Hydrology Research

Hydrology Research Vol 52 No 4, 927 doi: 10.2166/nh.2021.161

Hydrological probabilistic forecasting based on deep learning and Bayesian optimization algorithm

Haijun Bai[a], Guanjun Li[a], Changming Liu[a], Bin Li[a], Zhendong Zhang[b] and Hui Qin[b,*]
[a] Hunan Wushui Hydropower Development Co., Ltd, China Energy, Shaoyang, Hunan, China
[b] School of Civil and Hydraulic Engineering, Huazhong University of Science and Technology, Wuhan, Hubei, China
*Corresponding author. E-mail: hqin@hust.edu.cn

ABSTRACT

Obtaining accurate runoff prediction results and quantifying the uncertainty of the forecasting are critical to the planning and management of water resources. However, the strong randomness of runoff makes it difficult to predict. In this study, a hybrid model based on XGBoost (XGB) and Gaussian process regression (GPR) with Bayesian optimization algorithm (BOA) is proposed for runoff probabilistic forecasting. XGB is first used to obtain point prediction results, which can guarantee the accuracy of forecast. Then, GPR is constructed to obtain runoff probability prediction results. To make the model show better performance, the hyper-parameters of the model are optimized by BOA. Finally, the proposed hybrid model XGB-GPR-BOA is applied to four runoff prediction cases in the Yangtze River Basin, China and compared with eight state-of-the-art runoff prediction methods from three aspects: point prediction accuracy, interval prediction suitability and probability prediction comprehensive performance. The experimental results show that the proposed model can obtain high-precision point prediction, appropriate prediction interval and reliable probabilistic prediction results on the runoff prediction problems.

Key words: Bayesian optimization algorithm, Gaussian process regression, probabilistic forecasting, runoff, XGBoost

HIGHLIGHTS

- A novel deep learning method XGBoost is applied to predict runoff.
- A new hybrid model is proposed to quantify the uncertainty of prediction.
- The maximum information coefficient is used to select feature inputs.
- Bayesian optimization algorithm is used to optimize the hyper-parameters of the model.

1. INTRODUCTION

With the increasing shortage of energy, hydropower has received worldwide attention as a clean renewable energy (Liu *et al.* 2018a). However, flood disasters can pose risks to the safety of the reservoir and also cause economic benefits (Liu *et al.* 2018b). Therefore, hydrological probabilistic forecasting is very important for the operation and management of the reservoir, which contains two aspects: accurately predicting runoff and quantifying the uncertainty of the forecasting.

Runoff prediction methods are usually divided into two categories: process-driven method and data-driven method (Wen *et al.* 2019). The process-driven model is based on the hydrological concept and focuses on the description of the physical mechanism of runoff yield and concentration, such as the Xin'anjiang hydrological model (Fang *et al.* 2017) and numerical weather prediction (Wu & Lin 2017). These models have high prediction accuracy and are highly interpretable, but their data collection is difficult and the solution is time-consuming (Wu & Lin 2017). The data-driven model evolves runoff by mining the information contained in time series (Behera *et al.* 2006). The time-series model is a commonly used method for runoff prediction, mainly including auto-regressive model (AR), moving average model (MA), auto-regressive moving average model (ARMA) and their variants (Papacharalampous *et al.* 2018). These models are based on data stationarity assumptions, so their prediction accuracy is limited because of the strong nonlinearity of runoff (Mauricio 1995). To deal with the nonlinearity, many machine learning methods are used to predict runoff. In data-driven methods, many machine learning methods are used to predict runoff. A hybrid framework based on support vector regression (SVR) and series decomposition is proposed for monthly runoff forecasting (Luo *et al.* 2019). Artificial neural network (ANN) is used to characterize the nonlinearity of runoff and predict it, combining with empirical mode decomposition to improve the forecast performance (Tan *et al.* 2018). With the rapid development of deep learning methods in recent years, they are gradually being used to predict runoff or other time series, such as long short-term memory (LSTM) network (Zhang *et al.* 2019a) and convolutional neural network (CNN) (Le Callet *et al.* 2006). XGBoost (XGB; Chen & Guestrin 2016) is a novel deep learning method that is popular in various data mining competitions and wins competitions, such as Kaggle and KDDCup. However, this excellent model has not been used for hydrological forecasting. XGB belongs to the family of gradient boosting tree model. The model well known in their family has gradient boosting decision tree (GBDT) (Rao *et al.* 2019), classification and regression tree (CART) (Yang *et al.* 2016). As the latest variant of this family, the prediction performance of XGB is better than GBDT and CART. Another new variant light gradient boosting machine (LightGBM, LGB) (Deng *et al.* 2018), its training time is lower than XGB but the prediction accuracy is still not as good as XGB. Therefore, XGB is applied to predict runoff in this paper, which can ensure the prediction accuracy.

All of the above methods are deterministic prediction models, which cannot quantify forecast uncertainty. Constructing the upper and lower bounds corresponding to a certain confidence level is one of the approaches, called interval prediction (Khosravi *et al.* 2011). More comprehensively, the Bayesian method can quantify uncertainty by obtaining the probability density function of predictions (Wang *et al.* 2009). Gaussian process regression (GPR) is employed for monthly streamflow forecasting, which has the advantage of high reliability (Sun *et al.* 2014). Quantile regression (QR) is also the type of runoff probability prediction method, which can obtain the conditional quantile of prediction (Zhang *et al.* 2019b). The probability density function can also be obtained by further combining the kernel density estimation (KDE) method. In this study, the GPR model is used to quantify forecast uncertainty.

To make the forecasting model show better performance, the hyper-parameter optimization is carried out for each one. Commonly used hyper-parameter optimization algorithms are grid search algorithm (GSA) (Kong *et al.* 2017), random search algorithm (RSA) (Al-Muhammed & Abu Zitar 2018) and Bayesian optimization algorithm (BOA) (He *et al.* 2019). GSA is essentially an exhaustive method and its search efficiency is low. The loss function of RSA is prone to fluctuations, and the search process has no direction. BOA can estimate the distribution between the hyper-parameters and the loss so that the search process has a direction and the efficiency is high. Therefore, BOA is used to optimize the hyper-parameters of the hybrid model in this paper.

In this paper, a probabilistic forecasting hybrid model called XGB-GPR-BOA is proposed to predict runoff and quantify the uncertainty of prediction. The main contributions are outlined as follows:

1. A novel deep learning method XGB is applied to predict runoff, which can ensure the prediction accuracy.
2. A new hybrid model combined XGB and GPR is proposed to obtain runoff probabilistic prediction results.
3. To make the hybrid model show better performance, BOA is used to optimize hyper-parameters.
4. The proposed hybrid model XGB-GPR-BOA is applied to four runoff prediction cases in the Yangtze River Basin, China and compared with eight state-of-the-art runoff prediction methods. The rest of the paper is organized as follows: In Section 2, the implementation details of the hybrid model XGB-GPR-BOA are introduced. In Section 3, the evaluation metrics of prediction performance are explained. In Section 4, the proposed model is applied to solve the practical prediction problem and performed the comparison experiments. In Section 5, the work of this paper is summarized and the conclusions are given.

2. METHODS

2.1. XGBoost

XGB is a novel gradient tree boosting algorithm, which has the advantages of strong ability to overcome overfitting and high prediction accuracy (Chen & Guestrin 2016). Suppose $D = [X_i, Y_i]$ $(i = 1, 2, \cdots, n)$ is a given training set with n examples and m features, where $X_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,m}]$ are features and $Y_i = [y_i]$ is label.

2.1.1. Model structure of XGB

XGB uses K additive functions to predict the label:

$$\hat{y}_i = \phi(X_i) = \sum_{k=1}^{K} f_k(X_i) \tag{1}$$

where $\hat{y}_i$ is the prediction value and $f_k(\cdot)$ is the kth regression tree weak model. K weak models are integrated into a strong model $\phi(\cdot)$.

The tree ensemble model is shown in Figure 1. K different regression trees divide the samples into different leaves according to their own split conditions. The final prediction for a given example is the sum of predictions from each tree. It can be seen from the diagram that the prediction of a tree is determined by the structure of the tree and the weight of the leaves. The structure of the tree is actually these leaves, represented by the symbol q. The weight of the leaves is represented by the symbol w.

2.1.2. Model training of XGB

Training an XGB model is to solve the number of regression tree (K), the structure of each regression tree (q), and the weight of each leaf (w). The purpose of training is to minimize the following regularized loss function:

$$L(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

where $l(\hat{y}_i, y_i)$ is a loss function that measures the difference between the prediction $\hat{y}_i$ and the label y_i. $\Omega(f_k)$ is a penalty that measures the complexity of the model, which helps to avoid overfitting, such as $L1$ regularization and $L2$ regularization (Chen & Guestrin 2016).

