Training	0.421	0.296	82.42	0.581	0.576	150.31
	Validation	0.411	0.314	81.89			
Oct-Dec (GP)	Training	0.631	0.660	154.82			
	Validation	0.592	0.675	135.59			

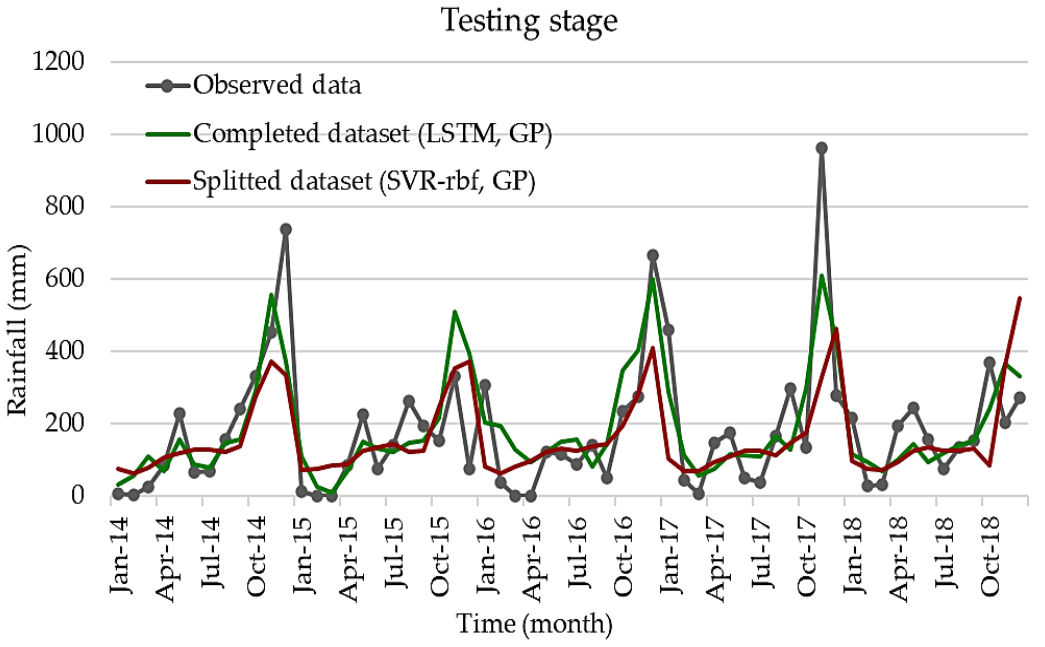

Fig. 11 Time series graph between the observed and predicted data (completed dataset and split dataset) in the testing stage.

overfitting and enhance the model's performance reliability, it can also make the model difficult to train with less data. Therefore, while using split datasets can address outlier data in October, November, and December, the smaller amount of data after splitting presents challenges in training the model and could negatively impact its performance. For these reasons, this study opted to use a complete dataset to train ML models for future monthly rainfall downscaling.

4.1.4 Model performance comparison

This section provides a comparison of model performance in downscaling monthly rainfall data. Tables 4 and 5 show the values of OI, r, and RMSE for the training and testing stages of the three stations (*i.e.*, #560011, #568005, and #568301)

Table 4. Summary of performance comparisons for seven models used to downscale rainfall data at three stations (under CMIP5).

Code station	Models	Training stage			Testing stage		
		OI	r	RMSE (mm)	OI	r	RMSE (mm)
#560011	M5	0.771	0.798	110.05	0.739	0.768	132.42
	RF	0.797	0.845	102.82	0.729	0.769	135.28
	SVR-poly	0.707	0.726	126.06	0.705	0.739	141.56
	SVR-rbf	0.839	0.878	90.34	0.745	0.770	130.77
	MLP	0.752	0.782	114.91	0.742	0.788	131.65
	GP	0.727	0.753	121.24	0.792	0.822	116.65
	LSTM	0.753	0.778	114.74	0.786	0.828	118.62
#568005	M5	0.795	0.833	85.71	0.667	0.691	129.85
	RF	0.928	0.964	46.29	0.688	0.726	125.23
	SVR-poly	0.682	0.707	109.32	0.660	0.718	131.40
	SVR-rbf	0.735	0.776	98.76	0.676	0.720	128.07
	MLP	0.865	0.912	67.31	0.690	0.727	124.78
	GP	0.775	0.806	96.42	0.752	0.802	125.13
	LSTM	0.786	0.826	87.62	0.757	0.814	109.00
#568301	M5	0.768	0.803	100.03	0.682	0.704	125.76
	RF	0.932	0.961	48.62	0.679	0.700	126.41
	SVR-poly	0.680	0.721	119.81	0.666	0.699	129.23
	SVR-rbf	0.688	0.732	118.13	0.666	0.703	129.06
	MLP	0.739	0.769	106.90	0.671	0.715	128.03
	GP	0.723	0.751	114.34	0.686	0.715	127.73
	LSTM	0.769	0.802	99.92	0.742	0.775	111.78

Table 5. Summary of performance comparisons for seven models used to downscale rainfall data at three stations (under CMIP6).

Code station	*Models*	Training stage			Testing stage		
		OI	r	RMSE (mm)	OI	r	RMSE (mm)
#560011	*M5*	0.749	0.766	107.42	0.665	0.707	149.26
	RF	0.819	0.850	89.53	0.637	0.680	156.07
	SVR-poly	0.708	0.722	116.54	0.660	0.716	150.57
	SVR-rbf	0.726	0.743	112.69	0.671	0.719	147.80
	MLP	0.732	0.778	111.24	0.644	0.743	154.50
	GP	0.697	0.733	118.99	0.677	0.729	146.05
	LSTM	0.725	0.737	114.13	0.751	0.841	132.52
#568005	*M5*	0.674	0.675	97.98	0.614	0.663	153.56
	RF	0.826	0.879	69.10	0.619	0.694	152.50
	SVR-poly	0.680	0.686	96.91	0.613	0.668	153.64
	SVR-rbf	0.791	0.835	76.71	0.654	0.759	144.25
	MLP	0.850	0.885	64.04	0.627	0.659	161.84
	GP	0.741	0.761	98.51	0.730	0.778	140.05
	LSTM	0.709	0.733	95.51	0.758	0.849	128.91
#568301	*M5*	0.735	0.761	98.69	0.538	0.612	173.45
	RF	0.655	0.681	114.21	0.508	0.611	179.59
	SVR-poly	0.681	0.715	109.35	0.590	0.649	162.20
	SVR-rbf	0.665	0.700	112.43	0.591	0.686	162.11
	MLP	0.678	0.710	110.01	0.627	0.663	154.00
	GP	0.754	0.786	105.09	0.696	0.769	155.24
	LSTM	0.770	0.836	93.50	0.681	0.797	153.34

included in this study under CMIP5 and CMIP6. It is important to note that this study focuses mainly on the testing stage, and the primary interest lies in the OI and r criteria since the RMSE is already part of the OI. Under CMIP5, #560011 exhibited the highest efficiency for the OI criteria using the GP, while the LSTM demonstrated the highest efficiency for r, with OI and r values of 0.792 and 0.822 for the GP and 0.786 and 0.828 for the LSTM, respectively. For #568005 and #568301, the LSTM was the most efficient, showing OI and r values of 0.757 and 0.814 and 0.742 and 0.775, respectively. Under CMIP6, the LSTM model was the best-performing model, with OI and r values of 0.751 and 0.841, respectively, for #560011 and 0.758 and 0.849, respectively, for #568005. However, for #568301, the results were different. When considering OI, the GP was the most efficient, with an OI and an r value of 0.696 and 0.769, respectively. Conversely, when considering r, the LSTM was the most efficient, with OI and r values of 0.681 and 0.797, respectively.

Moreover, observations indicate that while the RF model was highly effective in most cases during the training stage, this was not always the case during the testing stage. Despite being optimized for a key parameter, namely, bagSizePercent, to match the datasets, the RF model was found to be prone to overfitting. This parameter determines the size of each bag and thereby the number of samples used for training. It is typically set at 100, meaning that lower percentages indicate fewer training samples.[88] This randomizes the modeling process and helps to curb overfitting. Nevertheless, despite RF overfitting, its performance during the testing stage was found to be only slightly inferior to that of many other models.

