Machine Learning for Precipitation Forecasts Postprocessing: Multimodel Comparison and Experimental Investigation

YUHANG ZHANG[a] AND AIZHONG YE[a]

[a] *State Key Laboratory of Earth Surface Processes and Resource Ecology, Faculty of Geographical Science, Beijing Normal University, Beijing, China*

(Manuscript received 15 May 2021, in final form 27 August 2021)

ABSTRACT: Obtaining high-quality quantitative precipitation forecasts is a key precondition for hydrological forecast systems. Due to multisource uncertainties (e.g., initial conditions, model structures, and parameters), raw forecasts are subject to systematic biases; hence, statistical postprocessing is often required to reduce these errors before the forecasts can proceed to hydrological applications. Machine learning (ML) algorithms are canonical statistical models, and they are diverse in type and variation. It is important to verify and compare their performance in the same scenario (e.g., precipitation postprocessing). In this paper, we conduct a large-scale comparison study for the major ML models with diverse model structures and regularization strategies as postprocessors for improving the quality of precipitation forecasts. Specifically, we compare the efficiency and effectiveness of 21 ML algorithms on solving this task. Daily reforecast precipitation with lead times up to 8 days from the Global Ensemble Forecast System and corresponding observations are employed to determine the usability of different models in the Yalong River basin in China. The performance of each model is validated by a group of carefully designed experiments and statistical metrics. The results reveal that improvements in model structures are more effective than regularization strategies. Among these algorithms, the optimized extra-trees regressor exhibits the best performance, effectively reducing overestimation and achieving the best skill in forecasting precipitation. Eleven ensemble members and a 3-day forward-rolling time window can be used as predictors to obtain the best model performance. The systematic experiments and findings also offer useful guidelines for other related studies.

KEYWORDS: Numerical weather prediction/forecasting; Postprocessing; Regression analysis

1. Introduction

Obtaining skillful and accurate quantitative precipitation forecasts (QPFs) is not only a primary goal of operational prediction centers but also a critical necessity for integrated hydrometeorological forecast systems (Fritsch and Carbone 2004; Yuan et al. 2015; Ye et al. 2017). There are many uncertainties in the operation of numerical weather prediction models (NWPs) (Lorenc 1986; Demargne et al. 2014; Li et al. 2019). It is necessary to develop a statistical postprocessing model to improve their skill and reliability as much as possible in both meteorological and hydrological studies (Guan et al. 2015; Ye et al. 2014, 2015). Statistical postprocessing refers to the establishment of a statistical model between historical observations and corresponding reforecast pairs (Medina et al. 2019; Li et al. 2017; Piani et al. 2010; Hamill and Whitaker 2006). Machine learning (ML) algorithms are canonical statistical postprocessing model for precipitation forecast.

Based on statistical learning theory, machine learning (ML) algorithms extract features from multidimensional data, which can fit complex linear or nonlinear relationships, and have been applied to solve or simulate multiple processes or key variables involved in the hydrological cycle, such as precipitation, air temperature, wind speed, radiation, evapotranspiration, soil moisture, and runoff (Oppel and Fischer 2020; Wang et al. 2020; Ghaith et al. 2020; Zhao et al. 2019; Zhang et al. 2019; Voyant et al. 2017; Raghavendra and Deka 2014; Nourani et al. 2014). The advantages of ML methods are as follows: (i) there are various methods which are easy to implement simultaneously in one framework (e.g., scikit-learn); (ii) they can map flexible relationships between input and target variables; (iii) they can incorporate diverse features and automatically extract useful information; and (iv) they do not need very strict assumptions as numerical methods. These characteristics of ML models in postprocessing have also been confirmed in a few studies. For example, the neural network has been selected and compared with the XGBoost algorithm in the postprocessing of extended range 2-m maximum air temperature (Peng et al. 2020). The quantile regression forecast (QRF) was investigated and coupled with ensemble copula coupling methods to establish a station-based postprocessing model for surface temperature and a gridded postprocessing model for hourly rainfall amounts in the French national weather service operational forecasting chain (Taillardat et al. 2016; Taillardat and Mestre 2020). The QRF method was also used as a benchmark and was compared with fully connected networks to predict air temperatures in Germany (Rasp and Lerch 2018). Diez-Sierra and Del Jesus (2020) applied and compared multiple ML models including random forests (RF), *K*-nearest neighbors (KNN), neural networks (NN), support vector machines (SVM), and logistic regression to forecast long-term rainfall in Spain. These various ML models (e.g., KNN, NN, SVM) were also employed to reduce the uncertainties of future precipitation and temperature

Supplemental information related to this paper is available at the Journals Online website: https://doi.org/10.1175/JHM-D-21-0096.s1.

Corresponding author: Aizhong Ye, azye@bnu.edu.cn

DOI: 10.1175/JHM-D-21-0096.1

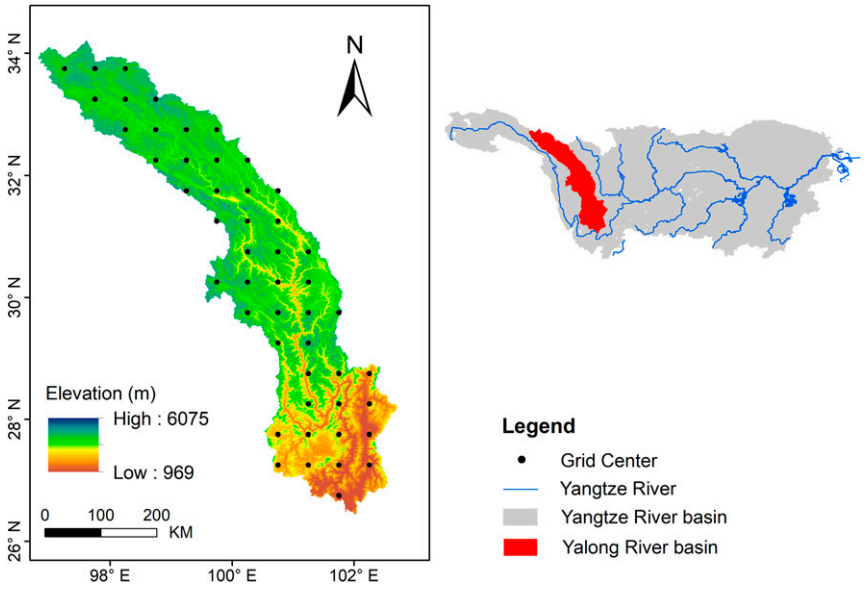

FIG. 1. The location of the Yalong River (YLR) basin.

projections from global climate models (Sachindra et al. 2018; Ahmed et al. 2020). Moreover, such ML models expressed impressive performance for both deterministic and probabilistic streamflow forecasts and postprocessing in large-scale studies. For instance, Tyralis et al. (2020) proposed a super ensemble learning framework that weighted 10 individual ML algorithms, and the proposed super ensemble model was used and compared with individual ML models for daily streamflow mean value predictions in 511 basins in the contiguous United States. Using the same dataset, a series of quantile regression-based individual and ensemble models for hydrological postprocessing successfully modeled the conditional quantiles of a target variable (Papacharalampous et al. 2019; Tyralis et al. 2019). Most of the above studies tend to compare or combine multiple ML models to fit the relationship between the simulations (or predictions) and observations. It not only shows the function of a single ML model as a base learner but also reveals the performance improvement of the ensemble method.

Different from the above studies, in this study, we attempt to enrich the case of machine learning solutions for precipitation forecasts postprocessing. Some of these methods cover more diverse regularization forms and different ensemble rules. Here, all algorithms we choose are easy to implement, so as to comprehensively compare their accuracy, efficiency, and generalization. Moreover, we test them as a regional model applied to small or medium-sized watersheds with complex topography. This is also different from some previous grid-based or station-based postprocessing methods for precipitation forecasts. Another focus of this study is to explore different predictors used for precipitation postprocessing. Based on the above motivations, we choose the Yalong River (YLR) basin with complex climatic and topographic conditions in China and adopt 21 machine learning algorithms to establish postprocessors for precipitation forecasting. Meanwhile, we compared various experimental configurations to select a better combination of predictors to improve our model performance. This paper is organized as follows: section 2 describes the chosen study area and the data used for the study; section 3 introduces the methods, including the machine learning algorithms, data processing, selected evaluation metrics, and experimental design; and sections 4 and 5 present the results and discussion, respectively. The final section presents the conclusions of this paper.

2. Study area and data

a. Study area

The Yalong River, located in the southeastern part of the Qinghai–Tibet Plateau, is the largest tributary of the Jinsha River, and the latter is a major tributary on the upper reaches of the Yangtze River in China. (Fig. 1). The YLR basin spans a wide range of latitudes (26°32′–33°58′N) and longitudes (96°52′–102°48′E) due to its long and narrow shape. The total area of the YLR basin is approximately 136 000 km^2 and the length of the mainstream is approximately 1571 km. The topography in the basin is complex and mainly composed of mountains and valleys. Because of its large elevation differences and latitudinal span (Fig. 2a), the climate in the YLR basin varies with location. The upper area has a continental climate, which is relatively cold and dry. Affected by a subtropical climate, the middle and lower reaches have relatively high temperatures and large rainfall amounts (Fig. 2b).

To facilitate the following comparison between ML methods applied in different climate types, as well as their efficiency in a whole basin, we selected two typical grid cells, as shown in Figs. 2c and 2d. Grid cells 5 and 39 represent the relatively dry and humid climate conditions, respectively.

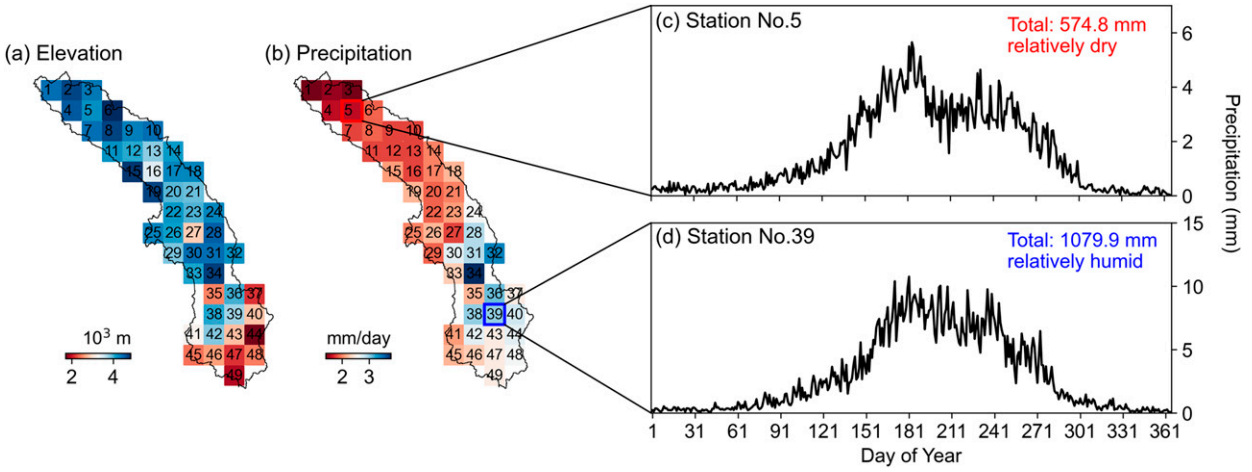

FIG. 2. The spatial distributions of (a) elevation and (b) average precipitation, and the observed climatology of the typical grid cells, (c) grid cell 5, and (d) grid cell 39. The average precipitation in the figure is drawn from the gridded observation data (1985–2019).

b. Data

1) FORECAST DATA

The raw precipitation forecast data were obtained from the Global Ensemble Forecast System reforecast version 2 (GEFSv2) operated by the National Centers for Environmental Prediction of National Oceanic and Atmospheric Administration. This dataset consists of one control forecast and 10 perturbed member forecasts. The model forecasts precipitation for the next 14 days beginning at 0000 UTC from December 1984 to the present. In this study, we downloaded the first 8 days of the forecast from 1985 to 2018. For example, when the forecast date is k, the forecasts are $p(k)$ (lead time 1 day), $p(k+1)$ (lead time 2 day), ..., $p(k+7)$ (lead time 8 day), respectively. The original temporal and spatial resolutions are 3 h and 1° × 1°, respectively (Hamill et al. 2013).

2) REFERENCE DATA

The 0.5° daily observed gridded precipitation dataset (1985–2018) used in this study was provided by the China Meteorological Administration (CMA). This dataset was interpolated and verified using thin plate smooth spline method and the high-quality precipitation data from about 2400 weather stations in China, which is the most accurate gridded dataset available and has been used for climate assessments, flood forecasts, drought predictions, and many other studies (Zhao et al. 2014; Wu et al. 2016; Lu et al. 2017). We choose these data because their spatial resolution corresponds to the interpolated raw precipitation forecasts (0.5° × 0.5°). We are more focused on discussing the feasibility of the ML methods for precipitation postprocessing, the error introduced in the data preprocessing processes is ignored.