2.1.2.1. The number of regression tree (K). Formally, $\hat{y}_i^{(k-1)}$ represents the prediction of the ith instance at the $(k-1)$th iteration. Next, we need to add f_k to minimize the following objective:

$$L^{(k)}(f_k) = \sum_{i=1}^{n} l(\hat{y}_i^{(k-1)} + f_k(X_i), y_i) + \Omega(f_k) \tag{3}$$

The new regression tree f_k is greedily added if it can significantly improve the model according to Equation (2). The final number of regression tree is the K value.

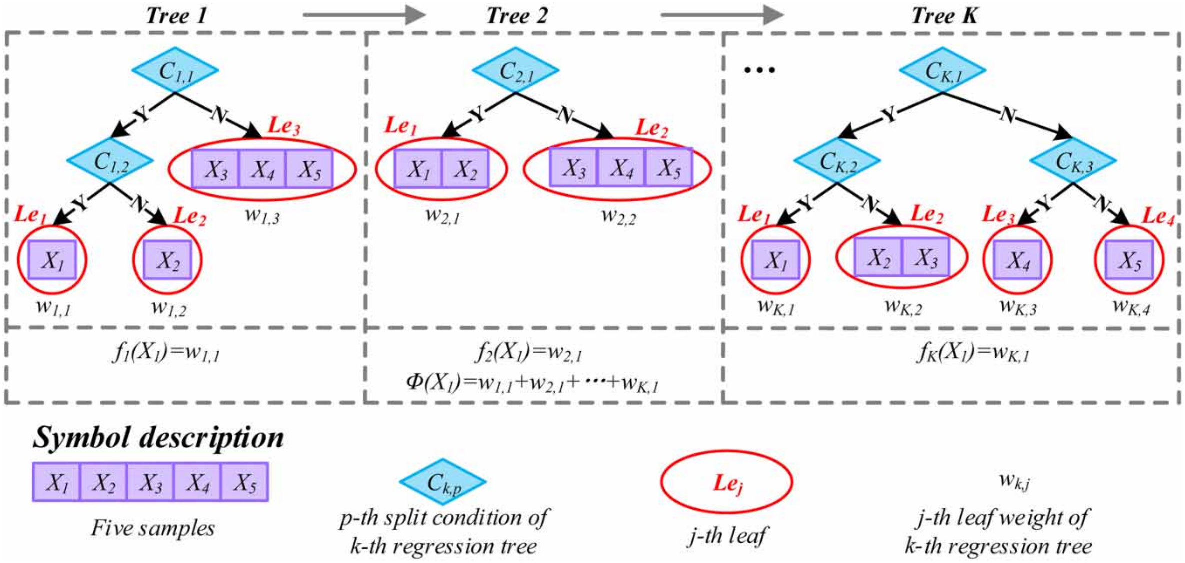


Figure 1 | Tree ensemble model.

2.1.2.2. Leaf weights (w_k). Each regression tree (f_k) corresponds to an independent tree structure (q_k) and leaf weights (w_k). Solving f_k means solving q_k and w_k. The objective $L^{(k)}(\phi)$ is approximated by second-order Taylor expansion as follows:

$$L^{(k)}(f_k) \simeq \sum_{i=1}^{n} [l(\hat{y}_i^{(k-1)}, y_i) + g_{k,i}\, f_k(X_i) + \frac{1}{2} h_{k,i}\, f_k^2(X_i)] + \Omega(f_k) \tag{4}$$

where $g_{k,i} = \partial l(\hat{y}_i^{(k-1)}, y_i)/\partial \hat{y}_i^{(k-1)}$ and $h_{k,i} = \partial^2 l(\hat{y}_i^{(k-1)}, y_i)/\partial \hat{y}_i^{(k-1)}$ are first- and second-order gradient statistics on the loss function, respectively. The second-order gradient statistic on the loss function is used in XGB, which is the biggest difference from other gradient tree boosting models and it is also the reason for its high prediction accuracy.

Removing the constant terms, the optimization objective can be simplified as follows:

$$\tilde{L}^{(k)}(f_k) = \sum_{i=1}^{n} [g_{k,i} f_k(X_i) + \frac{1}{2} h_{k,i} f_k^2(X_i)] + \Omega(f_k) \tag{5}$$

Define $I_{k,j} = \{i | q_k(X_i) = j\}$ as the instance set of jth leaf in the kth regression tree. The objective can be rewritten as follows. This process will be easier to understand in conjunction with Figure 1.

$$\begin{aligned} \tilde{L}^{(k)}(f_k) &= \sum_{i=1}^{n} [g_{k,i} f_k(X_i) + \frac{1}{2} h_{k,i} f_k^2(X_i)] + \Omega(f_k) \\ &= \sum_{j=1}^{T} \Bigg[\Bigg(\sum_{i\in I_{k,j}} g_{k,i})w_{k,j} + \frac{1}{2}\Bigg(\sum_{i\in I_{k,j}} h_{k,i}\Bigg) w_{k,j}^2 + \tilde{\Omega}(w_{k,j}\Bigg)\Bigg] \end{aligned} \tag{6}$$

where T is the number of leaves. $\tilde{\Omega}(w_{k,j})$ is a function describing the complexity of the jth leaf model of the kth regression tree. $\tilde{\Omega}(w_{k,j}) = 1/2\lambda w_{k,j}^2$ and $\tilde{\Omega}(w_{k,j}) = 1/2\lambda w_{k,j}$ are commonly used formulas of $\tilde{\Omega}(w_{k,j})$, which corresponds to $L2$ regularization and $L1$ regularization, respectively. λ is the penalty coefficient. In the following derivation, $L2$ regularization is taken as an example, and the other form of $\tilde{\Omega}(w_{k,j})$ is similar.

Since $\tilde{L}^{(k)}$ is the sum of T independent quadratic functions, for a fixed structure q_k, let $\partial\tilde{L}^{(k)}/\partial w_{k,j} = 0$, the optimal weight $w_{k,j}^*$ and optimal objective $\tilde{L}^{(k)*}$ can be computed as follows:

$$w_{k,j}^* = -\frac{\sum\limits_{i\in I_{k,j}} g_{k,i}}{\sum\limits_{i\in I_{k,j}} h_{k,i} + \lambda} \tag{7}$$

$$\tilde{L}^{(k)*} = -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum\limits_{i\in I_{k,j}} g_{k,i}\right)^2}{\sum\limits_{i\in I_{k,j}} h_{k,i} + \lambda} \tag{8}$$

2.1.2.3. Tree structure (q_k). Solving the tree structure (q_k) is actually determining the split conditions (C_k) and the instance set (I_k) on the leaf. Branches are greedily increased by score gain. Assume that I_k^L and I_k^R are the instance sets of the left and right nodes after split and $I_k = I_k^L \cup I_k^R$. The score gain is computed as follows:

$$s = \frac{1}{2}\left[\frac{(\sum\limits_{i\in I_k^L} g_{k,i})^2}{\sum\limits_{i\in I_k^L} h_{k,i} + \lambda} + \frac{(\sum\limits_{i\in I_k^R} g_{k,i})^2}{\sum\limits_{i\in I_k^R} h_{k,i} + \lambda} - \frac{(\sum\limits_{i\in I_k} g_{k,i})^2}{\sum\limits_{i\in I_k} h_{k,i} + \lambda}\right] \tag{9}$$

where s is the score gain. The maximum of score gain is the point of split. If the maximum of score gain is less than zero, it means that the current node is not split.

2.2. Gaussian process regression

The point prediction results of XGB are taken as the feature inputs, and the real runoff results are used as the labels to construct the GPR (Sun *et al.* 2014).

A regression model with noisy is assumed as follows:

$$Y = f(X) + \xi \tag{10}$$

where Y is the label and X is the feature input. The noisy $\xi \sim N(0, \sigma_n^2)$.

Then the prior distribution of the label Y, the joint prior distribution of the label Y and the prediction $\hat{y}$ can be obtained as follows:

$$Y \sim N(0, K(X, X) + \sigma_n^2 I_n) \tag{11}$$

$$\begin{aligned} \begin{bmatrix} Y \\ \hat{y} \end{bmatrix} &\sim N\left(0, \quad \begin{bmatrix} K(X, X) + \sigma_n^2 I_n & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \\ &= N\left(0, \quad \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \end{aligned} \tag{12}$$

where $K(X, X) = (\kappa_{ij})$ is a symmetric positive definite covariance matrix, whose elements κ_{ij} measure the correlation between X_i and X_j through a kernel function κ. $K(X_*, X) = K(X, X_*)^T$ is the covariance matrix between the validation set X_* and training set X. $K(X_*, X_*)$ is the covariance matrix of the validation set itself. I_n is an n-dimensional unit matrix.