Figure 12 displays a scatter plot that illustrates the relationship between the observed and predicted rainfall across seven ML models tested at three stations (#560011, #568005, and #568301) under CMIP5 and CMIP6. The 45-degree diagonal line represents the ideal data relationship. Analysis of the graph revealed that the points tended to cluster closer to the diagonal line within the lower range of values. Although the density distributions of most models remained similar across certain data ranges, several models demonstrated significant deviations. For example, MLP and GP yielded superior performances despite the model proximity in the top two rainfall spots in #560011_CMIP6 and #568005_CMIP6. The findings indicate that the MLP and GP models consistently outperformed the other models in scenarios with high rainfall, and they emerged as the preeminent models in both the CMIP5 and CMIP6 datasets. Conversely, the RF, SVR-poly, and SVR-rbf models consistently underestimated rainfall, as indicated by their lower points. In addition, minor differences in the predicted data between SVR-poly and SVR-rbf were detected in some

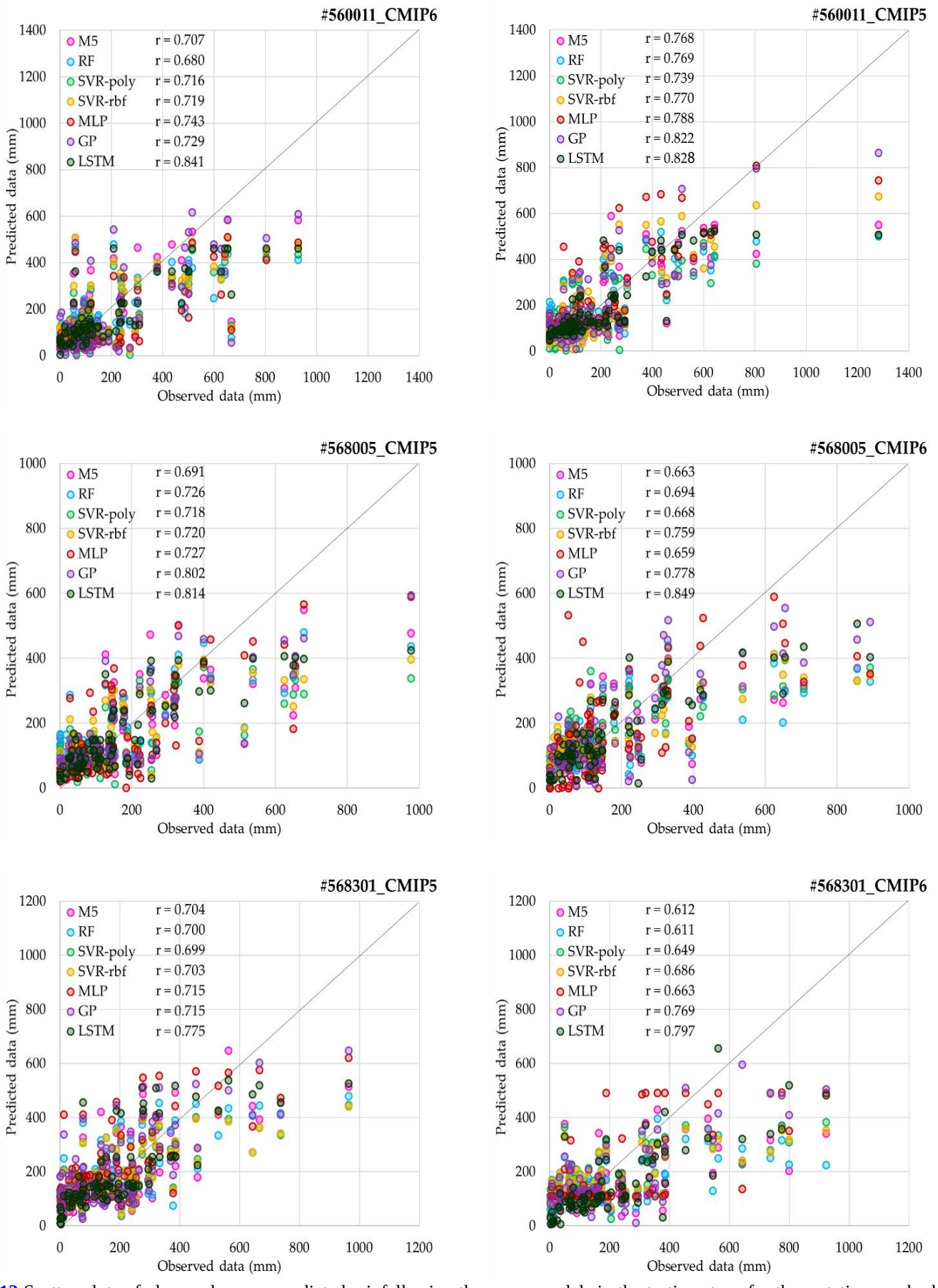

Fig. 12 Scatter plots of observed versus predicted rainfall using the seven models in the testing stage for three stations under both CMIP5 (left side) and CMIP6 (right side).

cases (such as for #568301_CMIP5 and #568301_CMIP6). Although these kernel functions are fundamentally different, they exhibited a striking resemblance, suggesting the possibility that they can generate similar predictions for some datasets, possibly involving some structured input properties that cause the two functions to perform similarly.

Considering the performance of downscaling monthly rainfall using the OI, r, and RMSE criteria presented in Tables 4 and 5 and Fig. 12, unequivocal findings indicate that LSTM is the most efficient model for this task, which is supported by previous studies such as Misra *et al*.[89] and Jose *et al*.[22]. Despite their effectiveness in many domains, machine learning (ML) models, including LSTM, often struggle to accurately estimate values during periods of high-range rainfall due to their limited ability to comprehend time series and rainfall patterns. They tend to focus primarily on the majority of the data, resulting in unreliable and unrealistic predictions that have limited practicality and are unsuitable for real-world applications. For instance, Cramer *et al*.[90] demonstrated that ML models such as GP, SVR-rbf, and M5 were ill suited for daily precipitation predictions since they failed to consider precipitation patterns on wet and dry days, leading to predictions of rainfall only in the average range. Therefore, it is imperative to consider models that can perform well even under the most challenging conditions of high-range rainfall to enhance the accuracy and practicality of monthly rainfall predictions.

For all the above reasons, this study focuses on the estimation of peak data. While the model performed well in terms of the OI, r, and RMSE, it is crucial to consider its ability to accurately estimate peak data. Fig. 13 displays the relative peak error (PE) percentage of seven ML models for three stations under CMIP5 and CMIP6 during the testing phase, where a lower PE indicates higher model efficiency, with values less than 50% considered acceptable.[75] While findings demonstrate that LSTM is the most efficient model, this is not the case when considering PE. The results indicate that the MLP and GP are better suited for simulating high-range rainfall, with the GP showing greater overall dominance than the MLP, except for #568005 under CMIP5, where the MLP is slightly superior. Similarly, Hadi Pour *et al*.[91] showed that GP predicts annual and seasonal extreme rainfall better than does ANN and linear regression-based SDSM. Analysis also indicated that the other models exhibited below-acceptable performance.

GP is a powerful method that excels in predicting high-range rainfall; however, its performance during normal rainfall periods is suboptimal due to its tendency to overestimate values, particularly extreme ones. This tendency can lead to deviations from true values during normal rainfall periods, reducing its reliability. To ensure more reliable predictions of future rainfall, a novel approach is proposed that utilizes appropriate models for different rainfall periods. Based on the average monthly rainfall depicted in Fig. 2, October, November, and December were identified as the months with the highest rainfall in the TSS. Therefore, in addition to analyzing the year-round rainfall trend, rainfall is examined separately in two periods: January to September (Jan-Sep) and October to December (Oct-Dec). This approach aims to achieve more accurate and reliable rainfall predictions for both normal and high-range rainfall periods.

To evaluate the performances of the monthly rainfall downscaling models across the year-round, Jan-Sep, and Oct-Dec periods, the OI and R criteria were employed for model

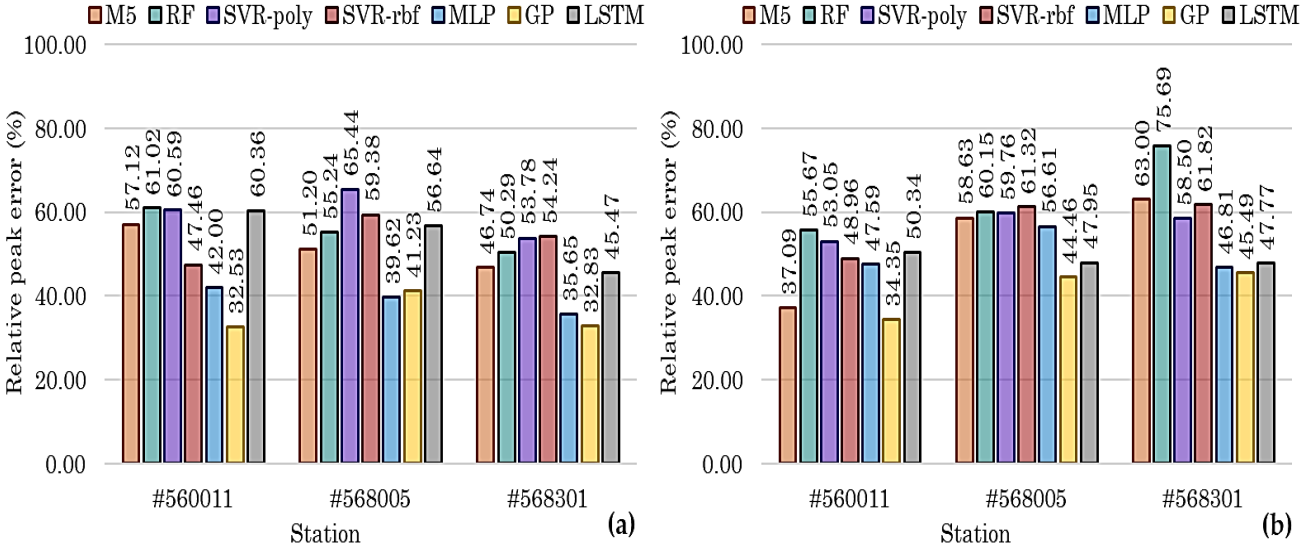

Fig. 13 The seven ML models' relative peak errors (PEs) for downscaling monthly rainfall data in the testing stage for three stations under a) CMIP5 and b) CMIP6.