3. Methods

The general workflow of a complete machine learning task involves (i) abstracting a real-world problem into a machine learning problem, (ii) extracting input and output features, and (iii) comparing different machine learning models through iterations to select the best model for the application. Here we define our machine learning task as trying to build a postprocessing model (regression model) to correct the bias of raw precipitation forecasts from NWPs. For all selected models, the input data include the raw forecast ensemble members from GEFS, and the output is the precipitation ensemble mean, which is further compared with the observations. Below we introduce the basic information of different ML models, feature selection, and experimental design.

a. Data processing

Data preprocessing mainly includes data extraction, combination, interpolation, and transformation. First, we extracted the CMA and GEFSv2 data using the YLR basin boundary (Fig. 1). The raw GEFSv2 data were then combined to a daily scale and interpolated into a 0.5° × 0.5° grid using the bilinear interpolation to ensure the consistency of the data resolution. Finally, to enable faster training of the ML models, we transformed the raw data to a normal distribution in the training period and inversely transformed the predicted values to compare with observation using the Yeo–Johnson power transformation (Yeo and Johnson 2000), which was accomplished by Eq. (1):

$$x_i^{(\lambda)} = \begin{cases} [(x_i+1)^{\lambda}-1]/\lambda, & \text{if} \quad \lambda \neq 0,\ x_i \geq 0 \\ \ln(x_i+1), & \text{if} \quad \lambda = 0,\ x_i \geq 0 \end{cases}, \tag{1}$$

where λ is a mapping factor, which is estimated by the maximum likelihood method; x_i represents daily precipitation (mm day^{-1}); and $x_i^{(\lambda)}$ is the transformed daily precipitation (mm day^{-1}).

b. Machine learning methods

In this study, we selected 21 widely used ML algorithms (Table 1) for systematic and extensive comparison. In addition, there are three multimodel averaging combinations for the parameterization, and more results can be found in the online supplemental material (Figs. S1–S3). The selected 21

TABLE 1. The ML methods, abbreviations, brief descriptions, and references.

ID	Full name and abbreviation	Brief description	Reference
M1	Multilayer perceptron (MLP)	Also known as an artificial neural network.	Rumelhart et al. (1986)
M2	Gradient boosting regressor (GBR)	An ensemble model based on the gradient boosting machine	Friedman (2001)
M3	CatBoost regressor (CATB)	Similar to the GBR, but optimized for category features	Prokhorenkova et al. (2018)
M4	Light gradient boosting (LGBM)	Similar to XGB, but introduces the histogram algorithm	Ke et al. (2017)
M5	Extreme gradient boosting (XGB)	Similar to the GBR, but optimizes efficiency, generalization, and robustness	Chen et al. (2015)
M6	AdaBoost regressor (ADAB)	An ensemble model based on the gradient boosting machine and weights of sample points	Freund and Schapire (1996)
M7	Random forest (RF)	An ensemble bagging model based on the DT	Breiman (2001)
M8	Extra-trees regressor (ET)	Similar to the RF, but randomness is added	Geurts et al. (2006)
M9	Decision tree (DT)	A binary tree with probabilities	Xu et al. (2005)
M10	*K*-neighbors regressor (KNN)	*K*-nearest neighbors with distances	Altman (1992)
M11	Linear regression (LR)	Original least squares linear regression	—
M12	Ridge regression (RIDGE)	Linear least squares with L2 regularization	Hoerl and Kennard (1970)
M13	Least angle regression (LAR)	Similar to forward stepwise regression	Efron et al. (2004)
M14	Lasso regression (LASSO)	Linear model trained with L1 regularization	Koh et al. (2007)
M15	Elastic net (EN)	Linear regression with combined L1 and L2 regularizations	Owen (2007)
M16	Huber regressor (HUBER)	A linear regression model that is robust to outliers	Huber (2004)
M17	Bayesian ridge (BR)	Similar to ridge, introduces Bayesian inference	Bishop (2006)
M18	Random sample consensus (RANSAC)	Similar to LR, but ignores outliers	Fischler and Bolles (1981)
M19	Passive aggressive regressor (PAR)	A stepwise learning method	Crammer et al. (2006)
M20	Orthogonal matching pursuit (OMP)	A greedy iteration method	Pati et al. (1993)
M21	Support vector machine (SVM)	A soft-margin-based model with kernel transformation	Schölkopf et al. (2002)
M22	Combine1	Equally weighted averaged all 21 ML models	—
M23	Combine2	Equally weighted averaged four tuned ML models (KNN, ET, LGBM, MLP)	—
M24	Combine3	Equally weighted averaged two tuned ML models (ET, LGBM)	—

models can be classified into the following categories according to their characteristics: 1) linear models; 2) soft-margin-based models; 3) decision-tree-based models; 4) KNN models; and 5) artificial neural networks (ANNs). It should be noted that comparing all existing ML models would be a hopeless task, so we only select some typical examples and provide a reference for such a large-scale comparative study. The reasons for selecting these algorithms are as follows:

1) The selected models are widely used and easy to implement. For example, they are completed under the same framework (e.g., scikit-learn). So, it is convenient to compare all models at the same time.
2) They are diverse enough to include linear and nonlinear models, or individual and ensemble models. This allows us to choose the model architecture more clearly. But they have some similarities, such as only different forms of loss functions to prevent overfitting. This allows us to compare whether such regularization strategies are effective.

Below is a short description of these models. More detailed theories and explanations can be found in the references in Table 1.

1) LINEAR MODELS

Linear models are a family of linear regression-based models, and their basic form is as Eqs. (A1)–(A3) in the appendix.

Methods to fit the model include the original least squares linear regression [Eq. (A4)], as well as extended solutions with various regularization, such as lasso regression with L1 regularization [Eq. (A5)], ridge regression with L2 regularization [Eq. (A6)], elastic net in the form of combined L1 and L2 regularization [Eq. (A7)], Huber regression in the form of linear loss [Eqs. (A8) and (A9)], and Bayesian ridge with additional prior information. Besides, we also selected two models similar to stepwise regression that utilize the correlation between features and targets, namely, the least angle regression (LAR) and the orthogonal matching pursuit (OMP). They differ in that the OMP has stronger sparsity by assuming nonzero terms before fitting the model.

2) SOFT-MARGIN-BASED MODELS

According to a hypothetical tolerance interval ε, the soft-margin-based model partitions the data into two parts, inliers and outliers. The model is then fitted based on the inliers while ignoring the effect of the outliers, so the model can be generalized more easily. Here, we selected three soft-margin-based ML algorithms with different complexity, which are random sample consensus (RANSAC), passive aggressive regressor (PAR), and support vector machine (SVM). RANSAC uses completely random sampling to classify the original data, while PAR is based on hinge loss function [Eq. (A10)] to complete the segmentation of inliers and outliers. Compared with PAR, the SVM additionally introduces a kernel function that can map nonlinear features to a high-dimensional space to obtain linear features. Support vector regression (SVR) is the version of SVM used for the regression task.

3) DECISION-TREE-BASED MODELS

Decision tree (DT) is a nonparametric supervised learning method used for classification (also known as classification tree) and regression (also known as regression tree). The regression tree continuously approximates the target by dividing the feature space by nodes and weighting the average from different subspace. DTs are often used as a base learner to obtain ensemble learning models through different strategies.

One commonly used ensemble learning strategy is to combine the DTs through bagging. Bagging is a parallelized method, that is, there is no dependency between the base learners. Such models include random forest (RF) and extra-trees regression (ET). Random forest uses bootstrap random sampling of training data and features to obtain multiple decision tree models, and then the final prediction is obtained by averaging multiple base learners. The extra-trees regression uses all data to train different base learners, and the node splitting is more randomized.

Another ensemble learning strategy aggregates the DTs by boosting, including adaptive boosting regressor (ADAB), gradient boosting regressor (GBR), extreme gradient boosting (XGB), light gradient boosting (LGBM), and categorical boosting regressor (CATB). Boosting is a serialized model, which is trained by stepped iteration. In the training period, only one base learner is trained for each round, and a strong regressor with higher accuracy is finally obtained. ADAB determines the weight of the base model by iteratively adjusting the weight of the sample and finally gets the ensemble model. GBR is a forward algorithm that uses gradient descent. In each iteration of training, the negative gradient of the loss function under the current model is estimated, so that the parameters are updated in the direction of minimizing the loss function. XGB is a further improvement to GBR, including 1) deriving a second-order Taylor expansion of the loss function and 2) adding a regular term to it. These improvements control the complexity of the model and help prevent overfitting. LGBM is a distributed and efficient framework that further develops the XGB model. It generates decision trees using leaf-wise splitting, finds feature split points through histogram-based algorithms, and supports parallel learning, which can process big data more efficiently. CATB better handles discrete categorical features, which enriches the feature dimensions.

4) *K*-NEAREST NEIGHBOR

The KNN algorithm is a nonparametric machine learning model, which is similar to the analog method. In the training period, k historical records similar to the forecast are found in the sample space, and the prediction is obtained by weighting calculation according to the k historical records. The similarity is estimated by the Euclidean distance.

5) ARTIFICIAL NEURAL NETWORKS

The artificial neural network (ANN) is one of the most widely used ML algorithms. It uses different features as input and maps the nonlinear relationship between input variables and target through neurons and activation functions. Here, we use a typical neural network model called multilayer perceptron (MLP).

c. Feature selection

A reasonable postprocessing model is a prerequisite for maximizing the values of NWPs, and postprocessing models are often constructed based on the statistical relationship between the observations and the predictors. Therefore, whether there is a good correlation between the predictors and the observations directly affects the skill of postprocessing models.

Predictors can be selected from various aspects, such as the element field, space, and time. In this study, we selected and validated different combinations of predictors to determine their feasibility. The principles we adhere to are as follows: 1) to simplify the complexity of the model and 2) to include as much valid information as possible. So, we only selected the precipitation ensemble forecasts members as predictors without considering the additional auxiliary variables. Meanwhile, feature engineering is performed by controlling the number of ensemble members and the spatial and temporal information of precipitation forecasts.

First, for the ensemble members, we tested the correlation skill between different ensemble members and observations. Figure 3a indicates that the correlation skill between the observations and all alternative ensemble predictors is greater than 0.4 for the 6-day lead time. Such a high correlation skill between the ensemble mean, ensemble standard deviation, and observations can even last up to 8 days. For different grid cells, the correlation skill varies with the ensemble members. Figures 3b