Through the knowledge of probability theory and linear algebra, the posterior distribution of the prediction $\hat{y}$ can be obtained as follows:

$$\hat{y}|Y \sim N(\bar{\hat{y}}, \hat{\sigma}_y^2) \tag{13}$$

$$\bar{\hat{y}} = K_* K^{-1} Y \tag{14}$$

$$\hat{\sigma}_y^2 = K_{**} - K_* K^{-1} K_*^T \tag{15}$$

Therefore, the point prediction result of runoff is $\bar{\hat{y}}$, and the interval prediction result corresponding to the 95% confidence level is $[\bar{\hat{y}} - 1.96\hat{\sigma}_y, \bar{\hat{y}} + 1.96\hat{\sigma}_y]$. The probability density function of ith predicted value is as follows:

$$p(\hat{y}_i) = \frac{1}{\sqrt{2\pi}\hat{\sigma}_{y,i}} \exp\left(-\frac{(\hat{y}_i - \bar{\hat{y}}_i)}{2\hat{\sigma}_{y,i}^2}\right) \tag{16}$$

2.3. Bayesian optimization algorithm

To make the model perform better, BOA is used to optimize hyper-parameters (He *et al.* 2019). The prediction framework flowchart of XGB-GPR-BOA is shown in Figure 2.

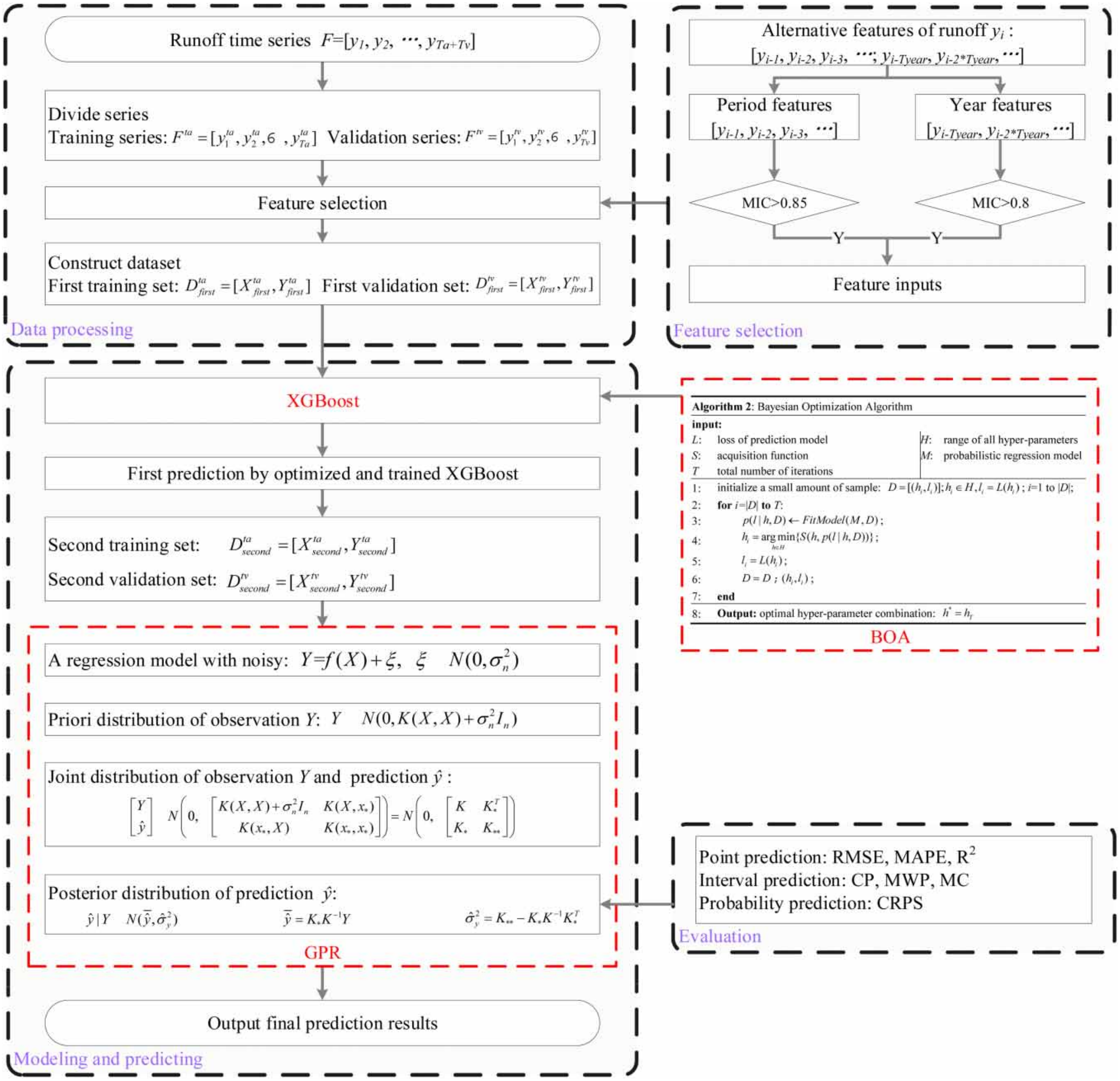
Taking minimizing the loss function as an example, the hyper-parameter optimization problem can be given as follows:

$$h^* = \underset{h \in H}{argmin}\{L(h)\} \tag{17}$$

where H is the range of all hyper-parameters. $L(h)$ is the loss of prediction model under the hyper-parameter combination h, and calculating it is a time-consuming step. h^* is the optimal hyper-parameter combination.

The optimization steps of the hyper-parameter are as follows:

Step 1: A small number of hyper-parameter combinations $[h_i]$ are randomly initialized in the definition domain H, and each combination h_i is input into the model (XGB-GPR) proposed in this paper. The corresponding loss function l_i is further calculated to construct an initial data set D.


Runoff time series F=[y_1, y_2, ⋯, y_{Ta+Tv}]
Divide series
Training series: $F^{ta}=[y_1^{ta},y_2^{ta},6\ \ ,y_{Ta}^{ta}]$ Validation series: $F^{tv}=[y_1^{tv},y_2^{tv},6\ \ ,y_{Tv}^{tv}]$
Feature selection
Construct dataset
First training set: $D_{first}^{ta}=[X_{first}^{ta},Y_{first}^{ta}]$ First validation set: $D_{first}^{tv}=[X_{first}^{tv},Y_{first}^{tv}]$
Data processing
Alternative features of runoff y_i :
[y_{i-1}, y_{i-2}, y_{i-3}, ⋯, $y_{i-Tyear}$, $y_{i-2*Tyear}$, ⋯]
Period features
[y_{i-1}, y_{i-2}, y_{i-3}, ⋯]
Year features
[$y_{i-Tyear}$, $y_{i-2*Tyear}$, ⋯]
MIC>0.85
MIC>0.8
Y
Y
Feature inputs
Feature selection
XGBoost
First prediction by optimized and trained XGBoost
Second training set: $D_{second}^{ta}=[X_{second}^{ta},Y_{second}^{ta}]$
Second validation set: $D_{second}^{tv}=[X_{second}^{tv},Y_{second}^{tv}]$
A regression model with noisy: $Y=f(X)+\xi,\ \ \xi\ \ \ N(0,\sigma_n^2)$
Priori distribution of observation Y: $Y\ \ \ N(0,K(X,X)+\sigma_n^2 I_n)$
Joint distribution of observation Y and prediction $\hat{y}$:
$\begin{bmatrix} Y \\ \hat{y} \end{bmatrix}\ \ N\left(0,\ \begin{bmatrix} K(X,X)+\sigma_n^2 I_n & K(X,x_*) \\ K(x_*,X) & K(x_*,x_*) \end{bmatrix}\right)=N\left(0,\ \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right)$
Posterior distribution of prediction $\hat{y}$:
$\hat{y}\,|\,Y\ \ \ N(\bar{\hat{y}},\hat{\sigma}_y^2)$
$\bar{\hat{y}}=K_*K^{-1}Y$
$\hat{\sigma}_y^2=K_{**}-K_*K^{-1}K_*^T$
GPR
Output final prediction results
Modeling and predicting
BOA
Point prediction: RMSE, MAPE, R^2
Interval prediction: CP, MWP, MC
Probability prediction: CRPS
Evaluation

```
Algorithm 2: Bayesian Optimization Algorithm
input:
L: loss of prediction model            H: range of all hyper-parameters
S: acquisition function                M: probabilistic regression model
T  total number of iterations
initialize a small amount of sample: $D=[(h_i,l_i)]; h_i \in H, l_i=L(h_i)$ ; i=1 to |D|;
for i=|D| to T:
    $p(l\,|\,h,D) \leftarrow FitModel(M,D)$ ;
    $h_i=\arg\min_{h\in H}\{S(h,p(l\,|\,h,D))\}$ ;
    $l_i=L(h_i)$ ;
    $D=D\ ;\ (h_i,l_i)$ ;
end
Output: optimal hyper-parameter combination: $h^*=h_T$
```

Figure 2 | The prediction framework flowchart of XGB-GPR-BOA.