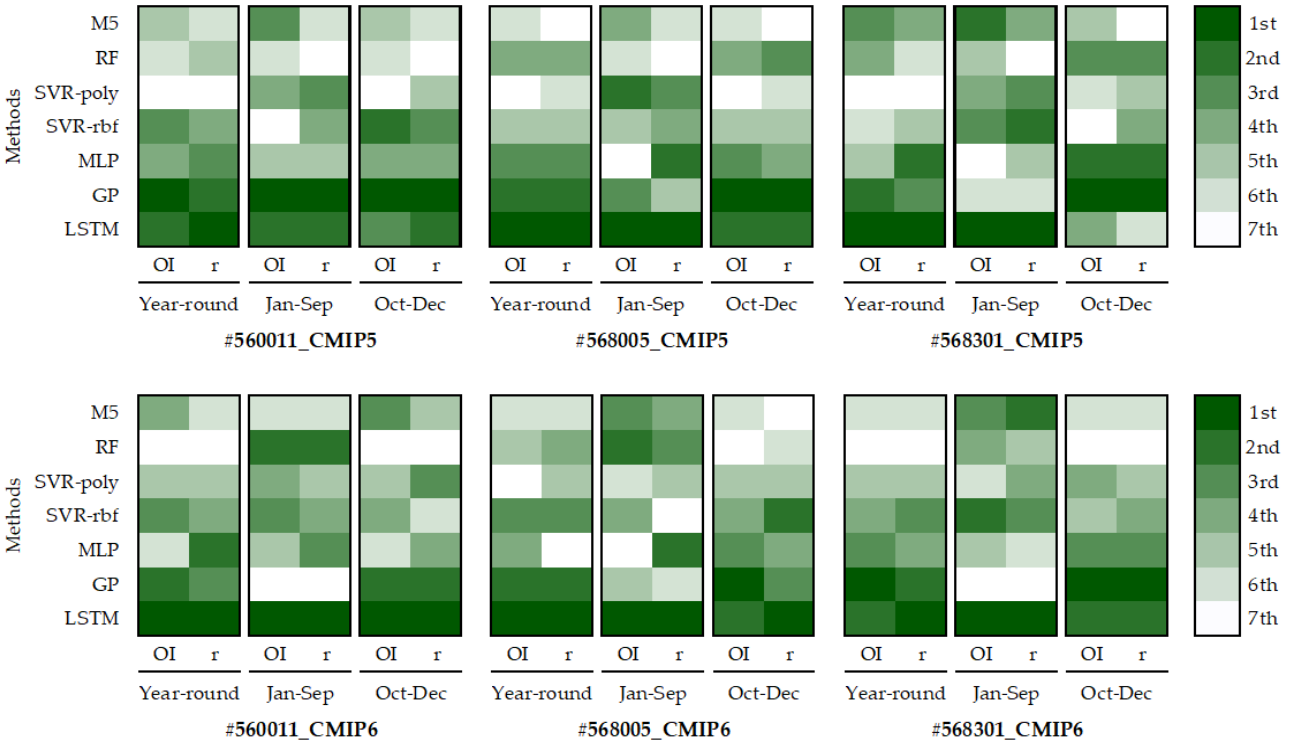

Fig. 14 Ranking of the seven models for three stations in the testing stage using the OI and R criteria.

ranking. Fig. 14 presents the rankings of seven models (y-axis) tested at three stations and under CMIP5 and CMIP6 (x-axis). The legend on the right-hand side of Fig. 14 denotes the 1st rank (the best model) as dark green, the 7th rank (the worst model) as white, and the intermediate ranks with a sequential color scheme. The primary focus was on the OI criterion, which is a stringent and sensitive metric for evaluating changes in predicted data. Analysis revealed that the LSTM model emerged as the top-performing model during the year-round period, ranking first in four out of six cases, followed by the GP model, which ranked first in the remaining two cases. In the Jan-Sep period, LSTM ranked first in five cases, except for #560011_CMIP5, where GP took the lead. During Oct-Dec, GP ranked fifth in six cases, except for #560011_CMIP6, where LSTM outperformed the other models.

According to these rankings, on average, LSTM outperforms the other models, and GP can be more effective in Oct-Dec. Therefore, based on the results of comparing the performances of the seven models on monthly rainfall downscaling, LSTM was selected for future rainfall downscaling in Jan-Sep, GP in Oct-Dec, and a combination of both models year-round.

4.2 LSTM-GP performance

In this section, the performance of a combined LSTM and GP model is examined. This model demonstrates the highest efficiency for monthly rainfall reduction during two distinct periods: Jan-Sep and Oct-Dec. The performance of the LSTM-GP testing stage is summarized in Table 6, where its performance against the OI, r, and RMSE criteria for three stations under CMIP5 and CMIP6 is assessed. To ensure that the analysis was representative, three sample stations for the TSS were selected, and the focus was on the average of their performance metrics. Notably, the average efficacies of OI, r, and RMSE for CMIP5 were 0.794, 0.834, and 104.08 mm, respectively. For CMIP6, these metrics were 0.763, 0.834, and 119.20 mm, respectively. Based on the analysis, an OI greater than 0.60 and an r greater than 0.80, indicating a very strong relationship,[92] were found. Thus, it is concluded that the LSTM-GP model is well suited for downscaling rainfall data in this area.

Figure 15 presents the Taylor diagram, which serves to evaluate the performance of seven models, including LSTM-GP, during the testing stage at three stations under both CMIP5 and CMIP6. The r value and standard deviation (SD) were used to evaluate the predictive performance of the models. Notably, both the LSTM and LSTM-GP models exhibit superior predictive capability compared to other models (*i.e.*, M5, RF, SVR-poly, SVR-rbf, MLP, and GP). Specifically, the LSTM-GP model outperforms the LSTM model, with the SD values indicating a closer distribution of the data to the observed rainfall.