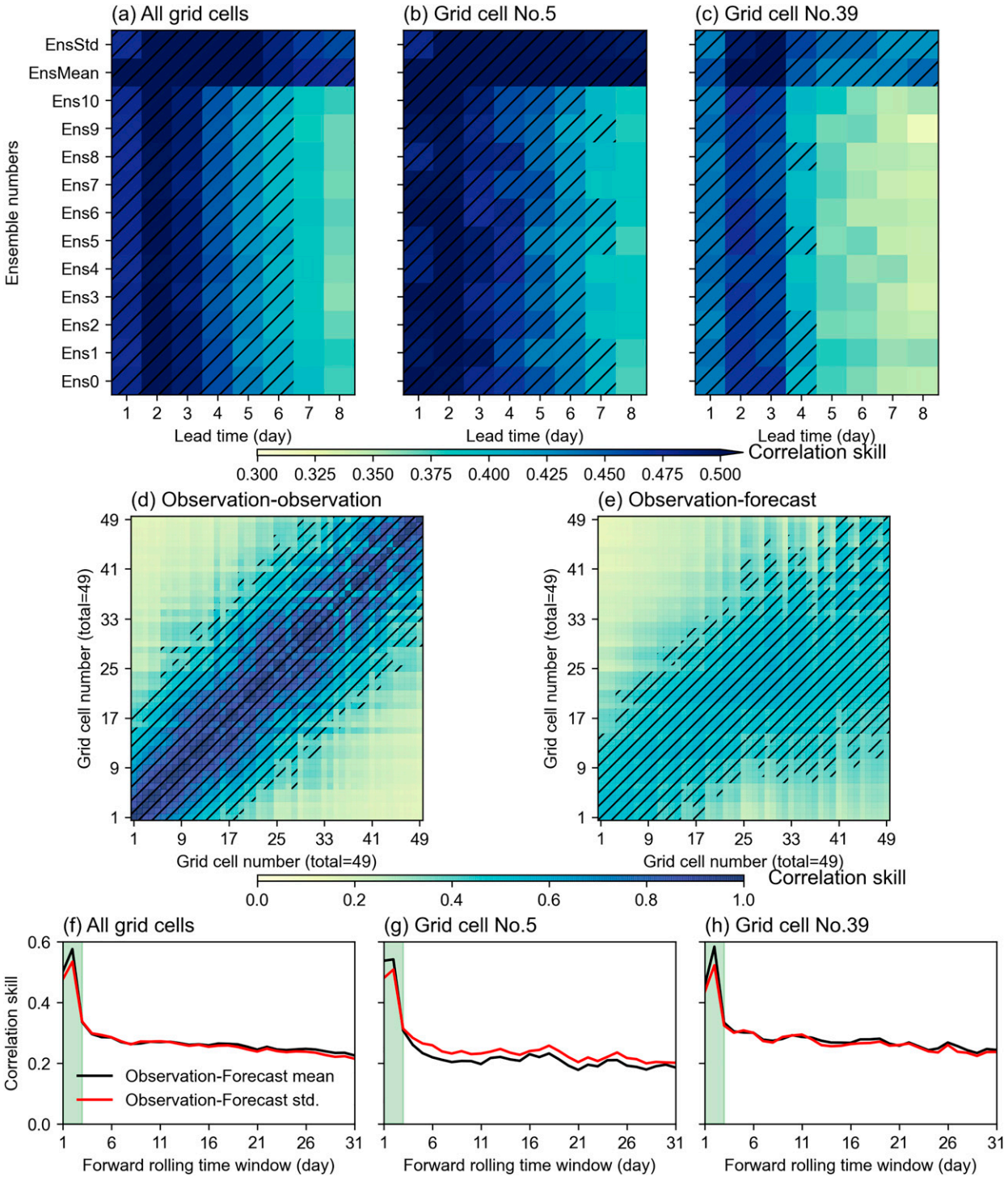

FIG. 3. Correlation skills of different predictors. (a) The regional averaged correlation skill between ensemble members and observations with lead times up to 8 days. (b),(c) The correlation skill between ensemble members and observations with lead times up to 8 days at grid cells 5 and 39, respectively. (d) The correlation skill between observations at a specific grid cell and observations at the other grid cells. (e) The correlation skill between observations at a specific grid cell and forecasts at the other grid cells. (f) The regional averaged correlation skill between the observations on a specific day and forecasts ahead of that day. (g),(h) The correlation skill between the observations on a specific day and forecasts ahead of that day at grid cells 5 and 39, respectively. The hatching in (a)–(e) highlights the correlation skill exceeding 0.4.

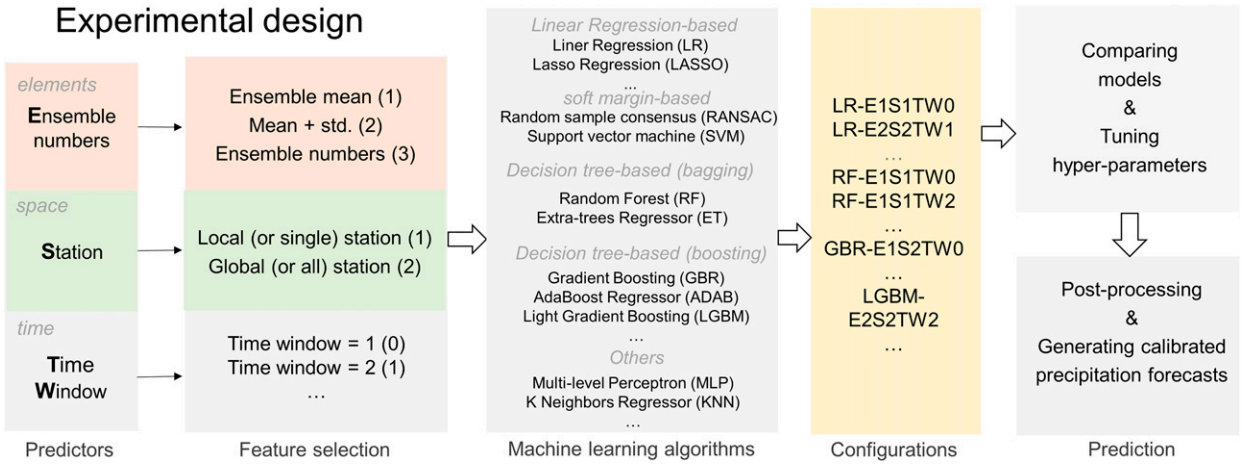

FIG. 4. Experimental design.

and 3c indicates that the correlation skills at grid cell 5 are higher than those at grid cell 39.

The search for predictors in the spatial dimension is based on the following two scenarios: 1) the center of precipitation forecasts may be shifted, and 2) similarities exist in regional precipitation patterns (e.g., between neighboring grid cells) (Scheuerer and Hamill 2015). Here, we examined the correlation skill between the observations among different grid cells (Fig. 3d) and the correlation skill between the observations and forecasts of the surrounding grid cells (Fig. 3e) to explore the similarities among different grid cells. Combining the correlation skill between the observations and forecasts, almost all specific grid cells have more than 20 similar grid cells (except grid cells near the basin boundary). There is a broad range of correlation skills for the precipitation forecasts (Fig. 3e). Therefore, considering that we selected a relatively small watershed and to simplify the complexity of the model, we selected all the data within the basin to build regional models.

For the time dimension, in general, the forecast skills of NWPs decrease with increasing lead times. And, there are no corresponding observations for the future forecasts of the operated NWPs. So, we only considered forecasts from the day of interest to n days before (n = 1–31) to identify the appropriate predictors. The results (Figs. 3f–h) show that $n = 3$ is a suitable choice for all grid cells or specific grid cells. This strategy is forward rolling time window (FRTW).

d. Experimental design and performance metrics

After the preliminary selection of predictors, we further designed several comparative experiments to verify whether the valid predictor information could be transferred to the postprocessing models. The entire experimental design framework is shown in Fig. 4, while the input and output variables, feature selection, and training sample size for each experiment are shown in Table 2.

For the selection of the ensemble members, we set up three sets of experiments (including the ensemble mean (EnsMean), the ensemble mean (EnsMean) and standard deviation (EnsStd), and all ensemble members.

For the selection of the spatial predictors, because the YLR basin is relatively small, forecasts for all grid cells in the basin were selected as predictors to achieve the following two objectives: 1) simplify the complexity of the model and fully compare the performance of different ML models and 2) determine whether a regional model can be applied to local grid cells.

For the selection of the forward rolling time window (FRTW), we set up two experiments (FRTW = 1 day and FRTW = 3 days). For example, when the target forecast is $p(k)$, the input variables are $f(k, k)$ with FRTW = 1, and $f(k, k)$, $f(k-1, k)$, $f(k-2, k)$ with FRTW = 3, respectively. In addition, 10-yr (2009–18) and 20-yr (1999–2018) data were selected to compare the effects of different sample sizes on model performance.

According to the above experimental design, we compared the effects of different input features and sample sizes on the model skills. The tenfold cross-validation method (also known as the leave-one-out method) was used to calibrate and evaluate the model accuracy. For example, the training set (2009–18, 178 948 samples) was randomly divided into 10 groups, of which nine groups (161 053 samples) were used for training and one group (17 895 samples) was used for testing; this process was iterated 10 times. Finally, we averaged the 10 results to evaluate the model performance. The Pearson correlation coefficient (PCC), mean absolute error (MAE), and root-mean-square error (RMSE) were used as performance metrics, and their formulas are given by Eqs. (2)–(4):

$$\mathrm{PCC} = \frac{\sum_{i=1}^{n}(O_i - \overline{O})(S_i - \overline{S})}{\sqrt{\sum_{i=1}^{n}(O_i - \overline{O})^2}\sqrt{\sum_{i=1}^{n}(S_i - \overline{S})^2}}, \tag{2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|S_i - O_i|, \tag{3}$$

$$\mathrm{RMSE} = \left[\frac{1}{n}\sum_{i=1}^{n}(S_i - O_i)^2\right]^{1/2}, \tag{4}$$

TABLE 2. Experimental design.

ID	Input	Target	Period	Ensemble member	Time window	Sample size	Feature size
EX1	$f(k, k)$	$p(k)$	2009–18	EnsMean	1	178 948	1
EX2	$f(k, k)$	$p(k)$	1999–2018	EnsMean	1	357 945	1
EX3	$f(k, k)$ $f(k, k-1)$ $f(k, k-2)$	$p(k)$	2009–18	EnsMean	3	178 948	3
EX4	$f(k, k)$	$p(k)$	2009–18	EnsMean EnsStd	1	178 948	2
EX5	$f(k, k)$	$p(k)$	1999–2018	EnsMean EnsStd	1	357 945	2
EX6	$f(k, k)$ $f(k, k-1)$ $f(k, k-2)$	$p(k)$	2009–18	EnsMean EnsStd	3	178 948	6
EX7	$f(k, k)$	$p(k)$	2009–18	11 ensemble members	1	178 948	11
EX8	$f(k, k)$	$p(k)$	1999–2018	11 ensemble members	1	357 945	11
EX9	$f(k, k)$ $f(k, k-1)$ $f(k, k-2)$	$p(k)$	2009–18	11 ensemble members	3	178 948	33

where O is defined as the observation, S is defined as the forecasts, n is the length of the sequence, and i is the day of interest.

The PCC measures the linear correlation between observations and forecasts. The MAE and RMSE measure the differences between observations and forecasts and therefore provide the overall performance of the model. A high PCC indicates a good linear correlation between the observations and forecasts, and low RMSE and MAE values indicate that the forecast is fitted well to the observations. The RMSE is more susceptible to extreme values than the other indices are.

To eliminate differences in the magnitude of precipitation and to quantify the improvement of the ML models, we also introduced the PCC skill score (PCCSS), MAE skill score (MAESS), and RMSE skill score (RMSESS), which are given by Eqs. (5)–(7):

$$\mathrm{PCCSS} = \frac{\mathrm{PCC}_{\mathrm{ml}}}{\mathrm{PCC}_{\mathrm{ref}}} - 1, \tag{5}$$

$$\mathrm{MAESS} = 1 - \frac{\mathrm{MAE}_{\mathrm{ml}}}{\mathrm{MAE}_{\mathrm{ref}}}, \tag{6}$$

$$\mathrm{RMSESS} = 1 - \frac{\mathrm{RMSE}_{\mathrm{ml}}}{\mathrm{RMSE}_{\mathrm{ref}}}, \tag{7}$$

where ref represents a reference forecast and ml represents the ML models. High PCCSS, MAESS, and RMSESS values indicate optimal improvements of the ML models in terms of the reference method.

To compare the computational efficiencies of the different models, we also considered the time required to train the models. Our experimental platform was based on a personal desktop PC with an Inter(R) Core (TM) i7–7700 CPU @ 3.6 GHz.

4. Results

a. Baseline experiments

First, for the subsequent model selection and optimization, we obtained the results of our baseline experiments based on the experimental designs indicated above and the selected machine learning algorithms. In this stage, we only compared the effects of sample sizes and predictor features. Therefore, the default parameters of all models were used to fit the ML models. All-important default parameters are listed in the Table S1. Finally, we obtained 189 postprocessing models (Fig. 5). The vertical axis includes all 21 ML models shown in Table 1 (without multimodel combination), and the horizontal axis includes all experimental designs shown in Table 2. The evaluation metrics used here not only included the three metrics we introduced in section 3c (PCC, MAE, and RMSE) but also comprised the run time, which was considered to embody the model efficiency (Fig. 5d). We compared models by controlling different variables. In detail, we compared EX1 and 2 (or EX4 and 5, or EX7 and 8) to obtain the effects of different training sizes on model skill. We compared the effect of different ensemble members on model accuracy with EX1, 4, and 7 (or EX2, 5, and 8, or EX3, 6, and 9). The effects of the forward rolling time window (FRTW) on model accuracy were compared by EX1 and 3 (or EX4 and 6, or EX7 and 9).