Step 2: Training a probabilistic regression model M on the data set D, the probability distribution function $p(l|h, D)$ of the objective function (loss function) l is obtained by trained M. M is an existing probabilistic regression model, such as random forest and tree parzen estimators (He *et al.* 2019).

Step 3: Defining the acquisition function S, using the current probability distribution function $p(l|h, D)$ as a cheap surrogate for the expensive objective l, the new location h_i is obtained by minimizing S. Common forms of acquisition function are the probability of improvement, excepted improvement and entropy search (He *et al.* 2019).

Step 4: Calculate the loss function l_i of new location h_i, add it (h_i, l_i) into the data set D, repeat steps 2 and 3 until the total number of iterations is reached and output the final parameter combination h_T as the optimal parameter combination h^*.

3. EVALUATION METRICS

3.1. Evaluation metric of point prediction

To evaluate the accuracy of point prediction, root-mean-square error (RMSE) (Zhang *et al.* 2020), mean absolute percentage error (MAPE) (Zhang *et al.* 2019c) and coefficient of determination (R^2) are used to evaluate the deviation between the predictions and observations.

3.1.1. Root-mean-square error

The smaller the RMSE, the higher the point prediction accuracy. Its formula is as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T_\mathrm{v}}\sum_{i=1}^{T_\mathrm{v}}(\hat{y}_i - y_i)^2} \tag{18}$$

where $\hat{y}_i$ and y_i are the prediction and observation, respectively. T_v is the size of validation set.

3.1.2. Mean absolute percentage error

The smaller the MAPE, the higher the point prediction accuracy. Its formula is as follows:

$$\mathrm{MAPE} = \frac{1}{T_\mathrm{v}}\sum_{i=1}^{T_\mathrm{v}}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{19}$$

3.1.3. Coefficient of determination

The closer the value of R^2 is to 1, the higher the point prediction accuracy. Its formula is as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{T_\mathrm{v}}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{T_\mathrm{v}}(y_i - \bar{y}_i)^2} \tag{20}$$

where $\bar{y}_i$ is the mean of observations.

3.2. Evaluation metric of interval prediction

To evaluate the suitability of interval prediction, coverage probability (CP) (Li & Jin 2018) and mean width percentage (MWP) (Li & Jin 2018) are used in this paper.

3.2.1. Coverage probability

CP_α is defined as the probability that the observation falls within the prediction interval under the confidence level of α. Its formula is as follows:

$$CP_\alpha = \frac{c_\alpha}{T_\mathrm{v}} \times 100\% \tag{21}$$

where c_α is the number of samples whose observations fall within the prediction interval.

3.2.2. Mean width percentage

MWP_α is used to measure the prediction interval width, whose formula is as follows:

$$\mathrm{MWP}_\alpha = \frac{1}{T_\mathrm{v}}\sum_{i=1}^{T_\mathrm{v}}\frac{\mathrm{up}_i - \mathrm{down}_i}{y_i} \tag{22}$$

where up_i and down_i are the upper and lower limits of the prediction interval.

3.2.3. Suitability metric

The ideal prediction interval should have high CP_α and low MWP_α; therefore, the comprehensive metric of interval prediction is defined as MC_α. The smaller the MC_α, the more suitable the prediction interval. Its formula is as follows:

$$\mathrm{MC}_\alpha = \mathrm{MWP}_\alpha/\mathrm{CP}_\alpha \tag{23}$$

3.3. Evaluation metric of probability prediction

To evaluate the comprehensive performance of probability prediction, continuous ranked probability score (CRPS) is used in this paper (Hersbach 2000). The smaller the CRPS, the better the comprehensive performance of probability prediction. The formula of CRPS is as follows:

$$\mathrm{CRPS} = \frac{1}{T_{\mathrm{v}}}\sum_{i=1}^{T_{\mathrm{v}}}\int_{-\infty}^{+\infty}[F(\hat{y}_i) - H(\hat{y}_i - y_i)]^2 d\hat{y}_i \tag{24}$$

$$F(\hat{y}_i) = \int_{-\infty}^{\hat{y}_i} p(x)dx \tag{25}$$

$$H(\hat{y}_i - y_t) = \begin{cases} 0 & \hat{y}_i t < y_t \\ 1 & \hat{y}_i \geq y_t \end{cases} \tag{26}$$

where $p(\hat{y}_t)$ is the probability density function of $\hat{y}_t$ and $F(\hat{y}_t)$ is its cumulative distribution function. $H(\hat{y}_i - y_t)$ is the Heaviside function.

4. CASE INTRODUCTION

To verify the performance of the proposed model XGB-GPR-BOA, four datasets with different hydrologic stations and different lengths are gathered for experiments. Statistical information of four datasets is shown in Table 1. The data of these datasets are from Yichang Station, Gaochang Station or Binshan Station, which are located in the Yangtze River Basin of China. The study area is shown in Figure 3. In the table, *T*, *Ta* and *T*v represent the size of total dataset, training set and validation set, respectively.

Table 1 | Statistical information of four datasets

Datasets	Station	Time	*T*	T_a	T_v	Min.	Mean	Max.	SD
Unit		**1 period = 1 day**	**Period**			**m³/s**	**m³/s**	**m³/s**	
Dataset 1	Yichang	1 January 2000–31 December 2004	1,827	1,096	731	2,950	13,323	58,400	9,974
Dataset 2	Yichang	1 January 2007–31 December 2011	1,826	1,096	730	3,292	12,369	66,488	9,310
Dataset 3	Gaochang	1 January 2004–31 December 2010	2,557	1,461	1,096	515	2,470	15,900	1,908
Dataset 4	Binshan	1 January 2001–31 December 2007	2,556	1,461	1,095	1,180	4,695	22,100	3,856

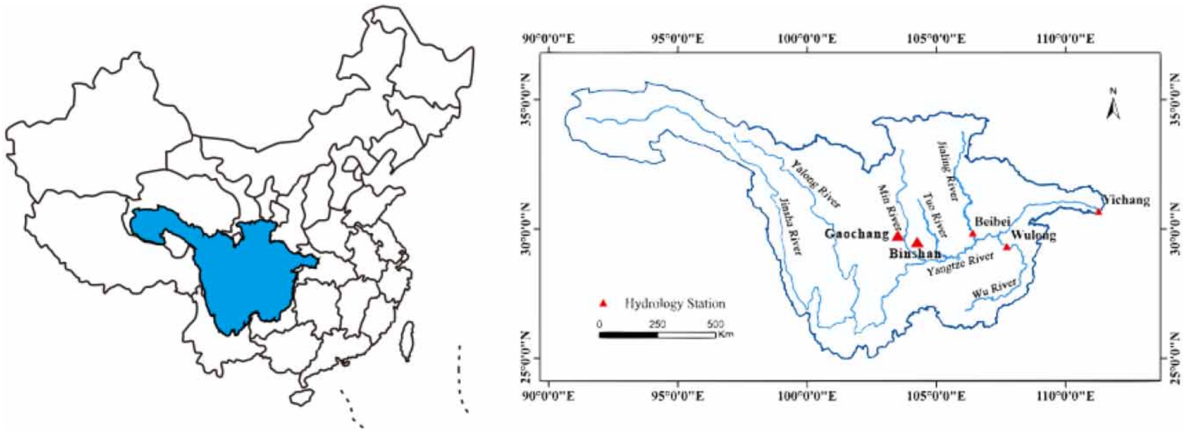


Figure 3 | The study area.

5. RESULTS AND DISCUSSION

5.1. Experimental design and parameter settings

To verify the performance of the proposed method, LGB (Deng *et al.* 2018), Gradient Boosting Regression Tree (GBR) (Rao *et al.* 2019), LSTM network (Zhang *et al.* 2019a), CNN (Le Callet *et al.* 2006), ANN (Tan *et al.* 2018), SVR (Luo *et al.* 2019), QR (Zhang *et al.* 2019b) and GPR (Sun *et al.* 2014) are compared with XGB. Using the framework proposed in this paper, combined with GPR, these models are further transformed to obtain the probabilistic forecasting model: XGB-GPR, LGB-GPR, GBR-GPR, LSTM-GPR, CNN-GPR, ANN-GPR and SVR-GPR. QR and GPR are all probabilistic forecasting models. The prediction results obtained by QR are a series of conditional quantiles of runoff, and it is necessary to combine the KDE (He & Li 2018) to obtain the probability distribution function, called QR-KDE. These models are realized by Python.

For the fairness of comparison, the nine models use the same set of feature inputs, and the super-parameters of these models are optimized by BOA. The detailed hyper-parameters of nine models are shown in Table 2. The optimal prediction results of these models are taken as the final results.

In this case, there are five tasks that need to be completed.

Task I: Analyze the correlation factors of runoff and construct feature inputs.

Task II: Verify the convergence of BOA and prediction model to ensure the prediction results obtained by each model are optimal.

Task III: Compare different models from point prediction accuracy, interval prediction suitability, and probability prediction comprehensive performance.