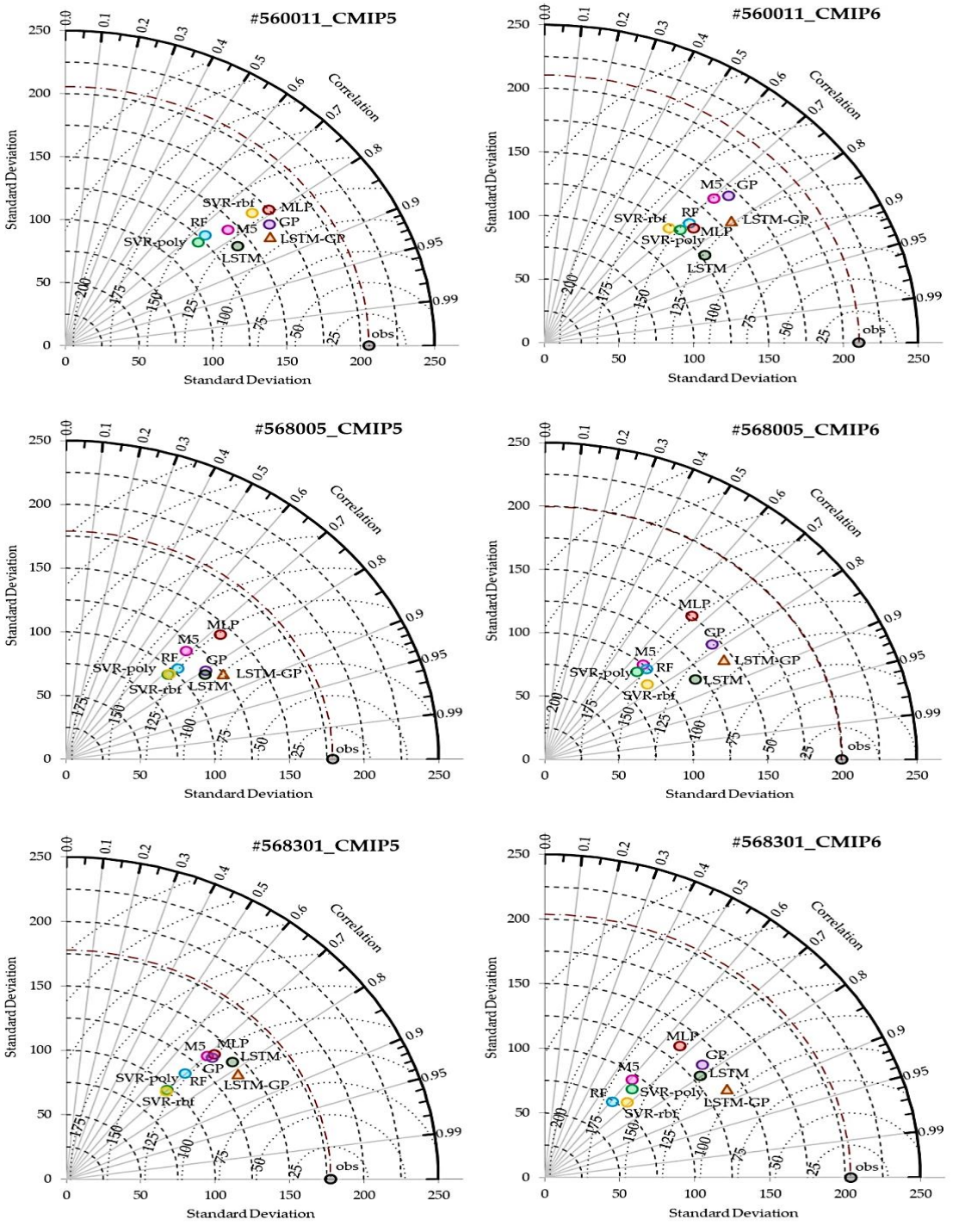

Fig. 15 Taylor diagram comparing six models and LSTM-GP for three stations during the testing stage for both CMIP5 (left side) and CMIP6 (right side).

Table 6. Summary of the performance of the LSTM-GP model in downscaling rainfall data for three stations in the testing stage under CMIP5 and CMIP6.

Code	CMIP5			CMIP6		
station	OI	r	RMSE (mm)	OI	r	RMSE (mm)
#560011	0.817	0.849	108.58	0.745	0.794	127.67
#568005	0.791	0.843	100.11	0.781	0.837	110.93
#568301	0.775	0.811	103.55	0.762	0.871	118.99
Average	*0.794*	*0.834*	*104.08*	*0.763*	*0.834*	*119.20*

4.3 Examining trends in future rainfall

Figure 16 depicts the relative percentage change in projected monthly rainfall for the 2030s, 2060s, and 2080s at three TSS sample stations, displaying variations in precipitation under different scenarios, including CMIP5 (RCP4.5 and RCP8.5) and CMIP6 (SSP245 and SSP585). Fig. 17 provides an overview of the average future rainfall in the same periods as that examined for the three sample stations of the TSS, namely, year-round, Jan-Sep, and Oct-Dec. Analysis reveals that year-round rainfall in the 2030s is likely to remain constant or exhibit a slight decrease under all four emission scenarios. The most significant downward trend is observed under RCP4.5 and SSP245, which show decreases of -7.11% and -3.66%, respectively, while RCP8.5 and SSP585 show little change in this decade. However, in the 2060s and 2080s, the monthly precipitation is expected to increase significantly throughout the century, with the most severe increase projected under RCP8.5. Specifically, the increase in rainfall is estimated to be 49.16% and 78.59% in the 2060s and 2080s, respectively.

With regard to the period from Jan-Sep, a comparable trend of slight change in the 2030s is noted, with an increase of only 0.56% to 3.13% projected under all scenarios. However, in the 2060s, there is a discrepancy in the predictions between CMIP5 and CMIP6, with RCP4.5 and RCP8.5 showing increases of 4.78% to 7.25%, whereas SSP585 predicts a decrease of -4.05%. Notably, in the 2080s, SSP585 showed a decreasing trend of -10.47%, unlike the other scenarios, which indicated a trend of increasing rainfall of 7.25% to 17.06%. During the heavy rainfall period (Oct-Dec), all four scenarios show a consistent trend except in the 2030s, where RCP4.5 predicts a downward trend of -6.39%. In contrast, SSP245, RCP8.5, and SSP585 predicted increases of 3.29%, 10.89%, and 26.16%, respectively. The analysis suggested that precipitation had a particularly strong upward trend in the late century, increasing as much as 93.86% and 60.05% in the 2060s and 96.49% and 138.44% in the 2080s under RCP8.5 and SSP585, respectively.

The findings of this study suggest that climate change is likely to have only a minor impact on rainfall patterns in the TSS in the near future, especially in the 2030s. However, significant changes are expected in the later stages of this century (2060s and 2080s), with Oct-Dec experiencing particularly drastic shifts. These observations align with those of Supharatid *et al.*,[93] who reported that the probability of increased rainfall increases during the southwest (S/W) and northeast (N/E) monsoons in Thailand. Under SSP245,

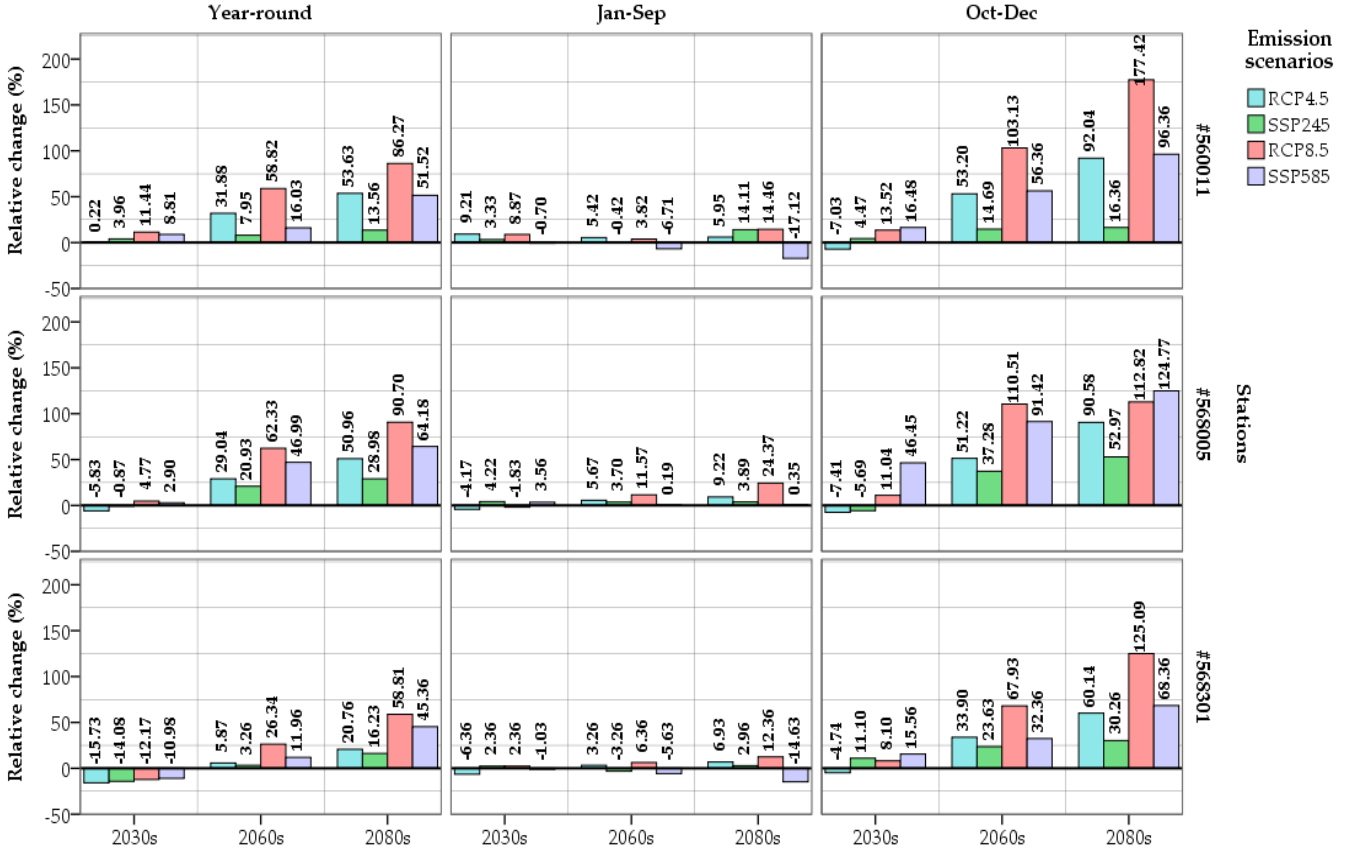

Fig. 16 Relative changes in rainfall at three stations in the year-round, Jan-Sep, and Oct-Dec periods under four emission scenarios (RCP4.5, SSP245, RCP8.5, and SSP585).