From the results in Fig. 5, we can draw the following conclusions:

1) For all ML methods, compared to the sample size, the use of a forward rolling time window during the training period significantly increased model performance. The ensemble members had a relatively inconspicuous effect on model performance. The selected predictor EX9 was the best-performing model configuration among all experiments (Figs. 5a–c).
2) When comparing the performances of the different ML models (Figs. 5a–c), it was found that more complex models, such as those including ensemble learning methods (e.g., RF and LGBM), usually significantly outperformed linear regression-based models, especially when more features were added (e.g., LGBM-EX9).
3) In terms of the runtime necessary for model training (Fig. 5d), in general, one model with a simple structure

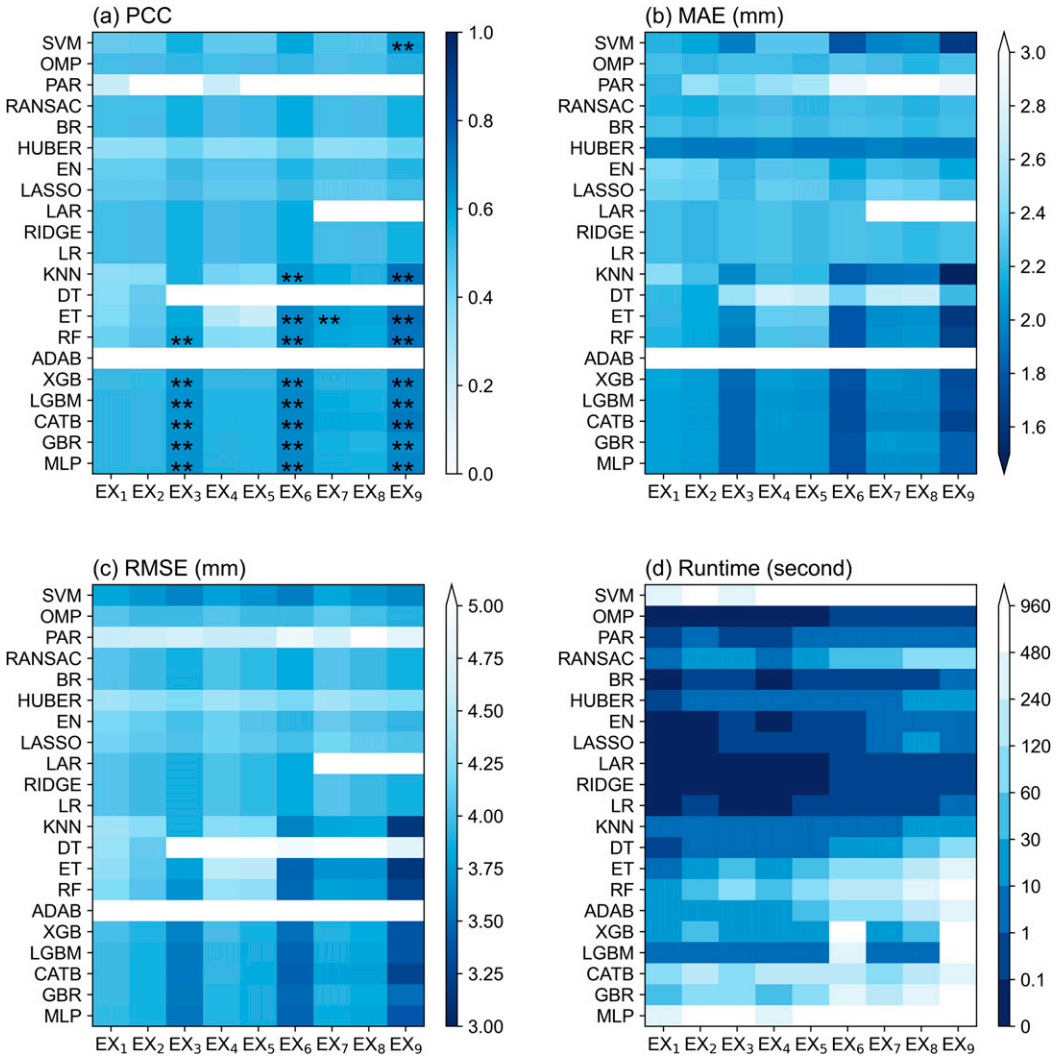

FIG. 5. Model performances in baseline experiments (only for lead time 1 day). (a) Pearson correlation coefficient (PCC). (b) Mean absolute error (MAE). (c) Root-mean-square error (RMSE). (d) Runtime required for one tenfold cross validation. In all panels, a darker color (blue) indicates a better model performance. The "**" in (a) highlights PCCs exceeding 0.6.

and a few samples and features required short runtime. However, there were exceptions, such as LGBM.

By comparing the results of the baseline experiments, we selected five models from different categories for further analysis, namely, LR, ET, KNN, LGBM, and MLP. The LR model is the simplest linear model. KNN is a model that is similar to the analog method and has fewer hyperparameters and can handle high-dimensional datasets. The ET model is an extended version of the RF algorithm and is more robust. The LGBM model is a member of the gradient boosting decision tree (GBDT) family of methods. The algorithm is highly parallelized and can be trained by both CPUs and GPUs. The MLP is the most primitive shallow artificial neural network algorithm and is a long-standing model that is widely used. Moreover, we followed the configuration of EX9, with a 10-yr training sample size (2009–18), a 3-day forward rolling time window, and 11 ensemble members as predictors.

b. Comprehensive performance with model hyperparameter optimization

In the baseline experiments in section 4a, we focused on comparing the performance of different model structures and feature selection strategies without tune the model hyperparameters. Therefore, in this subsection, we select and further tuned the four representative models to achieve their individually optimized model configurations (the LR model has no hyperparameters), and evaluated model performance. For each model, with RMSE as the objective function, we used the grid search and the 10-fold cross-validation method. The optimization of hyperparameters is divided into two steps. In the first step, we change a single parameter and screen out the sensitive parameters in the model by comparing the model performance. In the second step, we use grid search to continuously adjust the sensitive parameters until the

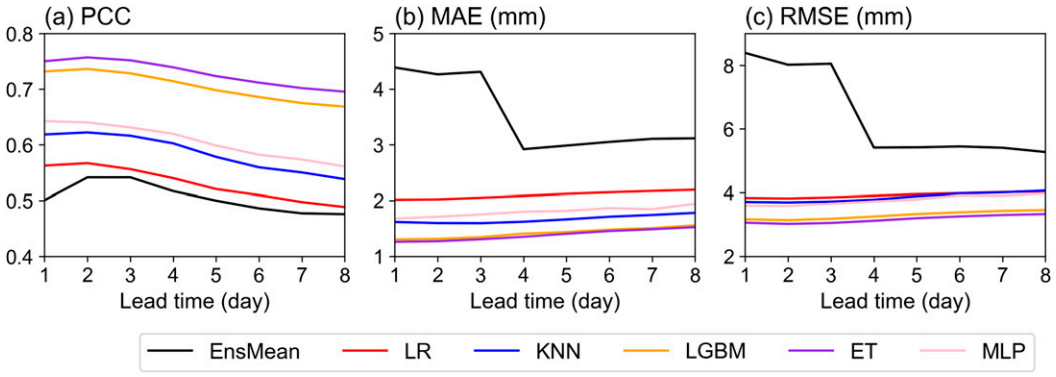

FIG. 6. Regional average model performances with lead times up to 8 days. (a) Pearson correlation coefficient (PCC). (b) Mean absolute error (MAE). (c) Root-mean-square error (RMSE). EnsMean in the figure represents the mean of the raw ensemble members.

model performance no longer changes significantly to finally obtain the optimal parameters. The sensitive parameters, parameter space, and optimum are shown in Table S2 in the supplemental material.

Based on the final models, we obtained the postprocessed precipitation forecasts and the overall model performances and improvements relative to the raw forecasts (Fig. 6 and Table 3). In general, the precipitation forecasting skill deteriorates as the increase of forecasting period. Therefore, to ensure the accuracy of the model, for each leading day, we train independent models. All five selected machine learning models improved the raw forecasts significantly. Among them, ET achieved the best performance, with an average RMSESS = 0.48 ($RMSE_{avg}$ from 6.43 mm from the raw forecast to 3.17 mm from the postprocessed model) for a lead time of 8 days. The performance of LGBM was close to that of ET, ranking second among the five models.

In terms of model performances under lead times ranging from 1 to 8 days, a very promising finding was that ML methods were able to correct original biases to a relatively small degree regardless of the lead time. For example, ET corrected the bias (MAE) to 1.385 ± 0.13 mm, with MAESS = 0.59 ± 0.12.

Figure 7 shows the regional average model performances of different ML methods on each day of the year. We can clearly observe that there were significant seasonal differences in the forecasting skill of the GEFS raw forecast; the model performed better skill from November to April and poorer skill from May to October. The use of different ML methods improved the accuracy of precipitation forecasts. To quantify this seasonal difference, we further calculated evaluation metrics for the monsoon (May–October) and nonmonsoon seasons (November–April) (Table 4).

The average PCC between the raw forecasts and observations displayed poor skill throughout the year, and that during the monsoon season was better than that during the nonmonsoon season. The bias (MAE) of the raw forecast presented an opposite result, with lower biases in the nonmonsoon season (MAE = 1.1 mm) than in the monsoon

TABLE 3. Regional average forecast skill scores of the different ML methods over 1–8 lead days. Here, we take the raw forecast ensemble mean as a reference in Eqs. (5)–(7).

		Lead time (day)							
Metric	Model	1	2	3	4	5	6	7	8
PCCSS	LR	0.123	0.047	0.027	0.045	0.041	0.048	0.042	0.027
	KNN	0.235	0.149	0.138	0.166	0.157	0.151	0.152	0.131
	LGBM	0.461	0.359	0.345	0.382	0.396	0.410	0.414	0.404
	ET	0.498	0.398	0.388	0.430	0.447	0.463	0.470	0.461
	MLP	0.283	0.182	0.165	0.199	0.197	0.197	0.201	0.179
MAESS	LR	0.541	0.526	0.524	0.286	0.290	0.296	0.301	0.296
	KNN	0.631	0.625	0.629	0.444	0.442	0.438	0.438	0.428
	LGBM	0.702	0.691	0.688	0.517	0.518	0.514	0.514	0.500
	ET	0.712	0.701	0.696	0.537	0.528	0.522	0.521	0.510
	MLP	0.617	0.599	0.593	0.382	0.391	0.388	0.405	0.376
RMSESS	LR	0.543	0.524	0.522	0.280	0.271	0.269	0.257	0.234
	KNN	0.558	0.539	0.538	0.302	0.283	0.270	0.259	0.229
	LGBM	0.623	0.608	0.605	0.399	0.385	0.378	0.365	0.344
	ET	0.635	0.623	0.621	0.424	0.409	0.402	0.389	0.368
	MLP	0.571	0.552	0.546	0.312	0.301	0.282	0.278	0.246

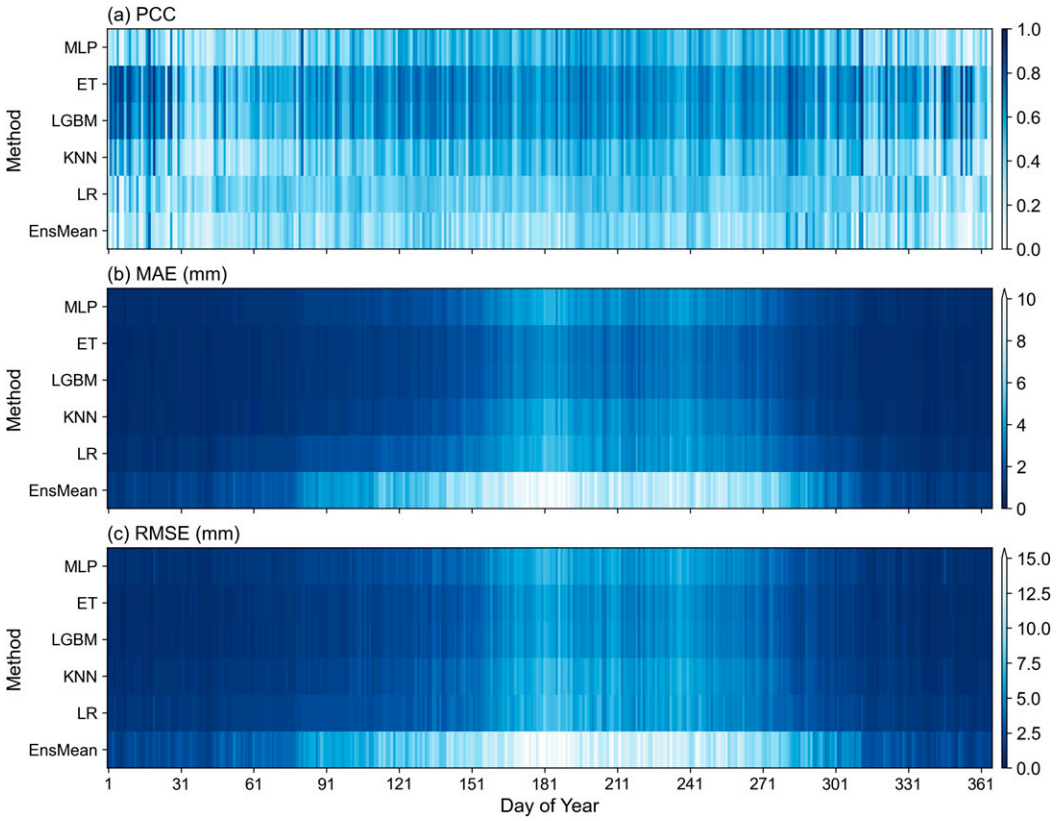

FIG. 7. Regional average model performances of different ML methods on each day of the year (only for lead time 1 day). (a) Pearson correlation coefficient (PCC). (b) Mean absolute error (MAE). (c) Root-mean-square error (RMSE). In all panels, a darker color (blue) indicates a better model performance. EnsMean in the figure represents the mean of the raw ensemble members.

season (MAE = 5.1 mm). This is because the monsoon season is associated with heavy precipitation events and the heteroscedasticity of precipitation forecasts that is implicit in the NWPs.