Task IV: Analyze hyper-parameter sensitivity and provide suggestions for hyper-parameter optimization.

5.2. Task I: construct feature inputs

To analyze the correlation factors of the runoff and improve the prediction accuracy of the model, historical runoff data are selected as alternative feature. The maximal information coefficient (MIC) values between historical runoff and current runoff are calculated in Table 3. The feature y_{i-1} represents the runoff of the previous period, called period feature. The feature $y_{i-T_{\text{year}}}$ represents the runoff of the day last year, called year feature. Period features with MIC values greater than 0.85 and year features with MIC values greater 0.8 are filled with gray.

Therefore, the observations (labels) for the four datasets are all $Y_i = [y_i]$, and the feature inputs for the four datasets are $X_i = [y_{i-T_{\text{year}}}, y_{i-2*T_{\text{year}}}, y_{i-1}, y_{i-2}, \cdots, y_{i-5}]$, $X_i = [y_{i-T_{\text{year}}}, y_{i-2*T_{\text{year}}}, y_{i-1}, y_{i-2}, \cdots, y_{i-6}]$, $X_i = [y_{i-T_{\text{year}}}, y_{i-2*T_{\text{year}}}, y_{i-1}, y_{i-2}, y_{i-3}]$ and $X_i = [y_{i-T_{\text{year}}}, y_{i-2*T_{\text{year}}}, y_{i-1}, y_{i-2}, \cdots, y_{i-6}]$, respectively.

5.3. Task II: verify convergence

To ensure that the prediction results obtained by each model are optimal, the XGB model of Dataset 4 is taken as an example to verify the convergence of the super-parameter optimization model BOA and the prediction model. Convergence curves of BOA and XGB are shown in Figure 4. The number of iterations of BOA and XGB models is set to 100 and 500, respectively. The loss function is measured by the metric RMSE, and it is calculated using the normalized runoff. In the BOA model, the loss of each epoch varies widely, but the optimal loss continues to decrease throughout the iteration and converges at approximately 65th epoch. The loss of XGB is declining. After 200 epochs, the loss varies very little. Therefore, the model can converge whether it is optimizing the hyper-parameter or training the prediction model.

5.4. Task III: compare different models

To fully verify the performance of the model proposed in this paper, the comparison is made from three aspects: point prediction, interval prediction and probability prediction.

5.4.1. Point prediction comparison

Point prediction comparison is to verify the prediction accuracy of XGB-GPR-BOA. Point prediction metrics of nine models on four datasets are shown in Table 4. The best and second-best metrics are highlighted with dark and light gray backgrounds, respectively. Taking Dataset 1 as an example, RMSE, MAPE and R^2 of XGB are 1,847 m^3/s, 8.09% and 0.965, respectively. Compared with metrics of other models, RMSE and MAPE of XGB are significantly smaller, and R^2 of XGB is much higher. There are similar results in other datasets. These metrics indicate that the runoff prediction results obtained by XGB are the most accurate.

Table 2 | The hyper-parameters of the nine models

Model	Hyper-parameter	Search range
XGB	Maximum depth of a tree	[1, 10]; Integer
	Learning rate	[0.001, 0.3]; Real
	Penalty coefficient of $L1$ regularization	[0, 1]; Real
	Penalty coefficient of $L2$ regularization	[0, 1]; Real
	Subsample ratio of the training instances	[0.7, 1]; Real
	Subsample ratio of columns	[0.7, 1]; Real
	Subsample ratio of columns for each level	[0.7, 1]; Real
	Subsample ratio of columns for each node	[0.7, 1]; Real
LGB	Maximum depth of a tree	[1, 10]; Integer
	Maximum number of leaves in one tree	[20, 200, 5]; Integer
	Learning rate	[0.001, 0.3]; Real
	Penalty coefficient of $L1$ regularization	[0, 1]; Real
	Penalty coefficient of $L2$ regularization	[0, 1]; Real
	Minimal number of data in one leaf	[10, 200, 5]; Integer
	Minimal sum hessian in one leaf	[0.0001, 0.005]; Real
	Bagging fraction	[0.1, 1]; Real
	Frequency for bagging	[20, 60, 5]; Integer
	Feature fraction	[0.7, 1]; Real
GBR	Maximum depth of a tree	[1, 10]; Integer
	Learning rate	[0.001, 0.3]; Real
	The number of boosting stages to perform	[50, 200, 10]; Integer
	Subsample ratio of the training instances	[0.7, 1]; Real
	The minimum number of samples required to split an internal node	[10, 200, 5]; Integer
	The minimum number of samples required to be at a leaf node	[10, 200, 5]; Integer
LSTM	The number of hidden layer nodes	[64, 32, 16, 8, 4, 2, 1]
	Drop out rate	[0.01, 0.2]; Real
	Batch size	[64, 32, 16, 8]
	Epochs	50
	Optimizer	Adam
CNN	The number of hidden layer nodes	[64, 32, 16, 8, 4, 2, 1]
	Kernel size	1
	Strides	2
	Activation	[‘relu’, ‘tanh’, ‘sigmoid’]
	Batch size	[64, 32, 16, 8]
	Epochs	50
	Optimizer	Adam
ANN	The number of hidden layer nodes	[64, 32, 16, 8, 4, 2, 1]
	Activation	[‘relu’, ‘tanh’, ‘sigmoid’]
	Batch size	[64, 32, 16, 8]
	Epochs	50
	Optimizer	Adam
SVR	Kernel function	[‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’]
QR	Kernel function	[‘epa’, ‘cos’, ‘gau’, ‘par’]
GPR	Kernel function	[‘C’, ‘RBF’, ‘W’, ‘M’, ‘RQ’]

Table 3 | MIC between historical runoff and current runoff

Features	$y_{i-T_{year}}$	$y_{i-2*T_{year}}$	y_{i-1}	y_{i-2}	y_{i-3}	y_{i-4}	y_{i-5}	y_{i-6}	y_{i-7}	y_{i-8}	y_{i-9}
Dataset 1	0.846	0.875	0.983	0.943	0.911	0.881	0.861	0.839	0.830	0.814	0.800
Dataset 2	0.861	0.841	0.983	0.956	0.924	0.893	0.877	0.858	0.846	0.827	0.799
Dataset 3	0.861	0.843	0.908	0.879	0.858	0.839	0.825	0.796	0.781	0.766	0.756
Dataset 4	0.852	0.864	0.964	0.938	0.916	0.893	0.876	0.865	0.850	0.836	0.831

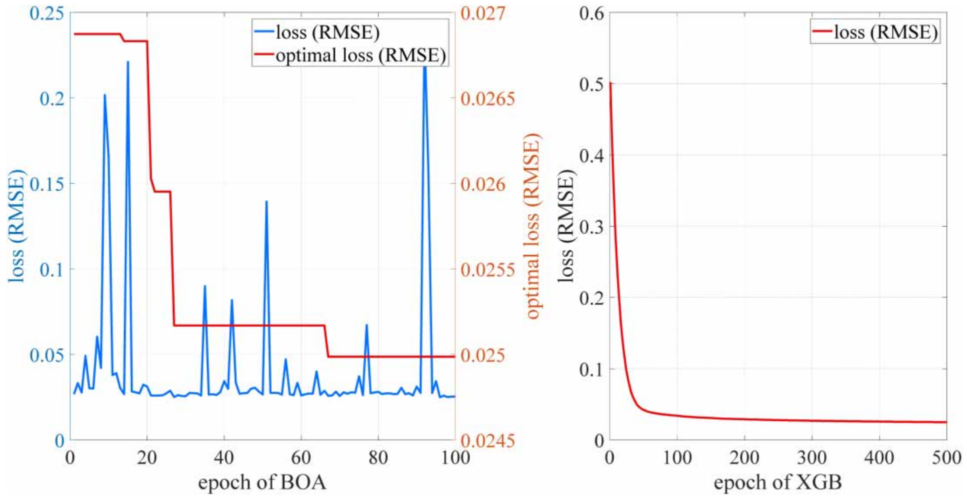


Figure 4 | Convergence curves of BOA and XGB on Dataset 4.

