

The Ground Temperature Prediction Model Based on Boosting Algorithm

Yumeng Ban

*Institute of Nanjing University of Information Science and Technology, Nanjing, China
15166525265@163.com*

Abstract: Traditional statistical methods are difficult to meet the needs of the demands of temperature prediction. To improve the predicted performance of Boosting algorithm, this study adopts an improved Random Forest algorithm for the feature selection of meteorological data and employs Random Search and Bayesian Optimization for the parameter optimization of the model. The experimental result indicates that the Gradient Boosting Decision Tree(GBDT) based on Bayesian Optimization demonstrates significant superiority in temperature prediction.

Keywords: boosting algorithm, temperature prediction, Random Forest

1. Introduction

In the recent few years, extreme high temperatures occur frequently around the world, which bring lots of inconvenience to human being's daily life. Therefore, the analysis and prediction of temperature changes has become an inevitable trend to solve the problems in climate development [1]. However, the traditional statistical methods show a lower prediction accuracy in the condition of sudden weather changes, data loss, etc. In this situation, deep learning methods can better solve these problems.

The data in this paper comes from the global ground meteorological data provided by the OpenWeatherMap meteorological data website. We select hourly meteorological data from the Nanjing Meteorological Observatory from January 1, 2019 to December 31, 2023.

The dataset mentioned in this study has obvious timing characteristics and meteorological characteristics. Moreover, considering Boosting algorithm's [2] robustness in missing value and excellent performance in large-scale datasets, we select three classic Boosting algorithms—AdaBoost, XGBoost, and GBDT—for temperature prediction. Random Search and Bayesian Optimization are used for parameter optimization to select the optimal model [3].

2. Theoretical basis

This chapter introduces four algorithms, including the Random Forest algorithm used for feature selection, and the AdaBoost, XGBoost, and GBDT algorithms used for comparison of temperature prediction performance.

2.1. Bayesian optimization-based random forest algorithm

Traditional feature selection methods (such as sequential search and random search) [4] can't capture the interactions between features and perform poorly in handling non-linear relationships. Therefore, this paper selects the Random Forest algorithm [5], which is suitable for high-dimensional data and complex features. By optimizing parameters through Bayesian Optimization, the algorithm's performance can be greatly enhanced and identify the optimal feature subset more efficiently [6].

Algorithm 1: Bayesian optimization-based random forest feature selection

Input: $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ represents the training dataset, x_i represents all features except the average temperature, y_i represents the average temperature. Input the number of features M , Input the number of trees T .

Output: Feature Importance Score

- 1、 Define hyperparameter space:
- 2、 Using Bayesian optimization to find the optimal hyperparameter combination.
- 3、 Initialize the Random Forest model with the optimal hyperparameters.

for $t = 1, 2, \dots, T$ **do**

- 4、 Sampling: Randomly draw N samples with replacement from the training set T to obtain a new training set T_t
- 5、 Constructing Decision Trees: For each node, randomly select m features (typically $m = \sqrt{M}$), and use the *Gini* index to choose the best split feature to construct the decision tree $G_t(x)$.
- 6、 Calculating Feature Importance:

$$\Delta Gini(s, j) = Gini(s) - (Gini(s_{left}) + Gini(s_{right}))$$

$Gini(s)$ is the *Gini* index of the node s before splitting, and $Gini(s_{left})$ and $Gini(s_{right})$ are the *Gini* index of the left and right child nodes after splitting.

7、 **end for**

- 8、 The importance score for each feature j is calculated as:

$$Importance_j = \frac{1}{T} \sum_{t=1}^T \sum_{s \in t} \Delta Gini(s, j)$$

9、 **return** $Importance_j$

2.2. Boosting algorithm

Boosting is an ensemble learning method enhances the overall prediction performance of the model through combining multiple weak classifiers into a strong classifier by weighting them together.