The use of the ML methods reduced the bias and improved the correlation skill between the forecasts and the observations. The ET model still achieved the greatest model skills, with an annual average of 1.169 for PCCSS, 0.51 for MAESS, and 0.395 for RMSESS. After postprocessing, the MAE was reduced to 1.53 mm.

c. The spatial patterns of precipitation forecasts

From the above analysis, we can draw a preliminary conclusion that for regional averaging, ML methods, especially the ET model, can greatly improve the accuracy of precipitation forecasts. In our proposed framework, we used all gridcell data at once as inputs for model training, which is the so-called regional model, and thus, we can obtain bias-corrected forecasts for all grid cells at once. Unlike previous postprocessing methods that trained models at each grid cell to ensure local accuracy, there may be several problems with the regional model. For example, how well does the regional model simulate precipitation at a specific grid cell? The amount of bias which exists? To what extent are the postprocessing predictions improved compared to the raw forecasts? Are there differences between the annual and seasonal scales? In this subsection, we attempted to answer these questions.

Figure 8 shows the spatial distributions of precipitation forecasts in the YLR basin, both at annual and seasonal scales, obtained from the observations (OBS), the GEFS raw forecasts (EnsMean), and the ML methods. All ML methods, even the raw forecasts, were able to capture the spatial pattern of precipitation. Precipitation is distributed relatively scarce in the upper reaches and more abundant in the middle and lower areas. However, there was a certain deviation in the precipitation forecasts obtained from different methods, and this deviation also varied with location and season. In general, the raw forecast expressed the largest bias, while the ET forecast displayed the smallest bias and was almost consistent with the observations.

On a seasonal scale, precipitation showed the largest bias in summer and the minimum bias in autumn. Except for the raw forecast, all ML methods underestimated the magnitude of summer precipitation in downstream area of the basin. This is a potential problem caused by data imbalances. Although we used 10 years of data for training, there were still very few samples of extreme precipitation events compared to the number of samples of slight precipitation events.

TABLE 4. Model performances and skill scores during the different periods. The monsoon season refers to May–October, and the nonmonsoon season refers to November–April.

		EnsMean	LR	KNN	LGBM	ET	MLP
$\overline{\text{PCC}}$ (mm)	Annual	0.2423	0.2685	0.3581	0.4870	0.5255	0.3255
	Monsoon	0.2777	0.3030	0.4097	0.5353	0.5710	0.4159
	Nonmonsoon	0.1994	0.2247	0.3035	0.4312	0.4722	0.2287
$\overline{\text{MAE}}$ (mm)	Annual	3.1159	2.1937	1.7832	1.5592	1.5273	1.9455
	Monsoon	5.0978	3.5507	3.0323	2.6106	2.5378	3.1936
	Nonmonsoon	1.0824	0.7980	0.5222	0.5030	0.5101	0.6844
$\overline{\text{RMSE}}$ (mm)	Annual	4.4692	3.3399	3.3437	2.7901	2.7051	3.3021
	Monsoon	6.9803	5.2766	5.3459	4.5667	4.3961	5.2432
	Nonmonsoon	1.9133	1.3826	1.3301	1.0178	1.0170	1.3497
$\overline{\text{PCCSS}}$	Annual	—	0.108	0.478	1.010	1.169	0.343
	Monsoon	—	0.091	0.475	0.928	1.056	0.498
	Nonmonsoon	—	0.127	0.522	1.162	1.368	0.147
$\overline{\text{MAESS}}$	Annual	—	0.296	0.428	0.500	0.510	0.376
	Monsoon	—	0.303	0.405	0.488	0.502	0.374
	Nonmonsoon	—	0.263	0.518	0.535	0.529	0.368
$\overline{\text{RMSESS}}$	Annual	—	0.253	0.252	0.376	0.395	0.261
	Monsoon	—	0.244	0.234	0.346	0.370	0.249
	Nonmonsoon	—	0.277	0.305	0.468	0.468	0.295

We further evaluated the improvements of our postprocessing models relative to the raw forecast. Figure 9 show the spatial distributions of PCCSS, MAESS, and RMSESS, respectively. The postprocessing models were able to improve the precipitation forecasting skills at all grid cells, and the ET model displayed the best performance. The spatial distribution of the improvements in skill scores indicated that the PCCSS, MAESS and RMSESS values in the downstream region were all larger than those in the upper region, explaining that the postprocessing models could significantly improve the forecasting skill for extreme precipitation events.

d. Performance comparison between regional and local models

The difference between regional and local models is that a regional model uses data from all grid cells for modeling, while a local model only uses data from a specific grid cell. Therefore, a regional model can be applied to all grid cells in a basin, while a local model can only be applied to the specific grid cell that is modeled.

In the previous section, we focused on evaluating and discussing the forecasting skills of regional models. Are there large gaps between regional and local models, especially when ML methods are applied? Are there significant differences among different locations with different climate types? To explore these gaps and differences, in this subsection, we built additional local models using grid cells 5 and 39 as representatives.

The climatology obtained from the observations, raw forecast, and model postprocessing at grid cells 5 and 39 are shown in Figs. 10 and 11, respectively. The black lines in the figures represent the observations, which are consistent with those in Figs. 2c and 2d. The precipitation at grid cell 5 is 574.8 mm, while the precipitation at grid cell 39 is 1079.9 mm. The total precipitation and intra-annual distribution indicated significantly different climatic characteristics of the two grid cells. The pink lines represent the raw forecast, which displayed overestimation. The results obtained after the postprocessing of the precipitation data based on ML methods more closely match the observations. By comparing regional and local models, almost all regional models showed comparable performance with local models. Among the five methods, the ET model displayed the smallest difference between regional and local models. Even when comparing two grid cells with different climate types, there was almost no difference observed between the local and regional ET models.

5. Discussion and future works

a. Model selection and comparison

In this study, 21 machine learning models were selected and their effectiveness in precipitation forecasting postprocessing studies was confirmed. As shown in section 4, all the machine learning models outperformed the raw predictions. This also confirms that there is a large bias in the original prediction, which requires bias postprocessing. However, we would like to reiterate here that our selection of models is limited compared to the vast library of models. Several broad classes of different model architectures are included as much as possible, including linear and nonlinear, individual and ensemble models. And for each category, several variants that include penalty terms and regularization strategies are also selected to increase model diversity. Then, these models are compared and evaluated based on comparing the model structure without tuning parameters. This may lead to missing the best models with well-tuned parameters. However, our results are reasonable. The models with more advanced structural design have better performance without parameter tuning. Although different models could achieve comparable or even surpassing performance through parameter tuning, the authors believe that improvements in model structure are more

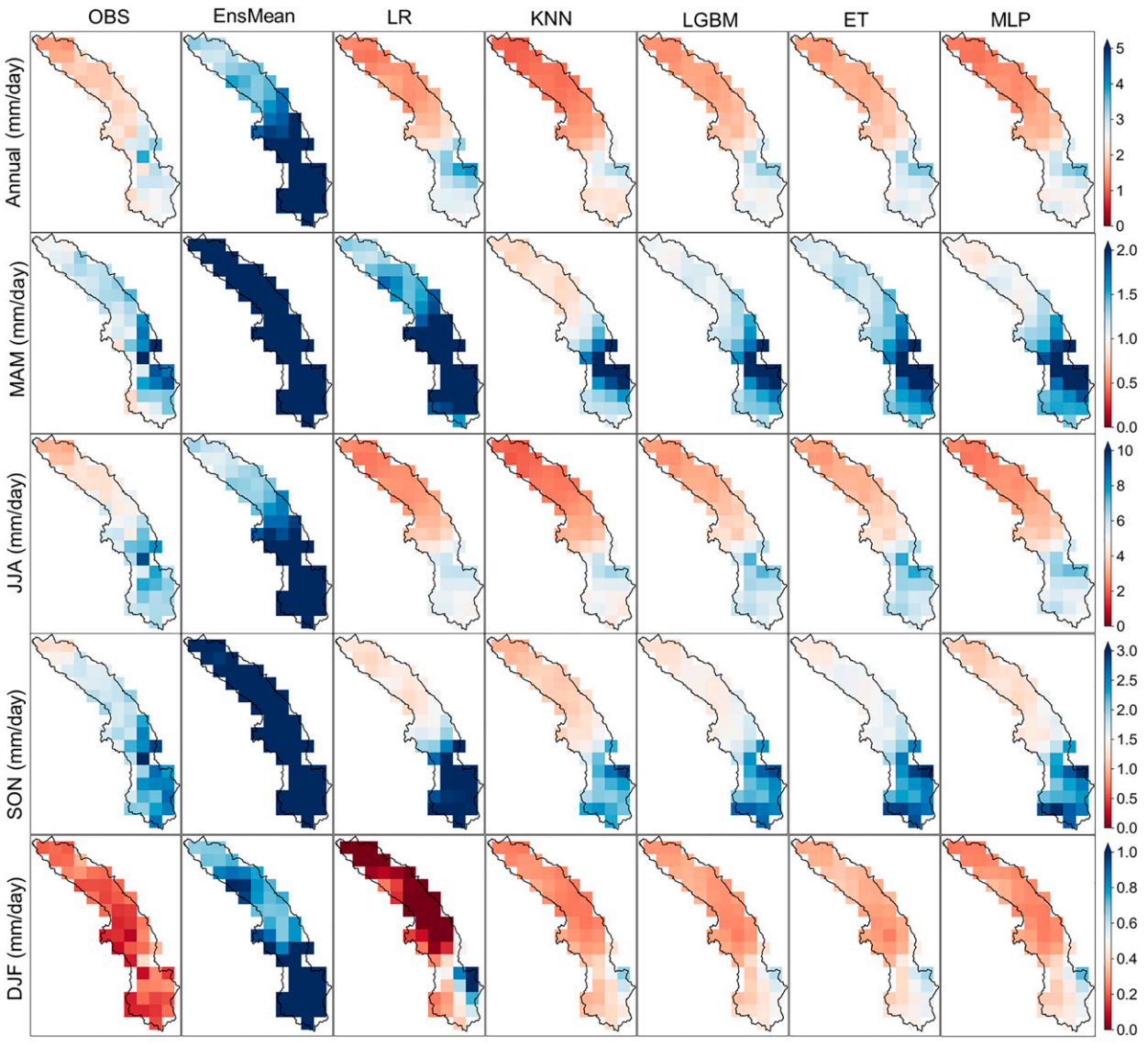

FIG. 8. Spatial distributions of precipitation forecasts from different models. The seasonal variations in the prediction are shown: MAM (March–May); JJA (June–August); SON (September–November); and DJF (December–February). OBS and EnsMean in the figure represent the observations and the means of the raw ensemble members, respectively.

enlightening for scientific research. Therefore, in the first stage only the model structures were compared, and models with different structural designs in each category were selected for the second stage of the experiment.

In the second stage, five models were selected as representatives based on a combination of model diversity, model performance, and training time. However, these choices are only representative and not optimal. For example, LGBM, as can be seen in Fig. 5, is inferior to CATB. This is because CATB is an improved version of LGBM based on category-based features. As the feature dimension increases, CATB better handles different types of features than LGBM. But LGBM is more widely used and in general it converges faster than CATB, so we have chosen it here as the representative of the boosting family. Among the selected machine learning models, ET and LGBM show surprising results in not only reducing the bias but also maintaining the bias at a low degree with different leading days. The difference between ET and LGBM is that the ET model is an ensemble learning method based on bagging, while LGBM is an ensemble learning method based on boosting. In our experiments, the ET model expresses a slightly better performance than the LGBM model, but the training time of LGBM is much less than that of the ET model, and the GPU version of the LGBM algorithm can shorten the training time even further. Thus, LGBM and its improved version (CATB) may be a more promising model in general.