Table 4 | Point prediction metrics of nine models on four datasets

Datasets	Metrics	XGB	LGB	GBR	LSTM	CNN	ANN	SVR	QR	GPR
Dataset 1	RMSE	1,847	2,196	2,235	2,533	2,241	2,298	2,364	2,383	2,756
	MAPE (%)	8.09	9.54	9.93	11.87	11.30	10.60	12.34	12.20	11.83
	R^2	0.965	0.951	0.949	0.935	0.949	0.946	0.943	0.942	0.923
Dataset 2	RMSE	2,734	3,089	3,289	3,450	3,273	3,069	3,740	3,484	3,670
	MAPE (%)	8.97	10.16	10.91	16.15	15.51	12.26	19.06	16.31	14.43
	R^2	0.910	0.885	0.869	0.856	0.871	0.886	0.831	0.853	0.837
Dataset 3	RMSE	790	904	978	868	881	898	1,005	980	964
	MAPE (%)	11.45	12.70	14.65	12.29	12.74	13.79	14.24	14.53	14.65
	R^2	0.823	0.769	0.729	0.787	0.780	0.772	0.714	0.728	0.737
Dataset 4	RMSE	530	573	606	620	637	572	631	565	744
	MAPE (%)	6.68	7.01	7.61	9.07	12.28	8.55	8.93	8.09	9.68
	R^2	0.977	0.973	0.970	0.968	0.967	0.973	0.967	0.974	0.954

5.4.2. Interval prediction comparison

Interval prediction comparison is to verify the suitability of interval obtained by XGB-GPR-BOA. Interval prediction metrics of the nine models on four datasets are shown in Table 5. The best and second best metrics are highlighted with dark and light gray background, respectively. Taking Dataset 1 as an example, since these models use almost the same mechanism to quantify the uncertainty of the forecast, the interval widths of the nine models are relatively close, both around 0.85. In the case where the interval widths are close, an interval with higher CP is more suitable. Since the point prediction accuracy of XGB is higher than other models, the CP of XGB-GPR is higher than other models. Therefore, $MC_{95\%}$ of XGB-GPR is 0.846, which is the smallest of the nine models, indicating that the interval obtained by XGB-GPR is most appropriate. There are similar results in other datasets.

To more vividly compare the interval prediction suitability, the interval prediction results on four datasets are shown in Figure 5. In each figure, the upper part is the interval prediction results of XGB-GPR, and the lower part is a comparison of the interval prediction metrics. It can be clearly seen from the figure that the blue curve and the red curve are very close, and most points of red curve are located in the gray interval, which indicates that the prediction results obtained by XGB-GPR have high accuracy and coverage. Meanwhile, the same conclusion can be drawn from the metric comparison histogram: the interval obtained by XGB-GPR is most appropriate among nine models.

Table 5 | Interval prediction metrics of nine models on four datasets

Datasets	Metrics	XGB-GPR	LGB-GPR	GBR-GPR	LSTM-GPR	CNN-GPR	ANN-GPR	SVR-GPR	QR-KDE	GPR
Dataset 1	$CP_{95\%}$	0.948	0.944	0.945	0.917	0.938	0.940	0.923	0.922	0.938
	$MWP_{95\%}$	0.802	0.844	0.905	0.882	0.873	0.845	0.884	0.844	0.849
	$MC_{95\%}$	0.846	0.894	0.957	0.962	0.930	0.899	0.957	0.915	0.905
Dataset 2	$CP_{95\%}$	0.933	0.927	0.923	0.905	0.915	0.930	0.921	0.901	0.926
	$MWP_{95\%}$	1.023	1.018	1.051	1.100	1.079	1.032	1.319	1.148	1.061
	$MC_{95\%}$	1.097	1.098	1.139	1.215	1.179	1.110	1.433	1.274	1.146
Dataset 3	$CP_{95\%}$	0.960	0.948	0.936	0.948	0.946	0.918	0.943	0.946	0.953
	$MWP_{95\%}$	1.451	1.500	1.546	1.513	1.478	1.588	1.539	1.527	1.516
	$MC_{95\%}$	1.512	1.582	1.651	1.596	1.562	1.730	1.633	1.614	1.589
Dataset 4	$CP_{95\%}$	0.971	0.962	0.952	0.965	0.942	0.962	0.951	0.958	0.964
	$MWP_{95\%}$	0.820	0.827	0.876	0.900	0.902	0.868	0.910	0.866	0.850
	$MC_{95\%}$	0.845	0.860	0.920	0.932	0.958	0.902	0.957	0.904	0.881

5.4.3. Probability prediction comparison

Probability prediction comparison is to verify the comprehensive performance of probability density function obtained by XGB-GPR-BOA. The probability prediction metrics (CRPS) on four datasets are shown in Table 6. The best and second-best metrics are highlighted with dark and light gray backgrounds, respectively. The CRPS values of XGB-GPR on four datasets are 874, 1,145, 204 and 262, which are the smallest among nine models, indicating that the comprehensive performance of probability density function obtained by XGB-GPR is the best. It is also consistent with the results of point prediction and interval prediction.

Probability density functions obtained by XGB-GPR of eight periods of equidistant sampling on Dataset 3 are shown in Figure 6. In general, these curves are very full, and no curve is excessively high or low, wide or narrow, indicating that probability density functions obtained by XGB-GPR are suitable. In period 1, 157, 783 and 1,096, the observation lines are near the center of the curve, which show these points' prediction accuracy is high. In periods 313, 470, 626 and 939, the observation lines are far from the center. In probability prediction results of validation set, some observations are close to the center and other observations are off-center, which indicates that the probabilistic forecast is reliable. If all points are at the center or far from the center, we may not be convinced of these probabilistic forecast results.

5.5. Task IV: analyze hyper-parameter sensitivity

It is necessary to analyze the change process of super-parameters when BOA is used to optimize the super-parameters of XGB, and provide the suggestions for practical application tuning. The scatter plot of the hyper-parameter and loss is shown in Figure 7. The eight scatter plots are the relationship plot between maximum depth of a tree ($P1$), learning rate ($P2$), penalty coefficient of $L1$ regularization ($P3$), penalty coefficient of $L2$ regularization ($P4$), subsample ratio of the training instances ($P5$), subsample ratio of columns ($P6$), subsample ratio of columns for each level ($P7$) and subsample ratio of columns for each node ($P8$) and loss.

Some parameter optimization results can be obtained by analyzing these scatter plots:

1. None of the scatter plots exhibit a distinct linear or nonlinear relationship. Most of the scatter shapes are nearly horizontal, indicating that the performance of XGB is not extremely sensitive to any of the hyper-parameters. When XGB is optimized by BOA, all parameters are combined to affect performance.
2. Hyper-parameter $P1$ is one of the most important core parameters in XGB. In theory, increasing $P1$ will improve the performance of the model, but too large $P1$ will make the model complex, consume more memory resources and more likely to overfitting. In the first subgraph, $P1$ under optimal loss is evenly distributed from 1 to 10, which indicates that the model performance is not sensitive to the $P1$.
3. Hyper-parameter $P2$ is also one of the most important core parameters in XGB. As can be seen from the second subgraph, a small learning rate will cause the XGB model to fail to converge. The learning rate under the optimal loss is mainly concentrated between 0.05 and 0.1; therefore, it is recommended to limit learning rate to this range when applied to the actual situation.

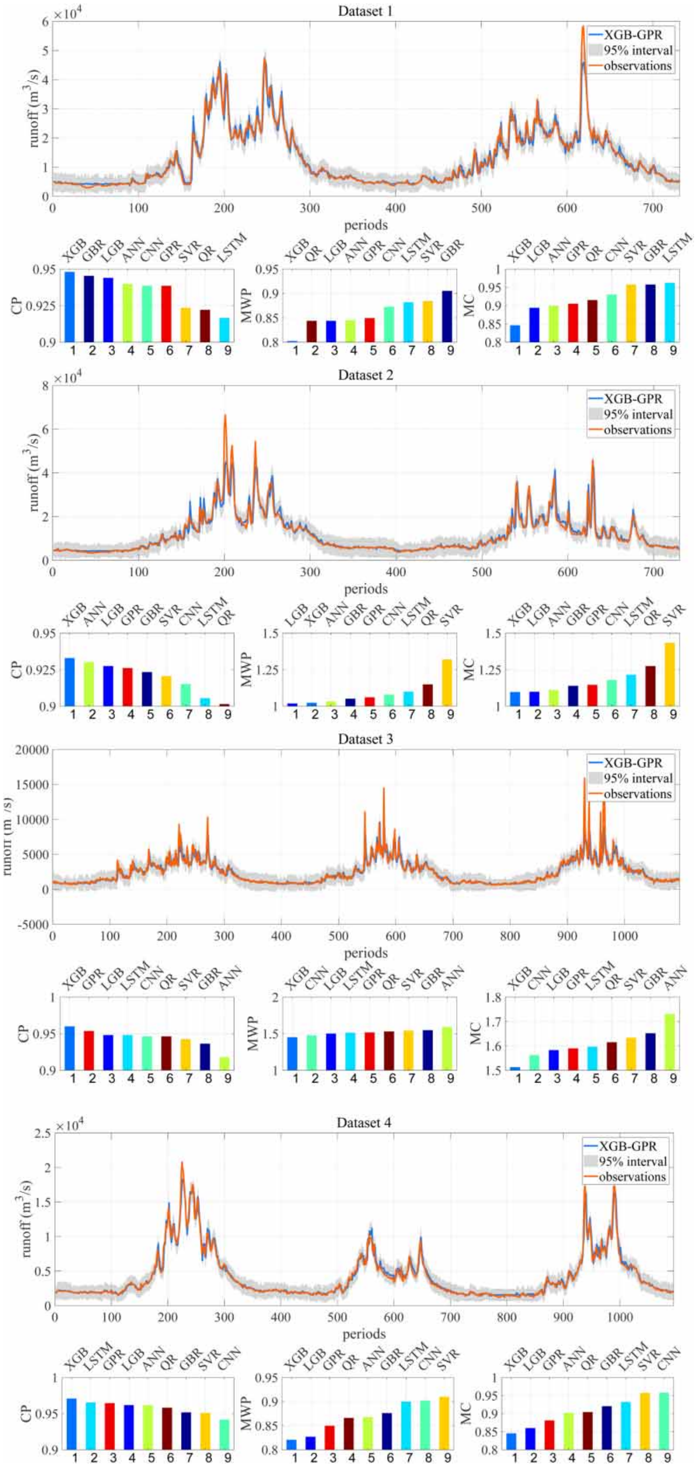


Figure 5 | Interval prediction results on four datasets. CP, coverage probability; MWP, mean width percentage; MC, suitability metric. Please refer to the online version of this paper to see this figure in color: http://dx.doi.org/10.2166/nh.2021.161.