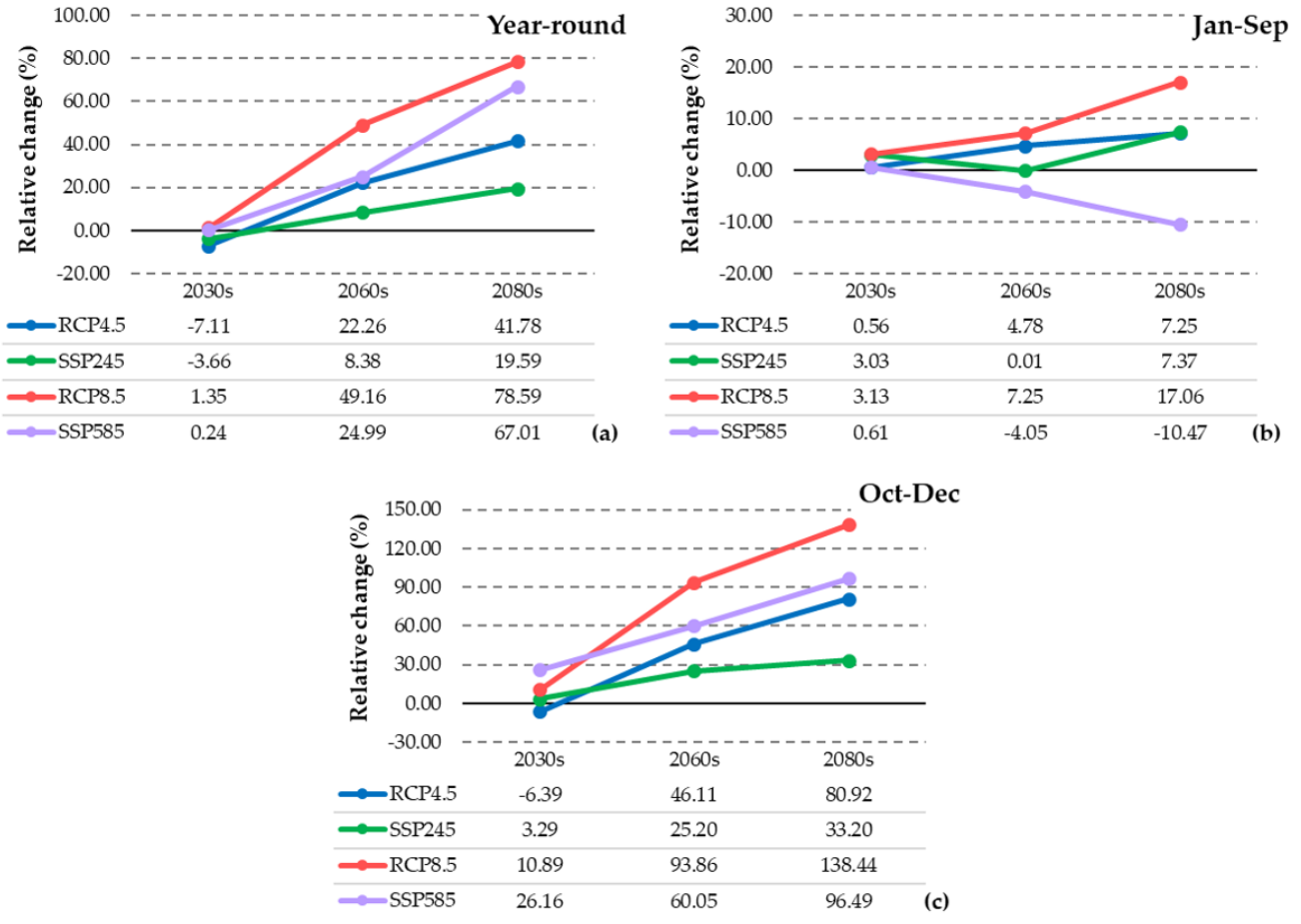

Fig. 17 The average relative change in TSS under four emission scenarios (RCP4.5, SSP245, RCP8.5, and SSP585) (a) year-round, (b) Jan-Sep, and (c) Oct-Dec.

increases of more than 11.17% and 13.47% are projected for 2050 and 2100, respectively, whereas SSP585 forecasts increases of more than 12.90% and 22.12%, respectively, in the same periods. It is worth noting that while this study differs from a national-level survey of rainfall trends, given that it is focused on the TSS watershed in southern Thailand, the area under investigation is a major watershed and heavily influenced by the N/E monsoon during the Oct-Dec period. Our results demonstrate that TSS tends to have a more significant impact during this season than in the country due to the region's location in relation to atmospheric circulation patterns, which are affected by the land–sea configuration and create imbalances in precipitation distribution.[94,95]

When comparing trends in precipitation changes predicted by CMIP5 and CMIP6, it was observed that CMIP5 exhibited more drastic changes in both the medium- (RCP4.5) and high-emission scenarios (RCP8.5). Previous studies have compared the trends in precipitation changes predicted by CMIP5 and CMIP6, including Zhang and Chen,[96] who found that CMIP5 tends to be more intense than CMIP6 when accounting for all uncertainties in the study, such as GHG emission scenarios, GCMs, intermodel variance, and variance in noise across all models, emission scenarios, and prediction periods. However, the results may show the opposite when some uncertainty is not considered. In contrast to our own reports, including those of Sobie *et al.*[97], Hamed *et al.*[98] and Du *et al.*,[99] other findings show that CMIP6 predictions are more drastic under high-emission scenarios at the end of the century. The trend in CMIP6 is clearly more extreme than that in CMIP5. However, different results were observed for each study, and these results were influenced by many factors, such as the type and number of GCMs, the type and number of climate variables, the behavior and patterns of rainfall data, and the downscaling methods and models.[94,96,100] These factors unavoidably affect the results of studies and represent inherent conditions and limitations in many studies. As a result, while the uncertainty of GCMs in CMIP6 has been reduced compared to that in previous versions, drawing firm conclusions about the results of these models under different conditions remains challenging.

A severe change, whether it is an increase or decrease, is not a good thing. The impact will be felt most acutely in major

economic activities such as agriculture, which accounts for the majority of the TSS. For monthly rainfall changes in the near, middle, and far future, this study clearly shows the need to prepare for climate change impacts, at least in the near future. Although there is still uncertainty in distant future predictions due to the inherent insecurity of future human activities that will affect GHG emissions, this means that the effects could be less severe or more severe than these findings.[101-103]

6. Conclusions

Research on comparing downscaling methods for monthly rainfall data in three sample stations within the TSS basin utilizing eight machine learning models. The outcomes of this study reveal that the combination of LSTM and GP models produces the most effective downscaling of monthly rainfall data. The OI and r-value demonstrate a highly robust relationship for the average values within the TSS watershed. These discoveries offer valuable insights into the unique strengths and weaknesses of each model type, which are influenced by factors such as the size and type of data used during model training. In regions characterized by intricate rainfall behavior and patterns, such as the TSS, conventional models may face challenges in accurately capturing all the intricacies. Nevertheless, the essential steps of data preparation, input selection, and hyperparameter tuning in machine learning significantly influence model performance. Therefore, these steps should not be underestimated, as they play a pivotal role in attaining meaningful and dependable results.

Examination of future rainfall change trends in the TSS Basin: This study revealed that in the near future, there will be a slight overall trend of rainfall variation in the TSS Basin under both moderate and extreme emission scenarios. However, as the years approach the middle to the end of the century, the rainfall trend in this region shows a significant increase, particularly during the monsoon season.

Acknowledgments

The authors would like to express gratitude to the College of Graduate Studies, Walailak University, for providing Walailak University Master's Degree Excellence Scholarships under Contract No. ME04/2021.

Conflict of interest

There are no conflicts of interest.

Supporting Information

Not applicable.

References

[1] R. S. Laddimath, N. S. Patil, Artificial neural network technique for statistical downscaling of global climate model, *MAPAN*, 2019, **34**, 121-127, doi: 10.1007/s12647-018-00299-0.

[2] M. T. Vu, T. Aribarg, S. Supratid, S. V. Raghavan, S.-Y. Liong, Statistical downscaling rainfall using artificial neural network: significantly wetter Bangkok? *Theoretical and Applied Climatology*, 2016, **126**, 453-467, doi: 10.1007/s00704-015-1580-1.

[3] IPCC, Climate Change and Land: an IPCC special report on climate change, desertification, land degradation, sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems, 2019.

[4] D. Eckstein, V. Künzel, L. Schäfer, *The Global climate risk index 2021*. Bonn: Germanwatch, 2021.