In addition, we only compared the improvements to the raw forecast by the machine learning models in this study and did not build traditional statistical postprocessing models. Our model

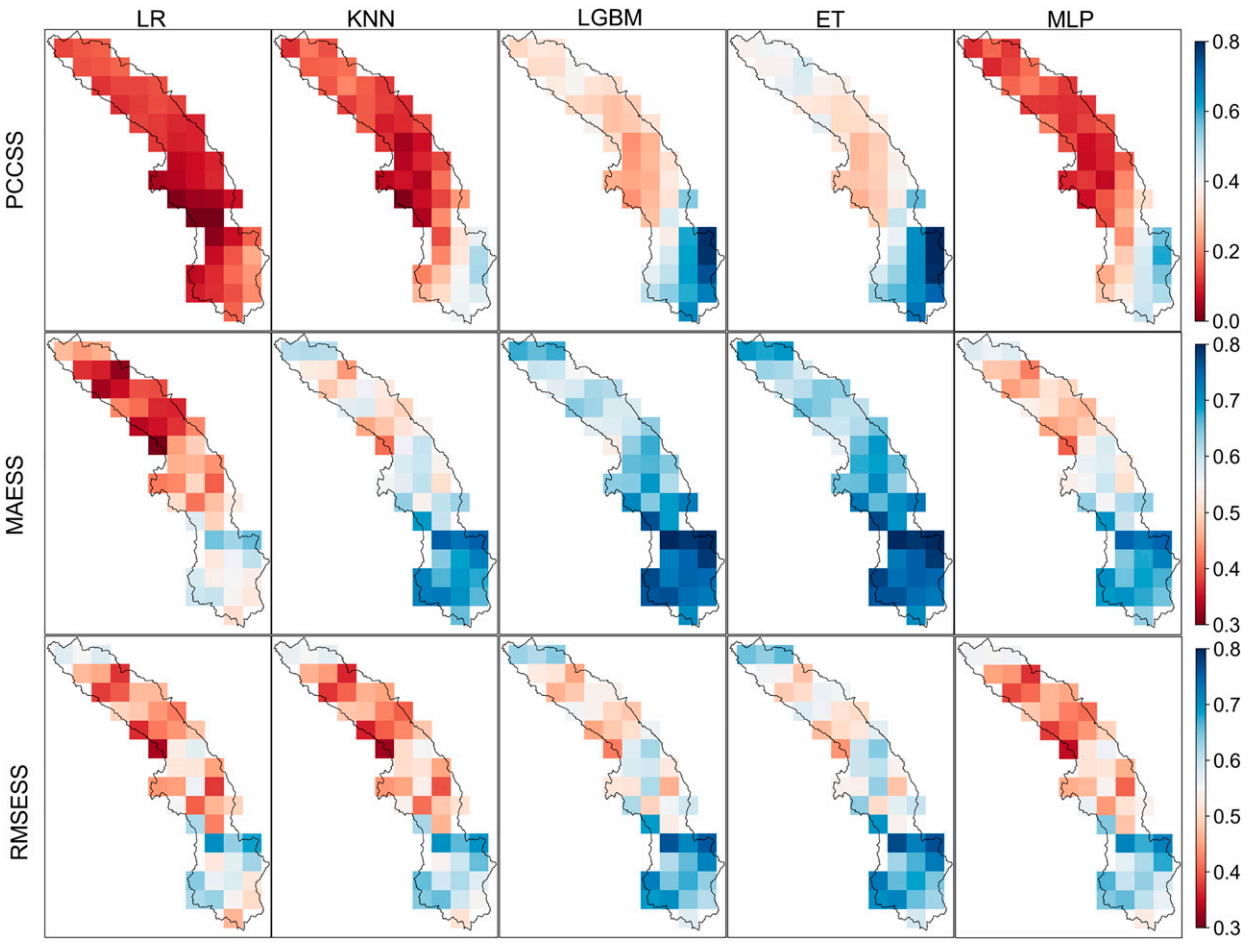

FIG. 9. Spatial distributions of precipitation forecast improvement skill scores from different models.

strategy and the selected predictors are different from those used in traditional methods. For example, a forward rolling time window (3 days) strategy is used in this study, while many studies select rolling time windows (15 days) before and after the event, which is a significant difference (Li et al. 2019). We selected a large number of predictors to build regional models, which is also different from the methods used for traditional statistical postprocessing models (Scheuerer and Hamill 2015). To determine whether the machine learning methods are better than the traditional methods, careful experiments should be set up in future studies for comparison (e.g., LGBM and CSGD).

In our experiments, we used a simple artificial neural network model (multilayer perceptron model, MLP) with only two hidden layers. We tuned the hyperparameters many times, including the learning rate, activation function, and the number of neurons. Our final model did not show a significantly better performance than those of other models. This may be caused by our insufficient training samples, as MLP tends to require a larger sample size compared to other ML models. Another possible reason is that we used a shallow neural network, and the shallower network layers underfit the nonlinear relationship between precipitation forecasts and observations. The major difference between the shallow neural network and deep neural networks is the number of layers. Advanced deep neural networks, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), contain deep hidden layers and even incorporate some feature extraction strategies, such as encoding–decoding. Such a group of deep learning models are very flexible and can perform many incredible tasks with careful tuning. In our research, we focused on comparisons between traditional machine learning models and therefore did not involve deep learning methods. Recent research suggested that deep learning methods may be able to provide more accurate precipitation forecasts or optimize postprocessing (Shi et al. 2017; Wu et al. 2020). Deep learning tends to require more computational resources. In the future, we can use more data and deeper neural networks to compare them with other traditional machine learning models (e.g., LGBM) with respect to their model performance and training time.

In addition to the boosting and bagging ensemble learning methods mentioned in the text, simple averaging is often used as the simplest form of ensemble learning (see e.g., Papacharalampous et al. 2019; Tyralis et al. 2019, 2020) and constitutes an alternative to hyperparameter optimization in the sense that they both improve predictive performance (see, e.g., Papacharalampous et al. 2019). We also made

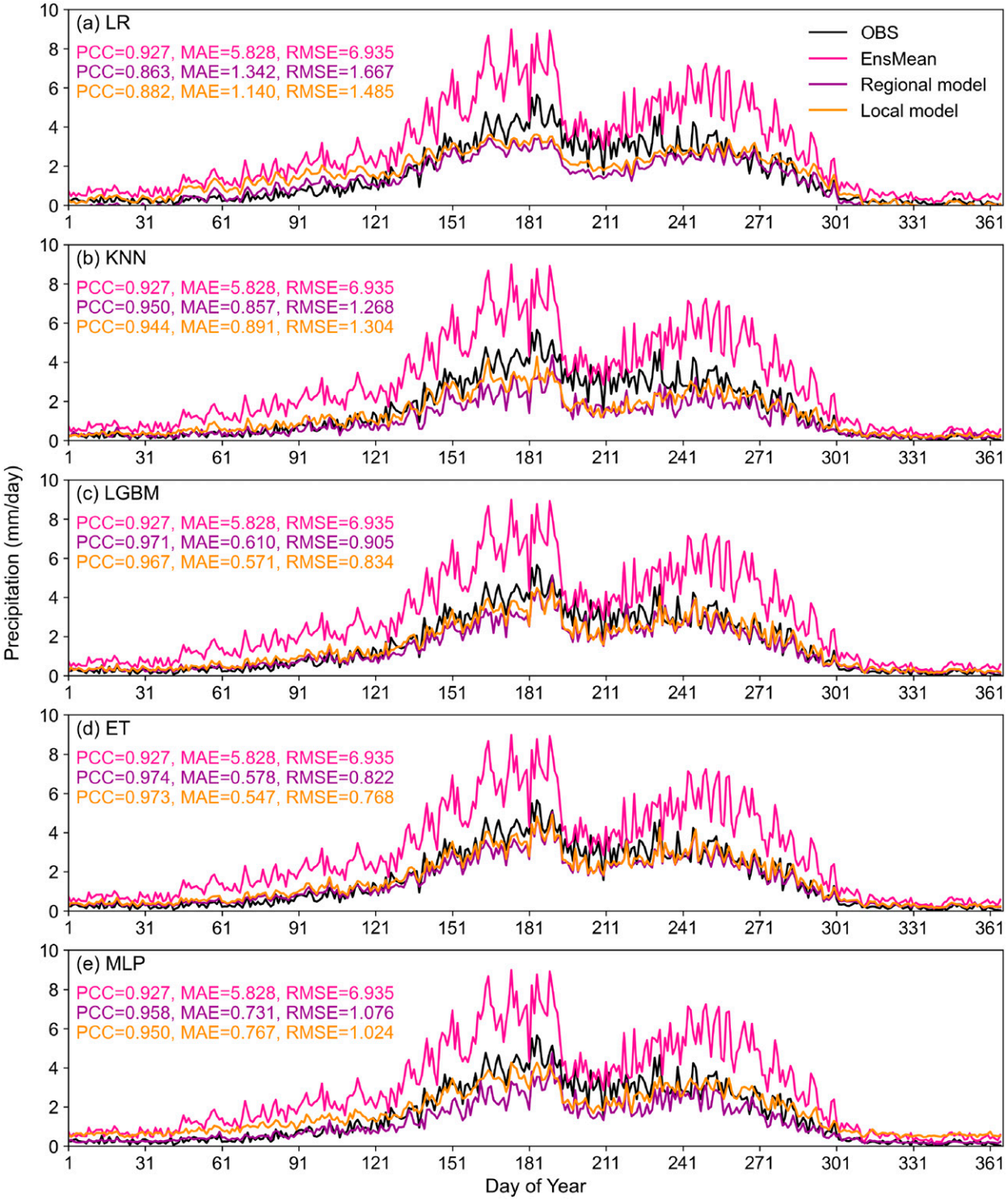

FIG. 10. The climatology obtained from the observations (OBS, black lines), raw forecast (EnsMean, pink lines), and regional (magenta lines) and local (orange lines) postprocessing models at grid cell 5.

similar attempts through three different model combinations (see Figs. S1–S3). They are the combination of all 21 models (combine1), the combination of four selected models (combine2), and the combination of the best two methods (combine3). This strategy does not seem to have achieved better results for combine1, because the performance of our base models is uneven. Inferior models applying equal weight averaging may affect the overall performance. But combination 2 and combination 3 seem

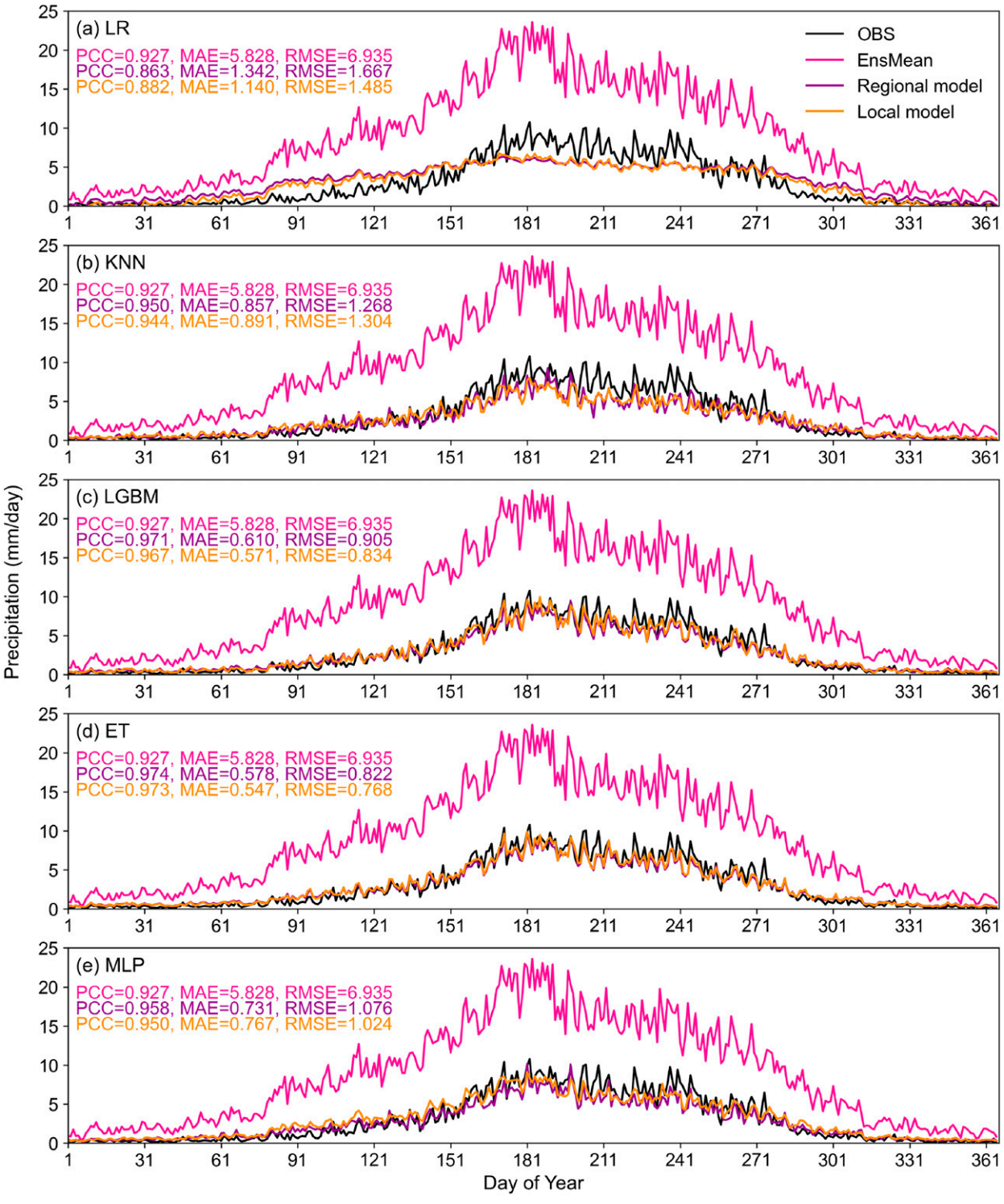

FIG. 11. The climatology obtained from the observations (OBS, black lines), raw forecast (EnsMean, pink lines), and regional (magenta lines) and local (orange lines) postprocessing models at grid cell 39.

to reach a comparable level, which confirms that strong combinations can better improve model performance. In this study, the good performance of the ensemble model based on bagging or boosting (e.g., RF and LGBM) shows that a weight averaging method are recommended to solve the problem of multimodel (or multimembers) precipitation postprocessing.