4. *P*3 and *P*4 are parameters that avoid overfitting by regularization. *P*3 and *P*4 points less than 0.5 are more than points greater than 0.5, so it is recommended to limit *P*3 and *P*4 to below 0.5.
5. *P*5–*P*8 are parameters that avoid overfitting by subsampling. Similar to results (3) and (4), it is recommended that *P*5 is between 0.7 and 0.9, *P*6 is between 0.7 and 1.0, *P*7 is between 0.8 and 1.0 and *P*8 is between 0.8 and 1.0.

Table 6 | Probability prediction metrics (CRPS) of nine models on four datasets

Datasets	XGB-GPR	LGB-GPR	GBR-GPR	LSTM-GPR	CNN-GPR	ANN-GPR	SVR-GPR	QR-KDE	GPR
Dataset 1	874	991	1,025	1,152	1,022	1,024	1,066	1,046	1,150
Dataset 2	1,145	1,258	1,319	1,503	1,453	1,292	1,634	1,435	1,519
Dataset 3	304	334	373	339	336	409	372	364	357
Dataset 4	262	278	293	307	331	287	311	276	338

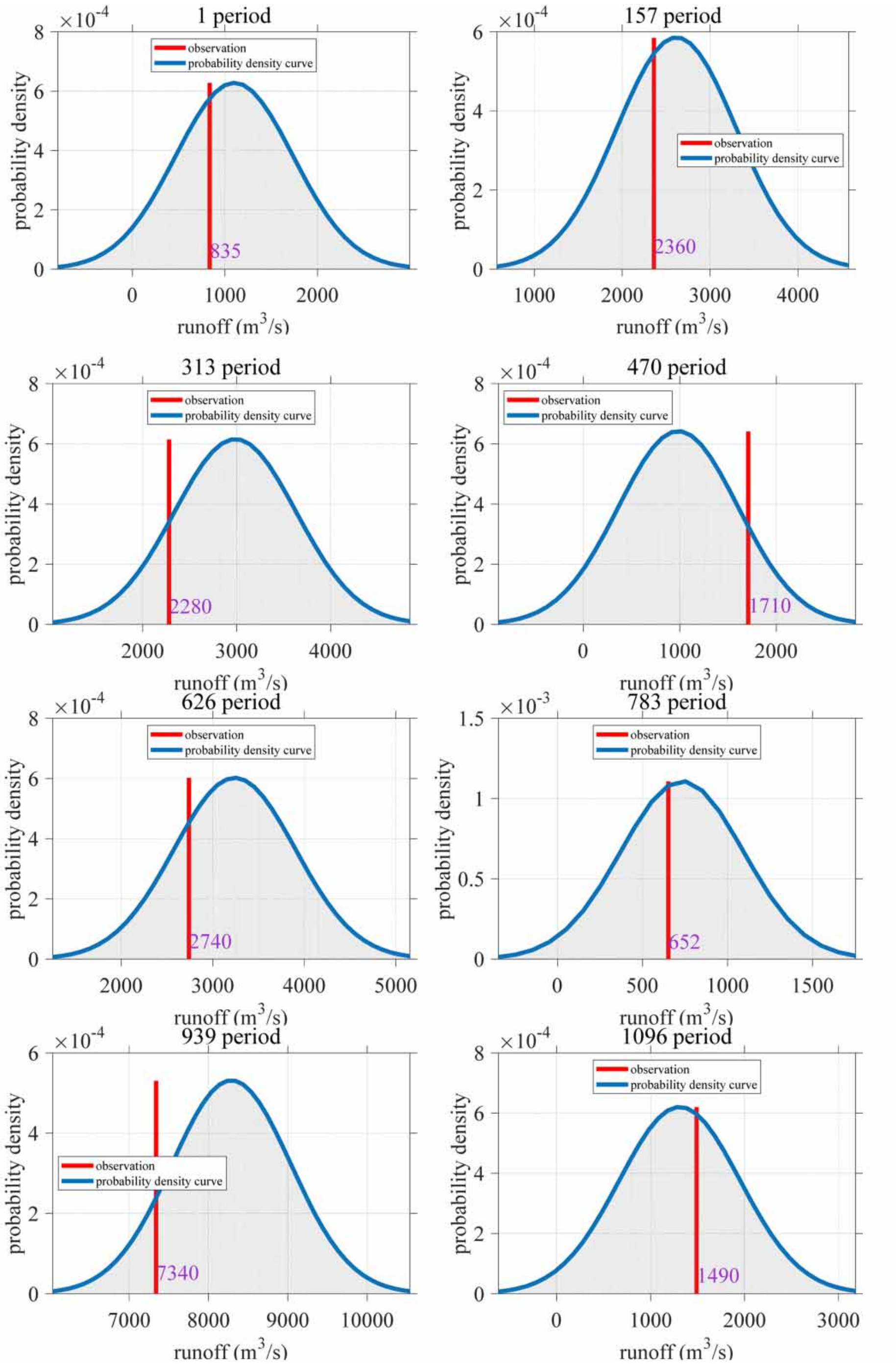


Figure 6 | Probability density function obtained by XGB-GPR on Dataset 3.

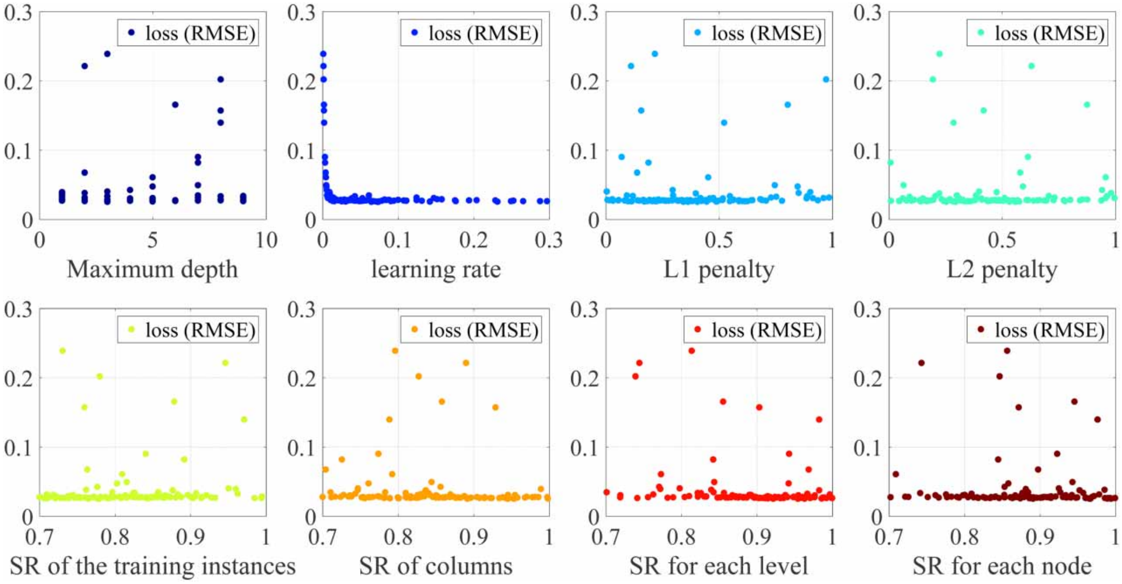


Figure 7 | Hyper-parameter sensitivity analysis of XGB-GPR on Dataset 4.

The most important parameter in the GPR of XGB-GPR is the selection of kernel functions. The losses of different kernel functions on Dataset 4 are listed in Table 7. *C*, *W*, *M*, RBF and RQ are constant kernel, white kernel, matern kernel, radial basis function kernel and rational quadratic kernel, respectively. The RQ corresponds to the least loss; therefore, the recommended kernel function is RQ in the practical application.