[5] X. Li, L. Tan, Y. Li, J. Qi, P. Feng, B. Li, De Li Liu, X. Zhang, G. W. Marek, Y. Zhang, H. Liu, R. Srinivasan, Y. Chen, Effects of global climate change on the hydrological cycle and crop growth under heavily irrigated management–A comparison between CMIP5 and CMIP6, *Computers and Electronics in Agriculture*, 2022, **202**, 107408, doi: 10.1016/j.compag.2022.107408.

[6] D. B. Lobell, W. Schlenker, J. Costa-Roberts, Climate trends and global crop production since 1980, *Science*, 2011, **333**, 616-620, doi: 10.1126/science.1204531.

[7] S. McGuire, FAO, IFAD, and WFP. the state of food insecurity in the world 2015: meeting the 2015 international hunger targets: taking stock of uneven progress. Rome: FAO, 2015, *Advances in Nutrition*, 2015, **6**, 623-624, doi: 10.3945/an.115.009936.

[8] Agricultural Economics Thailand. Agricultural utilization area by province. Retrieved from https://www.oae.go.th/ (accessed August, 2022).

[9] K. E. Taylor, R. J. Stouffer, G. A. Meehl, An overview of CMIP5 and the experiment design, *Bulletin of the American Meteorological Society*, 2012, **93**, 485-498, doi: 10.1175/bams-d-11-00094.1.

[10] V. Singh, M. K. Goyal, Analysis and trends of precipitation lapse rate and extreme indices over North Sikkim eastern Himalayas under CMIP5ESM-2M RCPs experiments, *Atmospheric Research*, 2016, **167**, 34-60, doi: 10.1016/j.atmosres.2015.07.005.

[11] Z. Sa'adi, S. Shahid, E.-S. Chung, T. bin Ismail, Projection of spatial and temporal changes of rainfall in Sarawak of Borneo Island using statistical downscaling of CMIP5 models, *Atmospheric Research*, 2017, **197**, 446-460, doi: 10.1016/j.atmosres.2017.08.002.

[12] A. Dosio, M. W. Jury, M. Almazroui, M. Ashfaq, I.

Diallo, F. A. Engelbrecht, N. A. B. Klutse, C. Lennard, I. Pinto, M. B. Sylla, A. T. Tamoffo, Projected future daily characteristics of African precipitation based on global (CMIP5, CMIP6) and regional (CORDEX, CORDEX-CORE) climate models, *Climate Dynamics*, 2021, **57**, 3135-3158, doi: 10.1007/s00382-021-05859-w.

[13] H. Rhymee, S. Shams, U. Ratnayake, E. K. A. Rahman, Comparing statistical downscaling and arithmetic mean in simulating CMIP6 multi-model ensemble over Brunei, *Hydrology*, 2022, **9**, 161, doi: 10.3390/hydrology9090161.

[14] M. F. Rabezanahary Tanteliniaina, M. H. Rahaman, J. Zhai, Assessment of the future impact of climate change on the hydrology of the mangoky river, Madagascar using ANN and SWAT, *Water*, 2021, **13**, 1239, doi: 10.3390/w13091239.

[15] J. M. Eden, M. Widmann, D. Maraun, M. Vrac, Comparison of GCM- and RCM-simulated precipitation following stochastic postprocessing, *Journal of Geophysical Research*: *Atmospheres*, 2014, **119**, 11040-11053, doi: 10.1002/2014jd021732.

[16] S. Cheevaprasert, R. Mehrotra, S. Thianpopirug, N. Sriwongsitanon, An evaluation of statistical downscaling techniques for simulating daily rainfall occurrences in the upper Ping River Basin, *Hydrology*, 2020, **7**, 63, doi: 10.3390/hydrology7030063.

[17] R. L. Wilby, T. M. L. Wigley, Downscaling general circulation model output: a review of methods and limitations, *Progress in Physical Geography*: *Earth and Environment*, 1997, **21**, 530-548, doi: 10.1177/030913339702100403.

[18] D. Maraun, F. Wetterhall, A. M. Ireson, R. E. Chandler, E. J. Kendon, M. Widmann, S. Brienen, H. W. Rust, T. Sauter, M. Themeßl, V. K. C. Venema, K. P. Chun, C. M. Goodess, R. G. Jones, C. Onof, M. Vrac, I. Thiele-Eich, Precipitation downscaling under climate change: recent developments to bridge the gap between dynamical models and the end user, *Reviews of Geophysics*, 2010, **48**, RG3003, doi: 10.1029/2009rg000314.

[19] D. A. Sachindra, K. Ahmed, M. M. Rashid, S. Shahid, B. J. C. Perera, Statistical downscaling of precipitation using machine learning techniques, *Atmospheric Research*, 2018, **212**, 240-258, doi: 10.1016/j.atmosres.2018.05.022.

[20] Q. Tran Anh, K. Taniguchi, Coupling dynamical and statistical downscaling for high-resolution rainfall forecasting: case study of the Red River Delta, Vietnam, *Progress in Earth and Planetary Science*, 2018, **5**, 28, doi: 10.1186/s40645-018-0185-6.

[21] J.-C. Han, W. Zheng, Z. Liu, Y. Zhou, Y. Huang, B. Li, Downscaling of precipitation for climate change projections using multiple machine learning techniques: case study of Shenzhen city, China, *Journal of Water Resources Planning and Management*, 2022, **148**, 05022008, doi: 10.1061/(asce)wr.1943-5452.0001612.

[22] D. M. Jose, A. M. Vincent, G. S. Dwarakish, Improving multiple model ensemble predictions of daily precipitation and temperature through machine learning techniques, *Scientific Reports*, 2022, **12**, 4678, doi: 10.1038/s41598-022-08786-w.

[23] D. Tran Anh, S. P. Van, T. D. Dang, L. P. Hoang, Downscaling rainfall using deep learning long short-term memory and feedforward neural network, *International Journal of Climatology*, 2019, **39**, 4170-4188, doi: 10.1002/joc.6066.

[24] A. Dey, D. P. Sahoo, R. Kumar, R. Remesan, A multimodel ensemble machine learning approach for CMIP6 climate model projections in an Indian River Basin, *International Journal of Climatology*, 2022, **42**, 9215-9236, doi: 10.1002/joc.7813.

[25] R. Xu, N. Chen, Y. Chen, Z. Chen, Downscaling and projection of multi-CMIP5 precipitation using machine learning methods in the upper Han River Basin, *Advances in Meteorology*, 2020, **2020**, 8680436, doi: 10.1155/2020/8680436.

[26] J. Das, U. V. Nanduri, Assessment and evaluation of potential climate change impact on monsoon flows using machine learning technique over Wainganga River Basin, India, *Hydrological Sciences Journal*, 2018, **63**, 1020-1046, doi: 10.1080/02626667.2018.1469757.

[27] M. Niazkar, M. R. Goodarzi, A. Fatehifar, M. J. Abedi, Machine learning-based downscaling: application of multi-gene genetic programming for downscaling daily temperature at Dogonbadan, Iran, under CMIP6 scenarios, *Theoretical and Applied Climatology*, 2023, **151**, 153-168, doi: 10.1007/s00704-022-04274-3.

[28] M. Norel, K. Krawiec, Z. W. Kundzewicz, Machine learning modeling of climate variability impact on river runoff, *Water*, 2021, **13**, 1177, doi: 10.3390/w13091177.

[29] W. Köppen, The thermal zones of the Earth according to the duration of hot, moderate and cold periods and to the impact of heat on the organic world, *Meteorologische Zeitschrift*, 2011, **20**, 351-360, doi: 10.1127/0941-2948/2011/105.

[30] S. Ali, K. Techato, J. Taweenkun, S. Gyawali, Assessment of land use suitability for natural rubber using GIS in the U-tapao River Basin, Thailand, *Kasetsart Journal of Social Sciences*, 2018, **41**, 110-117, doi: 10.1016/j.kjss.2018.07.002.

[31] A. S. Tabucanon, K. Kurisu, K. Hanaki, Assessment and mitigation of tangible flood damages driven by climate change in a tropical city: hat Yai Municipality, southern Thailand, *Science of the Total Environment*, 2021, **789**, 147983, doi: 10.1016/j.scitotenv.2021.147983.