In summary, ML models are essentially derived from statistical theory, but they go beyond traditional statistical

methods by being able to handle high-dimensional features and by being applicable to big data. With the development of NWPs model and computer technology, the previous challenges of sample size and computational resources no longer exist, so ML models are potential as alternatives to traditional methods (Boukabara et al. 2019; Vannitsem et al. 2021).

b. Feature selection techniques

Predictors can be selected from various aspects, such as the element field, space and time. Different combinations of features are selected and compared in this study (Fig. 5). Thus, some preliminary variable importance can be derived: 1) the forward rolling time window is the most important variable; 2) ensemble members is essential to increase the dispersion; and 3) regional model based on neighboring data within the basin reduces modeling complexity.

There are three types of predictors in terms of physical elements. The first is based on the direct outputs of NWPs, such as precipitation predictions and variables that are closely related to precipitation (e.g., temperature and wind speed, see Bremnes 2004; Peng et al. 2014). The second category is the diagnostic physical elements that are calculated from basic factors, such as the water vapor flux (Darby et al. 2019). For direct and extended variables, the ensemble mean and standard deviation can also be used to express the mean and dispersion of ensemble members (Rasp and Lerch 2018). The third category includes geographically related attributes, such as latitude, longitude, and elevation information (Kratzert et al. 2019). However, in this study, we only selected the precipitation ensemble forecasts as predictors without considering the additional auxiliary variables because that a large number of computing resources were spent on the model comparison. In the future study, more covariables could be used as predictors to build a more sophisticated model.

Similarly, as for spatial features, all data within the watershed are fed into the model as spatial information to reduce the modeling effort. However, as the scale of the study area increases, this strategy is not desirable. In most of the previous studies, the strategy for the selection of spatial predictors was to build a postprocessing model for each grid cell or site separately to ensure local accuracy (Li et al. 2019). This approach is very complex and cumbersome for multiple stations with multiple forecasting periods. In recent years, considering precipitation center shift and regional similarity, several studies have also introduced a hyperparameter radius (Scheuerer 2014; Scheuerer and Hamill 2015). That is, when a postprocessing model is built for a specific grid cell, the forecasts within the radius are selected as predictors. However, due to differences in topography, climate type, and study area, the hyperparameter radius varies and is usually selected by adaptive algorithms.

Another difference from the traditional method is the selection of time-dimensional features. A forward rolling time window strategy is used in this study. This is different from the traditional rolling time window, which tends to select several periods before and after the forecast day as predictors (Hamill et al. 2008). In fact, the traditional method selects a larger time window because it considers regional climatology information to increase the sample size. However, the results of the correlation skill analysis (Fig. 3c) confirm that the useful information of forecasts with larger time windows is limited. A shocking finding is $f(k, k-1)$ in forward rolling time window is of a higher correlation with $p(k)$ comparing to $f(k, k)$. A possible hypothesis is that we are comparing the correlation skill between ensemble mean and observations, and that $f(k, k-1)$ has a higher dispersion (EnsStd) relative to the ensemble members of $f(k, k)$, thus better describing the uncertainty of the weather state. However, whether this hypothesis holds for other study areas needs to be confirmed by more cases. If so, the features obtained using this strategy would be more effective and would reduce the complexity of the postprocessing models.

Another issue is that an increase in the number of predictors may lead to redundancy of information and increase the computational burden. Therefore, in future studies, when we focus on one machine learning algorithm (e.g., LGBM), we can select more predictors to enhance the model performance. Meanwhile, because we did not select auxiliary predictors as features and the maximum feature dimension was 33 in this study, only the correlation skill technique was applied for feature selection, and the dimensionality reduction strategy was not designed. Although ML models have automatic feature extraction capabilities, manually designed feature selection and dimensionality reduction are necessary to filter for more physically meaningful predictors and reduce model complexity. In the future study, more strategies for dimensionality reduction (e.g., principal component analysis, t-SNE) should be included in the process of feature engineering to preserve the most important features as much as possible.

The last issue is the sample size. The sample size should be regarded as a hyperparameter that affects the model performance. An increase in the sample size may lead to improved model performance, but may also lead to overfitting. Due to the limitation of the number of experiments, we only used 10- and 20-yr data as the training sets in our research and did not choose more combinations. For different models in future studies, especially more complex deep learning models, the sample size is a noteworthy hyperparameter that needs to be treated carefully. Another contradiction is that although our sample size is large enough, the sample size of extreme precipitation events is still very small. This also leads to a certain shortcoming in our ability to handle extreme precipitation events. The best way to solve data imbalances through different approaches (e.g., data supplementation) is also an area worthy of further research.

6. Conclusions

Although the development of existing techniques has greatly improved the accuracy of weather forecasting, there are still uncertainties and bias in precipitation predictions. Bias correction is crucial for hydrometeorological ensemble forecasting and applications in hydrology-related fields. In

recent years, especially in the last five years, machine learning models have gained much popularity in Earth science disciplines, including hydrology. These data-driven approaches are essentially statistical learning methods, but they surpass traditional statistical methods in that more predictors can be selected as input feature vectors, and such black-box models do not require a priori information and can fit nonlinear relationships between input and output variables. All these advantages make machine learning algorithm become a better solution in the postprocessing of hydrometeorological ensemble forecasts. In this study, we try to compare postprocessing models for NWPs (e.g., GEFS) using machine learning approaches. The selected machine learning models include simple linear regression models and their extensions, artificial neural networks (e.g., MLPs), and state-of-the-art ensemble learning models (e.g., RF and LGBM). To fully compare the usability of these models, we selected the Yalong River basin as an example. We established nine different experiments for each machine learning algorithm, which resulted in a total of 189 combinations. The results show that our experimental design can adequately compare the advantages and disadvantages of the different models and select the best predictors for postprocessing modeling. Overall, the following conclusions can be drawn from our research:

1) The model structure determines their higher priority. Under same experimental conditions, the nonlinear, ensemble models always outperformed the linear models. Combining the factors of training time and model accuracy, LGBM is the best one among all selected models; when considering only model accuracy, the ET model is recommended for application.
2) The selection of forecast predictors plays a significant role in the improvement of the forecast accuracy of the postprocessing models. Among nine sets of experiments in our study, the best model performance was obtained using 11 ensemble members and a 3-day forward rolling time window. And the selection of the time window is the most important.
3) Machine learning models are able to learn local features from mixed samples of the whole basin with different climatic conditions. Regional models can reduce modeling complexity and improve efficiency in operational forecasting.

Acknowledgments. This work was jointly supported by the Natural Science Foundation of China (51879009), the Second Tibetan Plateau Scientific Expedition and Research Program (2019QZKK0405), the National Key Research and Development Program of China (2018YFE0196000), and the Interdisciplinary Research Foundation of Beijing Normal University for the First-Year Doctoral Students (BNUXKJC1905).

Data availability statement. The data used in this study are publicly available. The GEFS reforecast v2 dataset can be downloaded from the National Oceanic and Atmospheric Administration Physical Sciences Laboratory website (NOAA-PSL, https://psl.noaa.gov/forecasts/reforecast2/download.html). The CMA 0.5° daily precipitation dataset can be downloaded from the National Meteorological Science Data Center website (http://data.cma.cn/data/cdcdetail/dataCode/SURF_CLI_CHN_PRE_DAY_GRID_0.5.html).

APPENDIX

Software and Machine Learning

a. Used software

We conducted all experiments in this study with Python programming language including following packages: xarray (http://xarray.pydata.org/), numpy (https://numpy.org) and pandas (https://pandas.pydata.org/) for data preprocessing; scikit-learn (https://scikit-learn.org), pycaret (www.pycaret.org), LGBM (https://github.com/microsoft/LightGBM), XGB (https://xgboost.ai/), and CATB (https://catboost.ai/) for machine learning modeling; matplotlib (https://matplotlib.org/) and cartopy (https://scitools.org.uk/cartopy) for visualizations.

b. Machine learning basics

A theoretical linear model and a trained model can be expressed as

$$y = wX + \varepsilon, \tag{A1}$$

$$\hat{y}(w, x) = wX = w_0 + w_1 x_1 + \ \ldots \ + w_p x_p + \varepsilon, \tag{A2}$$

where y is the target variable; $\hat{y}$ is the fitted target variable, X is the input variable; w is the weight, ε is the error term.

We fit the above function by minimizing the loss function and solving the parameters

$$\hat{w} = \arg\min_{w} L(y, \hat{y}), \tag{A3}$$

where $\hat{w}$ is estimated value w; $L(y, \hat{y})$ is the loss function.

Original least squares linear regression:

$$L(y, \hat{y}) = \|wX - y\|_2^2. \tag{A4}$$

L1 regularization in lasso regression:

$$L(y, \hat{y}) = \frac{1}{2n}\|wX - y\|_2^2 + \alpha\|w\|_1. \tag{A5}$$

L2 regularization in ridge regression:

$$L(y, \hat{y}) = \|wX - y\|_2^2 + \alpha\|w\|_2^2. \tag{A6}$$

Combined L1 and L2 regularization in elastic net

$$L(y, \hat{y}) = \frac{1}{2n_{\text{samples}}}\|\| + \alpha\rho\|w\|_1 + \frac{\alpha(1-\rho)}{2}\|w\|_2^2, \tag{A7}$$

where α and ρ are regularization parameters.

Linear loss in Huber regression:

$$L(y,\hat{y}) = \sum_{i=1}^{n}\left[\sigma + H_{\epsilon}\left(\frac{X_i w - y_i}{\sigma}\right)\sigma\right] + \alpha\|w\|_2^2, \tag{A8}$$

$$H_{\epsilon}(z) = \begin{cases} z^2, \text{ if } |z| < \epsilon \\ 2\epsilon|z| - \epsilon^2, \text{ otherwise} \end{cases}, \tag{A9}$$

where σ and α are regularization parameters; ϵ is a threshold value using for select outliers.

Hinge loss function in PAR and SVM:

$$L(w,\varepsilon) = \max(0, |y - Xw| - \varepsilon), \tag{A10}$$

where ε is a threshold value using for select outliers.

REFERENCES

Ahmed, K., D. A. Sachindra, S. Shahid, Z. Iqbal, N. Nawaz, and N. Khan, 2020: Multi-model ensemble predictions of precipitation and temperature using machine learning algorithms. *Atmos. Res.*, **236**, 104806, https://doi.org/10.1016/j.atmosres.2019.104806.

Altman, N. S., 1992: An introduction to kernel and nearest-neighbor nonparametric regression. *Amer. Stat.*, **46**, 175–185, https://doi.org/10.2307/2685209.

Bishop, C. M., 2006: *Pattern Recognition and Machine Learning*. Springer, 738 pp.

Boukabara, S., V. Krasnopolsky, J. Q. Stewart, E. S. Maddy, N. Shahroudi, and R. N. Hoffman, 2019: Leveraging modern artificial intelligence for remote sensing and NWP: Benefits and challenges. *Bull. Amer. Meteor. Soc.*, **100**, ES473–ES491, https://doi.org/10.1175/BAMS-D-18-0324.1.

Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

Bremnes, J. B., 2004: Probabilistic forecasts of precipitation in terms of quantiles using NWP model output. *Mon. Wea. Rev.*, **132**, 338–347, https://doi.org/10.1175/1520-0493(2004)132<0338:PFOPIT>2.0.CO;2.

Chen, T., T. He, M. Benesty, V. Khotilovich, and Y. Tang, 2015: Xgboost: Extreme Gradient Boosting, version 0.4-2 1-4. R package, https://cran.r-project.org/web/packages/xgboost/index.html.

Crammer, K., O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, and M. K. Warmuth, 2006: Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, **7**, 551–585.

Darby, L. S., A. B. White, D. J. Gottas, and T. Coleman, 2019: An evaluation of integrated water vapor, wind, and precipitation forecasts using water vapor flux observations in the western United States. *Wea. Forecasting*, **34**, 1867–1888, https://doi.org/10.1175/WAF-D-18-0159.1.

Demargne, J., and Coauthors, 2014: The science of NOAA's operational hydrologic ensemble forecast service. *Bull. Amer. Meteor. Soc.*, **95**, 79–98, https://doi.org/10.1175/BAMS-D-12-00081.1.

Diez-Sierra, J., and M. Del Jesus, 2020: Long-term rainfall prediction using atmospheric synoptic patterns in semi-arid climates with statistical and machine learning methods. *J. Hydrol.*, **586**, 124789, https://doi.org/10.1016/j.jhydrol.2020.124789.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani, 2004: Least angle regression. *Annu. Stat.*, **32**, 407–499, https://doi.org/10.1214/009053604000000067.

Fischler, M. A., and R. C. Bolles, 1981: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, **24**, 381–395, https://doi.org/10.1145/358669.358692.

Freund, Y., and R. E. Schapire, 1996: Experiments with a new boosting algorithm. *ICML'96: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., 148–156, https://dl.acm.org/doi/abs/10.5555/3091696.3091715.