5.6. Discussion

The advantages of the forecast model proposed in this study are mainly in two aspects: (1) probabilistic forecast results can quantify forecast uncertainty and provide decision-makers with richer information; (2) model hyper-parameters are screened by BOA to ensure forecast accuracy. The disadvantage of the model is that the prediction result is a one-step ahead forecasting result, and multiple models need to be trained in a multi-step ahead furcating scenario. In application, we can use historical data to realize runoff scroll forecasting in future coming years.

To clarify the knowledge gap between the existing research and our research, the differences are presented as follows:

1. Existing studies focus on the deterministic prediction of runoff, and the model proposed in our research is the deep learning probabilistic model.
2. In some studies, the model super-parameters were not selected or selected manually, while in our model, the super-parameters were obtained through BOA.

6. CONCLUSIONS AND FUTURE WORKS

In this study, a probabilistic forecasting hybrid model called XGB-GPR-BOA is proposed to predict runoff and quantify the uncertainty of prediction. At first, XGB, as a novel gradient tree boosting algorithm, is applied to predict runoff, which can ensure the prediction accuracy. However, XGB can only obtain point prediction results, unable to quantify the uncertainty of the forecast. Then, it is assumed that the runoff of each period obeys a Gaussian distribution. The combination of XGB and GPR makes it possible to quantify the uncertainty of the forecast. Finally, in order to make the model show better performance, the MIC is used to filter the feature input and the BOA is used to optimize the hyper-parameter of the model.

Table 7 | Losses of different kernel functions on Dataset 4

Kernel	*C*	*W*	*M*	RBF	RQ
Loss (RMSE)	0.170	0.222	0.196	0.379	0.025

The hybrid model XGB-GPR-BOA is applied to predict runoff for four actual cases in the Yangtze River Basin of China. The nine state-of-the-art models use seven evaluation metrics (RMSE, MAPE, R^2, CP_α, MWP_α, MC_α and CRPS) to verify from three aspects: point prediction accuracy, interval prediction suitability and probability prediction comprehensive performance.

The main conclusions of this study are summarized as follows:

1. XGB-GPR-BOA can obtain high-precision point prediction, appropriate prediction interval and high-performance probabilistic prediction results.
2. The optimal hyper-parameters of the model obtained by BOA are conducive to improving the prediction accuracy.

ACKNOWLEDGEMENTS

The authors want to thank Yongqi Liu and Shaoqian Pei for their help in the fieldwork.

AUTHOR CONTRIBUTIONS

H.B. and H.Q. conceptualized the study; G.L. did the data curation; C.L. investigated the study; B.L. formulated the study methodology; H.Q. provided resources; Z. Z. provided software; Z.Z. provided supervision; H.Q. provided validation; B.L. did the visualization; writing of original draft was done by H.B. and writing of review & editing was done by H.Q. and Z.Z.

FUNDING

This work is supported by the National Natural Science Foundation of China (No. 51979113), and special thanks are given to the anonymous reviewers and editors for their constructive comments.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

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

REFERENCES

Al-Muhammed, M. J. & Abu Zitar, R. 2018 Probability-directed random search algorithm for unconstrained optimization problem. *Appl. Soft Comput.* **71**, 165–182.

Behera, P. K., Adams, B. J. & Li, J. Y. 2006 Runoff quality analysis of urban catchments with analytical probabilistic models. *J. Water Resour. Plann. Manage.* **132** (1), 4–14.

Chen, T. & Guestrin, C. 2016 XGBoost: a scalable tree boosting system. *ACM* 785–794. https://dl.acm.org/doi/10.1145/2939672.2939785

Deng, L., Pan, J., Xu, X., Yang, W., Liu, C. & Liu, H. 2018 PDRLGB: precise DNA-binding residue prediction using a light gradient boosting machine. *BMC Bioinf.* **19**, 145–522.

Fang, Y., Zhang, X., Corbari, C., Mancini, M., Niu, G. & Zeng, W. 2017 Improving the Xin'anjiang hydrological model based on mass–energy balance. *Hydrol. Earth Syst. Sci.* **21**, 3359–3375.

He, Y. & Li, H. 2018 Probability density forecasting of wind power using quantile regression neural network and kernel density estimation. *Energy Convers. Manage.* **164**, 374–384.

He, F., Zhou, J., Feng, Z., Liu, G. & Yang, Y. 2019 A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm. *Appl. Energy* **237**, 103–116.

Hersbach, H. 2000 Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather Forecast.* **15**, 559–570.

Khosravi, A., Nahavandi, S., Creighton, D. & Atiya, A. F. 2011 Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Trans. Neural Netw. Learn. Syst.* **22**, 337–346.

Kong, X., Sun, Y., Su, R. & Shi, X. 2017 Real-time eutrophication status evaluation of coastal waters using support vector machine with grid search algorithm. *Mar. Pollut. Bull.* **119**, 307–319.

Le Callet, P., Viard-Gaudin, C. & Barba, D. 2006 A convolutional neural network approach for objective video quality assessment. *IEEE Trans. Neural Netw. Learn. Syst.* **17**, 1316–1327.

Li, R. & Jin, Y. 2018 A wind speed interval prediction system based on multi-objective optimization for machine learning method. *Appl. Energy* **228**, 2207–2220.

Liu, Y., Ye, L., Qin, H., Hong, X., Ye, J. & Yin, X. 2018a Monthly streamflow forecasting based on hidden Markov model and Gaussian mixture regression. *J. Hydrol.* **561**, 146–159.

Liu, Y., Qin, H., Mo, L., Wang, Y., Chen, D., Pang, S. & Yin, X. 2018b Hierarchical flood operation rules optimization using multi-objective cultured evolutionary algorithm based on decomposition. *Water Resour. Manage.* **33**, 337–354.

Luo, X., Yuan, X., Zhu, S., Xu, Z., Meng, L. & Peng, J. 2019 A hybrid support vector regression framework for streamflow forecast. *J. Hydrol.* **568**, 184–193.

Mauricio, J. A. 1995 Exact maximum likelihood estimation of stationary vector ARMA models. *J. Am. Stat. Assoc.* **90** (429), 282–291.

Papacharalampous, G., Tyralis, H. & Koutsoyiannis, D. 2018 Predictability of monthly temperature and precipitation using automatic time series forecasting methods. *Acta Geophys.* **66** (4), 807–831.

Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., Yuan, X. & Gu, L. 2019 Feature selection based on artificial bee colony and gradient boosting decision tree. *Appl. Soft Comput.* **74**, 634–642.

Sun, A. Y., Wang, D. & Xu, X. 2014 Monthly streamflow forecasting using Gaussian process regression. *J. Hydrol.* **511**, 72–81.

Tan, Q., Lei, X., Wang, X., Wang, H., Wen, X., Ji, Y. & Kang, A. 2018 An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach. *J. Hydrol.* **567**, 767–780.

Wang, Q. J., Robertson, D. E. & Chiew, F. H. S. 2009 A Bayesian joint probability modeling approach for seasonal forecasting of streamflows at multiple sites. *Water Resour. Res.* **45**, W5407.

Wen, X., Feng, Q., Deo, R. C., Wu, M., Yin, Z., Yang, L. & Singh, V. 2019 Two-phase extreme learning machines integrated with the complete ensemble empirical mode decomposition with adaptive noise algorithm for multi-scale runoff prediction problems. *J. Hydrol.* **570**, 167–184.

Wu, M. & Lin, G. 2017 The very short-term rainfall forecasting for a mountainous watershed by means of an ensemble numerical weather prediction system in Taiwan. *J. Hydrol.* **546**, 60–70.

Yang, T., Gao, X., Sorooshian, S. & Li, X. 2016 Simulating California reservoir operation using the classification and regression-tree algorithm combined with a shuffled cross-validation scheme. *Water Resour. Res.* **52**, 1626–1651.

Zhang, Z., Ye, L., Qin, H., Liu, Y., Wang, C., Yu, X., Yin, X. & Li, J. 2019a Wind speed prediction method using shared weight long short-term memory network and Gaussian process regression. *Appl. Energy* **247**, 270–284.

Zhang, Z., Qin, H., Liu, Y., Yao, L., Yu, X., Lu, J., Jiang, Z. & Feng, Z. 2019b Wind speed forecasting based on quantile regression minimal gated memory network and kernel density estimation. *Energy Convers. Manage.* **196**, 1395–1409.

Zhang, Z., Qin, H., Liu, Y., Wang, Y., Yao, L., Li, Q., Li, J. & Pei, S. 2019c Long short-term memory network based on neighborhood gates for processing complex causality in wind speed prediction. *Energy Convers. Manage.* **192**, 37–51.

Zhang, Z., Qin, H., Li, J., Liu, Y., Yao, L., Wang, Y., Wang, C., Pei, S. & Zhou, J. 2020 Short-term optimal operation of wind-solar-hydro hybrid system considering uncertainties. *Energy Convers. Manage.* **205**, 112405.

First received 25 October 2020; accepted in revised form 18 June 2021. Available online 8 July 2021