[32] S. Pinthong, P. Ditthakit, N. Salaeh, M. A. Hasan, C. T. Son, N. T. T. Linh, S. Islam, K. K. Yadav, Imputation of missing monthly rainfall data using machine learning and spatial interpolation approaches in Thale Sap Songkhla River Basin, Thailand, *Environmental Science and Pollution Research*, 2022,

1-17, doi: 10.1007/s11356-022-23022-8.
[33] S. Ghosh, SVM-PGSL coupled approach for statistical downscaling to predict rainfall from GCM output, *Journal of Geophysical Research*: *Atmospheres*, 2010, **115**, D22102, doi: 10.1029/2009jd013548.
[34] L. Campozano, D. Tenelanda, E. Sanchez, E. Samaniego, J. Feyen, Comparison of statistical downscaling methods for monthly total precipitation: case study for the paute river basin in southern Ecuador, *Advances in Meteorology*, 2016, **2016**, 6526341, doi: 10.1155/2016/6526341.
[35] U. Okkan, O. Fistikoglu, Evaluating climate change effects on runoff by statistical downscaling and hydrological model GR2M, *Theoretical and Applied Climatology*, 2014, **117**, 343-361, doi: 10.1007/s00704-013-1005-y.
[36] B. Wang, L. Zheng, D. L. Liu, F. Ji, A. Clark, Q. Yu, Using multi-model ensembles of CMIP5 global climate models to reproduce observed monthly rainfall and temperature with machine learning methods in Australia, *International Journal of Climatology*, 2018, **38**, 4891-4902, doi: 10.1002/joc.5705.
[37] N. A. F. Sulaiman, S. M. Shaharudin, S. Ismail, N. H. Zainuddin, M. L. Tan, Y. Abd Jalil, Predictive modelling of statistical downscaling based on hybrid machine learning model for daily rainfall in east-coast peninsular Malaysia, *Symmetry*, 2022, **14**, 927, doi: 10.3390/sym14050927.
[38] C. Yang, N. Wang, and S. Wang, A comparison of three predictor selection methods for statistical downscaling, *International Journal of Climatology*, 2017, **37**, 1238-1249, doi: https://doi.org/10.1002/joc.4772.
[39] C.-Y. Lin, C.-P. Tung, Procedure for selecting GCM datasets for climate risk assessment, *Terrestrial, Atmospheric and Oceanic Sciences*, 2017, **28**, 43-55, doi: 10.3319/tao.2016.06.14.01(cca).
[40] N. Zeydalinejad, H. R. Nassery, A. R. Shakiba, and F. Alijani, The Evaluations of NEX-GDDP and Marksim Downscaled Data Sets Over Lali Region, Southwest Iran, *Journal of the Earth and Space Physics*, 2021, **46**, 213-230.
[41] M. S. Abbasian, M. R. Najafi, A. Abrishamchi, Increasing risk of meteorological drought in the Lake Urmia Basin under climate change: introducing the precipitation–temperature deciles index, *Journal of Hydrology*, 2021, **592**, 125586, doi: 10.1016/j.jhydrol.2020.125586.
[42] N. Tessema, A. Kebede, D. Yadeta, Modelling the effects of climate change on streamflow using climate and hydrological models: the case of the Kesem sub-basin of the Awash River Basin, Ethiopia, *International Journal of River Basin Management*, 2021, **19**, 469-480, doi: 10.1080/15715124.2020.1755301.
[43] R. Duan, G. Huang, Y. Li, X. Zhou, J. Ren, C. Tian, Stepwise clustering future meteorological drought projection and multi-level factorial analysis under climate change: a case study of the Pearl River Basin, China, *Environmental Research*, 2021, **196**, 110368, doi: 10.1016/j.envres.2020.110368.
[44] M. M. Hamed, M. S. Nashwan, S. Shahid, T. bin Ismail, A. Dewan, M. Asaduzzaman, Thermal bioclimatic indicators over Southeast Asia: present status and future projection using CMIP6, *Environmental Science and Pollution Research*, 2022, **29**, 91212-91231, doi: 10.1007/s11356-022-22036-6.
[45] L. P. Hoang, H. Lauri, M. Kummu, J. Koponen, M. T. H. van Vliet, I. Supit, R. Leemans, P. Kabat, F. Ludwig, Mekong River flow and hydrological extremes under climate change, *Hydrology and Earth System Sciences*, 2016, **20**, 3027-3041, doi: 10.5194/hess-20-3027-2016.
[46] J. R. Quinlan, Learning with continuous classes, in 5th Australian joint conference on artificial intelligence, *World Scientific*, 1992, **92**, 343-348, doi: 10.1142/9789814536271.
[47] D. P. Solomatine, Y. Xue, M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai River in China, *Journal of Hydrologic Engineering*, 2004, **9**, 491-501, doi: 10.1061/(asce)1084-0699(2004)9: 6(491).
[48] P. Ditthakit, S. Pinthong, N. Salaeh, J. Weekaew, T. Thanh Tran, Q. Bao Pham, Comparative study of machine learning methods and GR2M model for monthly runoff prediction, *Ain Shams Engineering Journal*, 2023, **14**, 101941, doi: 10.1016/j.asej.2022.101941.
[49] Y. Wang, I. H. Witten, *Induction of model trees for predicting continuous classes, (Working paper 96/23)*. Hamilton, New Zealand: University of Waikato, Department of Computer Science, 1996. Available: https://hdl.handle.net/10289/1183.
[50] T. K. Ho, Random decision forests, *Proceedings of 3rd international conference on document analysis and recognition*, Montreal, QC, Canada, IEEE, 1995, 278-282, doi: doi: 10.1109/ICDAR.1995.598994.
[51] L. Breiman, Random forests, *Machine Learning*, 2001, **45**, 5-32, doi: 10.1023/A: 1010933404324.
[52] V. Jagannath, Random Forest Template for TIBCO Spotfire®. Retrieved from https://community.tibco.com/wiki/random-forest-template-tibco-spotfire (accessed August 2021).
[53] V. Vapnik, S. E. Golowich, A. Smola, Support vector method for function approximation, regression estimation and signal processing, *Proceedings of the 9th International Conference on Neural Information Processing Systems*, December 3-5, 1996, Denver, Colorado. ACM, 1996, 281-287, doi: 10.5555/2998981.2999021.
[54] V. N. Vapnik, Introduction: four periods in the research of the learning problem, in *The Nature of Statistical Learning Theory*. New York: Springer, 2000: 1-15, doi: 10.1007/978-1-4757-3264-1_1

[55] N. Salaeh, P. Ditthakit, S. Pinthong, M. A. Hasan, S. Islam, B. Mohammadi, N. T. T. Linh, Long-short term memory technique for monthly rainfall prediction in thale sap songkhla river basin, Thailand, *Symmetry*, 2022, **14**, 1599, doi: 10.3390/sym14081599.

[56] T. Kleynhans, M. Montanaro, A. Gerace, C. Kanan, Predicting top-of-atmosphere thermal radiance using MERRA-2 atmospheric data with deep learning, *Remote Sensing*, 2017, **9**, 1133, doi: 10.3390/rs9111133.

[57] W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, *The Bulletin of Mathematical Biophysics*, 1943, **5**, 115-133, doi: 10.1007/BF02478259.

[58] M. Minsky, S. A. Papert, *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry*. Cambridge, MA, USA: MIT press, 2017, 289.

[59] C. W. Dawson, R. L. Wilby, Hydrological modelling using artificial neural networks, *Progress in Physical Geography: Earth and Environment*, 2001, **25**, 80-108, doi: 10.1177/030913330102500104.

[60] J. L. McClelland, D. E. Rumelhart, P. R. Group, *Parallel distributed processing*. Cambridge, MA: MIT press, 1986.

[61] F. Rosenblatt, *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Washington, DC, USA: Spartan books, 1961. Available: https://apps.dtic.mil/sti/citations/AD0256582

[62] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning Internal Representations by Error Propagation, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, *Biometrika*, 1986, **71**, 599-607.

[63] J. Koza, J. P. Rice, Automatic programming of robots using genetic programming, *Proceedings of the Tenth National Conference on Artificial Intelligence*. San Jose, CA, USA, 1992, 194-201

[64] S. Bouktif, A. Fiaz, A. Ouni, M. A. Serhani, Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting, *Energies*, 2020, **13**, 391, doi: 10.3390/en13020391.

[65] H. Zhu, H. Zeng, J. Liu, X. Zhang, Logish: A new nonlinear nonmonotonic activation function for convolutional neural network, *Neurocomputing*, 2021, **458**, 490-499, doi: 10.1016/j.neucom.2021.06.067.

[66] S. Poornima, M. Pushpalatha, Prediction of rainfall using intensified LSTM based recurrent neural network with weighted linear units, *Atmosphere*, 2019, **10**, 668, doi: 10.3390/atmos10110668.

[67] D. N. Moriasi, J. G. Arnold, M. W. Van Liew, R. L. Bingner, R. D. Harmel, T. L. Veith, Model evaluation guidelines for systematic quantification of accuracy in watershed simulations, *Transactions of the ASABE*, 2007, **50**, 885-900, doi: 10.13031/2013.23153.

[68] C. Santhi, J. G. Arnold, J. R. Williams, W. A. Dugas, R. Srinivasan, L. M. Hauck, Validation of the swat model on a large rwer basin with point and nonpoint sources1, *JAWRA Journal of the American Water Resources Association*, 2001, **37**, 1169-1188, doi: 10.1111/j.1752-1688.2001.tb03630.x.

[69] M. W. Van Liew, T. L. Veith, D. D. Bosch, J. G. Arnold, Suitability of SWAT for the conservation effects assessment project: comparison on USDA agricultural research service watersheds, *Journal of Hydrologic Engineering*, 2007, **12**, 173-189, doi: 10.1061/(asce)1084-0699(2007)12: 2(173).