Friedman, J. H., 2001: Greedy function approximation: A gradient boosting machine. *Annu. Stat.*, **29**, 1189–1232, https://doi.org/10.1214/aos/1013203451.

Fritsch, J. M., and R. E. Carbone, 2004: Improving quantitative precipitation forecasts in the warm season: A USWRP research and development strategy. *Bull. Amer. Meteor. Soc.*, **85**, 955–966, https://doi.org/10.1175/BAMS-85-7-955.

Geurts, P., D. Ernst, and L. Wehenkel, 2006: Extremely randomized trees. *Mach. Learn.*, **63**, 3–42, https://doi.org/10.1007/s10994-006-6226-1.

Ghaith, M., A. Siam, Z. Li, and W. El-Dakhakhni, 2020: Hybrid hydrological data-driven approach for daily streamflow forecasting. *J. Hydrol. Eng.*, **25**, 04019063, https://doi.org/10.1061/(ASCE)HE.1943-5584.0001866.

Guan, H., B. Cui, and Y. Zhu, 2015: Improvement of statistical postprocessing using GEFS reforecast information. *Wea. Forecasting*, **30**, 841–854, https://doi.org/10.1175/WAF-D-14-00126.1.

Hamill, T. M., and J. S. Whitaker, 2006: Probabilistic quantitative precipitation forecasts based on reforecast analogs: Theory and application. *Mon. Wea. Rev.*, **134**, 3209–3229, https://doi.org/10.1175/MWR3237.1.

——, R. Hagedorn, and J. S. Whitaker, 2008: Probabilistic forecast calibration using ECMWF and GFS ensemble reforecasts. Part II: Precipitation. *Mon. Wea. Rev.*, **136**, 2620–2632, https://doi.org/10.1175/2007MWR2411.1.

——, G. T. Bates, J. S. Whitaker, D. R. Murray, and W. Lapenta, 2013: NOAA's second-generation global medium-range ensemble reforecast dataset. *Bull. Amer. Meteor. Soc.*, **94**, 1553–1565, https://doi.org/10.1175/BAMS-D-12-00014.1.

Hoerl, A. E., and R. W. Kennard, 1970: Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67, https://doi.org/10.1080/00401706.1970.10488634.

Huber, P. J., 2004: Robust statistics. *International Encyclopedia of Statistical Science*, M. Lovric, Ed., Springer, https://doi.org/10.1007/978-3-642-04898-2_594.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, 2017: LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, I. Guyon et al., Eds., Neural Information Processing Systems Foundation, 3146–3154.

Koh, K., S. J. Kim, and S. Boyd, 2007: An interior-point method for large-scale L1-regularized logistic regression. *J. Mach. Learn. Res.*, **8**, 1519–1555.

Kratzert, F., D. Klotz, M. Herrnegger, A. K. Sampson, S. Hochreiter, and G. S. Nearing, 2019: Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resour. Res.*, **55**, 11 344–11 354, https://doi.org/10.1029/2019WR026065.

Li, W., Q. Duan, C. Miao, A. Ye, W. Gong, and Z. Di, 2017: A review on statistical postprocessing methods for hydrometeorological ensemble forecasting. *Wiley Interdiscip. Rev.: Water*, **4**, e1246, https://doi.org/10.1002/wat2.1246.

——, ——, A. Ye, and C. Miao, 2019: An improved meta-Gaussian distribution model for post-processing of precipitation forecasts by censored maximum likelihood estimation. *J. Hydrol.*, **574**, 801–810, https://doi.org/10.1016/j.jhydrol.2019.04.073.

Lorenc, A. C., 1986: Analysis methods for numerical weather prediction. *Quart. J. Roy. Meteor. Soc.*, **112**, 1177–1194, https://doi.org/10.1002/qj.49711247414.

Lu, E., W. Zhao, X. Zou, D. Ye, C. Zhao, and Q. Zhang, 2017: Temporal-spatial monitoring of an extreme precipitation event: Determining simultaneously the time period it lasts and the geographic region it affects. *J. Climate*, **30**, 6123–6132, https://doi.org/10.1175/JCLI-D-17-0105.1.

Medina, H., D. Tian, F. R. Marin, and G. B. Chirico, 2019: Comparing GEFS, ECMWF, and postprocessing methods for ensemble precipitation forecasts over Brazil. *J. Hydrometeor.*, **20**, 773–790, https://doi.org/10.1175/JHM-D-18-0125.1.

Nourani, V., A. Baghanam, J. Adamowski, and O. Kisi, 2014: Applications of hybrid wavelet–Artificial intelligence models in hydrology: A review. *J. Hydrol.*, **514**, 358–377, https://doi.org/10.1016/j.jhydrol.2014.03.057.

Oppel, H., and S. Fischer, 2020: A new unsupervised learning method to assess clusters of temporal distribution of rainfall and their coherence with flood types. *Water Resour. Res.*, **56**, e2019WR026511, https://doi.org/10.1029/2019WR026511.

Owen, A. B., 2007: A robust hybrid of lasso and ridge regression. *Contemp. Math.*, **443**, 59–72, https://doi.org/10.1090/conm/443/08555.

Papacharalampous, G., H. Tyralis, A. Langousis, A. W. Jayawardena, B. Sivakumar, N. Mamassis, A. Montanari, and D. Koutsoyiannis, 2019: Probabilistic hydrological post-processing at scale: Why and how to apply machine-learning quantile regression algorithms. *Water*, **11**, 2126, https://doi.org/10.3390/w11102126.

Pati, Y. C., R. Rezaiifar, and P. S. Krishnaprasad, 1993: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *Proc. 27th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, IEEE, 40–44, https://doi.org/10.1109/ACSSC.1993.342465.

Peng, T., X. Zhi, Y. Ji, L. Ji, and Y. Tian, 2020: Prediction skill of extended range 2-m maximum air temperature probabilistic forecasts using machine learning post-processing methods. *Atmosphere*, **11**, 823, https://doi.org/10.3390/atmos11080823.

Peng, Z., Q. J. Wang, J. C. Bennett, P. Pokhrel, and Z. Wang, 2014: Seasonal precipitation forecasts over China using monthly large-scale oceanic-atmospheric indices. *J. Hydrol.*, **519**, 792–802, https://doi.org/10.1016/j.jhydrol.2014.08.012.

Piani, C., G. P. Weedon, M. Best, S. M. Gomes, P. Viterbo, S. Hagemann, and J. O. Haerter, 2010: Statistical bias correction of global simulated daily precipitation and temperature for the application of hydrological models. *J. Hydrol.*, **395**, 199–215, https://doi.org/10.1016/j.jhydrol.2010.10.024.

Prokhorenkova, L., G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, 2018: CatBoost: Unbiased boosting with categorical features. *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, S. Bengio et al., Eds., Curran Associates, 6639–6649, https://dl.acm.org/doi/10.5555/3327757.3327770.

Raghavendra, S., and P. C. Deka, 2014: Support vector machine applications in the field of hydrology: A review. *Appl. Soft Comput.*, **19**, 372–386, https://doi.org/10.1016/j.asoc.2014.02.002.

Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Mon. Wea. Rev.*, **146**, 3885–3900, https://doi.org/10.1175/MWR-D-18-0187.1.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986: Learning representations by back-propagating errors. *Nature*, **323**, 533–536, https://doi.org/10.1038/323533a0.

Sachindra, D. A., K. Ahmed, M. M. Rashid, S. Shahid, and B. J. C. Perera, 2018: Statistical downscaling of precipitation using machine learning techniques. *Atmos. Res.*, **212**, 240–258, https://doi.org/10.1016/j.atmosres.2018.05.022.

Scheuerer, M., 2014: Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Quart. J. Roy. Meteor. Soc.*, **140**, 1086–1096, https://doi.org/10.1002/qj.2183.

——, and T. M. Hamill, 2015: Statistical postprocessing of ensemble precipitation forecasts by fitting censored, shifted gamma distributions. *Mon. Wea. Rev.*, **143**, 4578–4596, https://doi.org/10.1175/MWR-D-15-0061.1.

Schölkopf, B., A. J. Smola, and F. Bach, 2002: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 70 pp.

Shi, X., Z. Gao, L. Lausen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, 2017: Deep learning for precipitation nowcasting: A benchmark and a new model. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, U. von Luxburg et al., Eds., Curran Associates, 5622–5632.

Taillardat, M., and O. Mestre, 2020: From research to applications Examples of operational ensemble post-processing in France using machine learning. *Nonlinear Processes Geophys.*, **27**, 329–347, https://doi.org/10.5194/npg-27-329-2020.

——, ——, M. Zamo, and P. Naveau, 2016: Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Mon. Wea. Rev.*, **144**, 2375–2393, https://doi.org/10.1175/MWR-D-15-0260.1.

Tyralis, H., G. Papacharalampous, A. Burnetas, and A. Langousis, 2019: Hydrological post-processing using stacked generalization of quantile regression algorithms: Large-scale application over CONUS. *J. Hydrol.*, **577**, 123957, https://doi.org/10.1016/j.jhydrol.2019.123957.

——, ——, and A. Langousis, 2020: Super ensemble learning for daily streamflow forecasting: Large-scale demonstration and comparison with multiple machine learning algorithms. *Neural Comput. Appl.*, **33**, 3053–3068, https://doi.org/10.1007/s00521-020-05172-3.

Vannitsem, S., and Coauthors, 2021: Statistical postprocessing for weather forecasts: Review, challenges, and avenues in a big data world. *Bull. Amer. Meteor. Soc.*, **102**, E681–E699, https://doi.org/10.1175/BAMS-D-19-0308.1.

Voyant, C., G. Notton, S. Kalogirou, M. L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, 2017: Machine learning methods for solar radiation forecasting: A review. *Renew. Energy*, **105**, 569–582, https://doi.org/10.1016/j.renene.2016.12.095.

Wang, Q., and Coauthors, 2020: Sequence-based statistical downscaling and its application to hydrologic simulations based on machine learning and big data. *J. Hydrol.*, **586**, 124875, https://doi.org/10.1016/j.jhydrol.2020.124875.

Wu, H., Q. Yang, J. Liu, and G. Wang, 2020: A spatiotemporal deep fusion model for merging satellite and gauge precipitation in China. *J. Hydrol.*, **584**, 124664, https://doi.org/10.1016/j.jhydrol.2020.124664.

Wu, X., Z. Wang, X. Zhou, C. Lai, W. Lin, and X. Chen, 2016: Observed changes in precipitation extremes across 11 basins in China during 1961-2013. *Int. J. Climatol.*, **36**, 2866–2885, https://doi.org/10.1002/joc.4524.

Xu, M., P. Watanachaturaporn, P. K. Varshney, and M. K. Arora, 2005: Decision tree regression for soft classification of remote

sensing data. *Remote Sens. Environ.*, **97**, 322–336, https://doi.org/10.1016/j.rse.2005.05.008.

Ye, A., Q. Duan, X. Yuan, E. F. Wood, and J. Schaake, 2014: Hydrologic post-processing of MOPEX streamflow simulations. *J. Hydrol.*, **508**, 147–156, https://doi.org/10.1016/j.jhydrol.2013.10.055.

——, ——, J. Schaake, J. Xu, X. Deng, Z. Di, C. Miao, and W. Gong, 2015: Post-processing of ensemble forecasts in low-flow period. *Hydrol. Processes*, **29**, 2438–2453, https://doi.org/10.1002/hyp.10374.

——, X. Deng, F. Ma, Q. Duan, Z. Zhou, and C. Du, 2017: Integrating weather and climate predictions for seamless hydrologic ensemble forecasting: A case study in the Yalong River basin. *J. Hydrol.*, **547**, 196–207, https://doi.org/10.1016/j.jhydrol.2017.01.053.

Yeo, I. K., and R. A. Johnson, 2000: A new family of power transformations to improve normality or symmetry. *Biometrika*, **87**, 954–959, https://doi.org/10.1093/biomet/87.4.954.

Yuan, X., E. F. Wood, and M. Liang, 2015: Integrating weather and climate prediction: Toward seamless hydrologic forecasting. *Geophys. Res. Lett.*, **41**, 5891–5896, https://doi.org/10.1002/2014GL061076.

Zhang, Z., L. Ye, H. Qin, Y. Liu, C. Wang, X. Yu, X. Yin, and J. Li, 2019: Wind speed prediction method using shared weight long short-term memory network and Gaussian process regression. *Appl. Energy*, **247**, 270–284, https://doi.org/10.1016/j.apenergy.2019.04.047.

Zhao, W. L., and Coauthors, 2019: Physics-constrained machine learning of evapotranspiration. *Geophys. Res. Lett.*, **46**, 14 496–14 507, https://doi.org/10.1029/2019GL085291.

Zhao, Y., J. Zhu, and Y. Xu, 2014: Establishment and assessment of the grid precipitation datasets in China for recent 50 years (in Chinese). *J. Meteor. Sci.*, **34**, 414–420, https://doi.org/10.3969/2013jms.0008.