[70] R. J. Hyndman, A. B. Koehler, Another look at measures of forecast accuracy, *International Journal of Forecasting*, 2006, **22**, 679-688, doi: 10.1016/j.ijforecast.2006.03.001.

[71] T. Chai, R. R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE)? –Arguments against avoiding RMSE in the literature, *Geoscientific Model Development*, 2014, **7**, 1247-1250, doi: 10.5194/gmd-7-1247-2014.

[72] R. G. Pontius, O. Thontteh, H. Chen, Components of information for multiple resolution comparison between maps that share a real variable, *Environmental and Ecological Statistics*, 2008, **15**, 111-142, doi: 10.1007/s10651-007-0043-y.

[73] C. J. Willmott, K. Matsuura, On the use of dimensioned measures of error to evaluate the performance of spatial interpolators, *International Journal of Geographical Information Science*, 2006, **20**, 89-102, doi: 10.1080/13658810500286976.

[74] D. R. Legates, G. J. McCabe Jr, Evaluating the use of "goodness-of-fit" Measures in hydrologic and hydroclimatic model validation, *Water Resources Research*, 1999, **35**, 233-241, doi: 10.1029/1998wr900018.

[75] M. J. S. Safari, S. Rahimzadeh Arashloo, A. Danandeh Mehr, Rainfall-runoff modeling through regression in the reproducing kernel Hilbert space algorithm, *Journal of Hydrology*, 2020, **587**, 125014, doi: 10.1016/j.jhydrol.2020.125014.

[76] M. Buscema, M. Breda, W. Lodwick, Training with input selection and testing (TWIST) algorithm: a significant advance in pattern recognition performance of machine learning, *Journal of Intelligent Learning Systems and Applications*, 2013, **5**, 29-38, doi: 10.4236/jilsa.2013.51004.

[77] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: theory and practice, *Neurocomputing*, 2020, **415**, 295-316, doi: 10.1016/j.neucom.2020.07.061.

[78] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, *Neural Computation*, 2000, **12**, 2451-2471, doi: 10.1162/089976600300015015.

[79] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, Cambridge, Massachusetts: MIT press, 2016, 429.

[80] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, 2014, **15**, 1929-1958.

[81] S. Abreu, Automated architecture design for deep neural networks, *arXiv preprint arXiv: 1908.10714*, 2019, doi: 10.48550/arXiv.1908.10714

[82] M. H. J. P. Gunarathna, K. Sakai, T. Nakandakari, K. Momii, M. K. N. Kumari, Machine learning approaches to develop pedotransfer functions for tropical Sri Lankan soils, *Water*, 2019, **11**, 1940, doi: 10.3390/w11091940.

[83] K. Kar, N. Thakur, P. Sanghvi, Prediction of rainfall using fuzzy dataset, *International Journal of Computer Science and Mobile Computing*, 2019, **8**, 182-186.

[84] M. Gauch, J. Mai, J. Lin, The proper care and feeding of CAMELS: how limited training data affects streamflow prediction, *Environmental Modelling & Software*, 2021, **135**, 104926, doi: 10.1016/j.envsoft.2020.104926.

[85] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, Md. Mostofa Ali Patwary, Y. Yang, Y. Zhou, Deep learning scaling is predictable, empirically, *arXiv preprint arXiv:1712.00409*, 2017, doi: 10.48550/arXiv.1712.00409.

[86] R. C. Staudemeyer, E. R. Morris, Understanding LSTM: a tutorial into Long Short-Term Memory Recurrent Neural Networks, *arXiv preprint arXiv: 1909.09586*, 2019, doi: 10.48550/arXiv.1909.09586

[87] S. Lufi, S. Ery, R. Rispiningtati, Hydrological analysis of TRMM (tropical rainfall measuring mission) data in lesti sub watershed, *Civil and Environmental Science*, 2020, **3**, 18-30, doi: 10.21776/ub.civense.2020.00301.3.

[88] A. Sharafati, K. Khosravi, P. Khosravinia, K. Ahmed, S. A. Salman, Z. M. Yaseen, S. Shahid, The potential of novel data mining models for global solar radiation prediction, *International Journal of Environmental Science and Technology*, 2019, **16**, 7147-7164, doi: 10.1007/s13762-019-02344-0.

[89] S. Misra, S. Sarkar, P. Mitra, Statistical downscaling of precipitation using long short-term memory recurrent neural networks, *Theoretical and Applied Climatology*, 2018, **134**, 1179-1196, doi: 10.1007/s00704-017-2307-2.

[90] S. Cramer, M. Kampouridis, A. A. Freitas, A. K. Alexandridis, An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives, *Expert Systems with Applications*, 2017, **85**, 169-181, doi: 10.1016/j.eswa.2017.05.029.

[91] S. Pour, S. Harun, S. Shahid, Genetic programming for the downscaling of extreme rainfall events on the east coast of peninsular Malaysia, *Atmosphere*, 2014, **5**, 914-936, doi: 10.3390/atmos5040914.

[92] P. Sarzaeim, O. Bozorg-Haddad, A. Bozorgi, H. A. Loáiciga, Runoff projection under climate change conditions with data-mining methods, *Journal of Irrigation and Drainage Engineering*, 2017, **143**, 04017026, doi: 10.1061/(asce)ir.1943-4774.0001205.

[93] S. Supharatid, J. Nafung, T. Aribarg, Projected changes in temperature and precipitation over mainland Southeast Asia by CMIP6 models, *Journal of Water and Climate Change*, 2022, **13**, 337-356, doi: 10.2166/wcc.2021.015.

[94] M. M. Hamed, M. S. Nashwan, S. Shahid, T. bin Ismail, X.-J. Wang, A. Dewan, M. Asaduzzaman, Inconsistency in historical simulations and future projections of temperature and rainfall: a comparison of CMIP5 and CMIP6 models over Southeast Asia, *Atmospheric Research*, 2022, **265**, 105927, doi: 10.1016/j.atmosres.2021.105927.

[95] K. Yoneyama, C. Zhang, Years of the maritime continent, *Geophysical Research Letters*, 2020, **47**, e2020GL087182, doi: 10.1029/2020GL087182.

[96] S. Zhang, J. Chen, Uncertainty in projection of climate extremes: a comparison of CMIP5 and CMIP6, *Journal of Meteorological Research*, 2021, **35**, 646-662, doi: 10.1007/s13351-021-1012-3.

[97] S. R. Sobie, F. W. Zwiers, C. L. Curry, Climate model projections for Canada: a comparison of CMIP5 and CMIP6, *Atmosphere-Ocean*, 2021, **59**, 269-284, doi: 10.1080/07055900.2021.2011103.

[98] M. M. Hamed, M. S. Nashwan, S. Shahid, Inter-comparison of historical simulation and future projections of rainfall and temperature by CMIP5 and CMIP6 GCMs over Egypt, *International Journal of Climatology*, 2022, **42**, 4316-4332, doi: 10.1002/joc.7468.

[99] Y. Du, D. Wang, J. Zhu, D. Wang, X. Qi, J. Cai, Comprehensive assessment of CMIP5 and CMIP6 models in simulating and projecting precipitation over the global land, *International Journal of Climatology*, 2022, **42**, 6859-6875, doi: 10.1002/joc.7616.

[100] M. Kamruzzaman, S. Shahid, T. Islam A., S. Hwang, J.-P. Cho, Asad Uz Zaman M., Ahmed M., Mizanur Rahman M., M. B. Hossain, Comparison of CMIP6 and CMIP5 model performance in simulating historical precipitation and temperature in Bangladesh: a preliminary study, *Theoretical and Applied Climatology*, 2021, **145**, 1385-1406, doi: 10.1007/s00704-021-03691-0.

[101] J. Hu, J. Ma, C. Nie, L. Xue, Y. Zhang, F. Ni, Y. Deng, J. Liu, D. Zhou, L. Li, Z. Wang, Attribution Analysis of Runoff Change in Min-Tuo River Basin based on SWAT model simulations, China, *Scientific Reports*, 2020, **10**, 2900, doi: 10.1038/s41598-020-59659-z.

[102] F. Zeng, M.-G. Ma, D.-R. Di, W.-Y. Shi, Separating the impacts of climate change and human activities on runoff: a

review of method and application, *Water*, 2020, **12**, 2201, doi: 10.3390/w12082201.
[103] G. Q. Wang, J. Y. Zhang, Y. Q. Xuan, J. F. Liu, J. L. Jin, Z. X. Bao, R. M. He, C. S. Liu, Y. L. Liu, X. L. Yan, Simulating the impact of climate change on runoff in a typical river catchment of the loess plateau, China, *Journal of Hydrometeorology*, 2013, **14**, 1553-1561, doi: 10.1175/JHM-D-12-081.1.