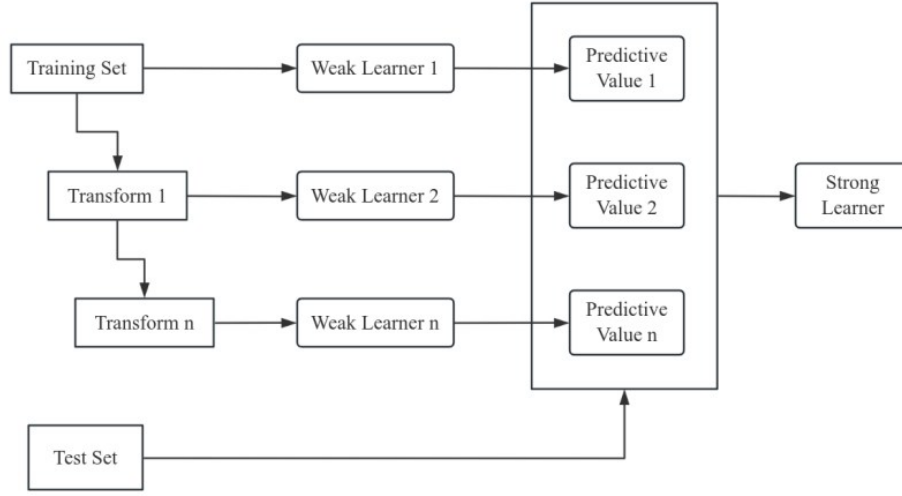


Figure 1: Boosting algorithm

2.2.1. AdaBoost

The AdaBoost model [7] is an adaptive boosting algorithm that constructs a strong classifier through assigning different weights to multiple weak classifiers.

Algorithm2: AdaBoost model

Input: $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ represents the training dataset, x_i are all features except average temperature, and y_i is the average temperature. Total number of samples: N . Number of weak learners: T

Output: The final strong classifier $H(x)$

Initialize the training data weights: $D_1(i) = \frac{1}{N}, i = 1, 2, \dots, N$

1、 **for** $t = 1, 2, \dots, T$ **do**

Train a weak classifier: $\epsilon_t = \sum_{i=1}^N D_t(i) \cdot II(y_i \neq h_t(x_i))$, D_t is the current weight distribution, $h_t(x)$ is the weak classifier, II is the indicator function.

Compute the weight of the weak classifier: $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$, (stop training if $\epsilon_t \geq 0.5$)

2、 Update sample weights:

$$D_{t+1}(i) = \frac{D_t(i) \cdot \exp(-\alpha_t y_i h_t(x_i))}{\sum_{i=1}^N D_t(i) \cdot \exp(-\alpha_t y_i h_t(x_i))}$$

3、 **end for**

Construct the final strong classifier: $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

4、 **return** $H(x)$

2.2.2.XGBoost

The XGBoost model [8] is a gradient boosting framework that constructs the model and enhances its accuracy through adding decision trees iteratively.

Algorithm 3: XGBoost model

Input: $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ represents the training dataset, where x_i are all features except average temperature, and y_i is the average temperature. Total number of samples: N . Loss function: $L(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$, where y_i is the true value of the i -th sample, and $\hat{y}_i$ is the predicted value of the i -th sample. Learning rate: η , Maximum depth of trees: max_depth , Regularization parameters: λ and γ

Output: Final prediction function $F(x)$

1、 Initialize the model: $f_0(x) = \frac{1}{N} \sum_{i=1}^N y_i$

2、 **for** $t = 1, 2, \dots, T$ **do**

 Compute residuals: $r_i^{(t)} = y_i - \hat{y}_i^{(t-1)}$

3、 Build the t -th tree $q_t(x)$ and define the objective function: $\sum_{i=1}^N [r_i^{(t)} q_t(x_i) + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T \omega_j^2]$, ω_j represents the weight of the j -th leaf node

4、 Update the model: $F_t(x) = F_{t-1}(x) + \eta q_t(x)$

5、 **end for**

6、 Final prediction function: $F(x) = F_T(x)$

7、 **return** $F(x)$

2.2.3. Gradient boosting decision tree(GBDT)

Gradient Boosting Decision Tree (GBDT) [9] is a model based on gradient boosting trees. Its fundamental idea is constructing the model by reducing the residuals.

Algorithm 4: Gradient boosting decision tree (GBDT) model

Input: $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ represents the training dataset, x_i are all features except average temperature, and y_i is the average temperature. Total number of samples: N . Learning rate: η .

Output: Final prediction function $F(x)$.

1、 **Initialize the training data weights:** $D_I(i) = \frac{1}{N}, i = 1, 2, \dots, N$

2、 **for** $t = 1, 2, \dots, T$ **do**

3、 Compute residuals: $r_i^{(t)} = y_i - \hat{y}_i^{(t-1)}$

4、 **Train a new decision tree** $h_t(x)$ using the residuals $r_i^{(t)}$ as the target values.

5、 Update the model: $F_t(x) = F_{t-1}(x) + \eta h_t(x)$

6、 **end for**

7、 Final prediction function: $F(x) = \sum_{t=1}^T \eta h_t(x)$

8、 **return** $F(x)$

2.3. Parameter optimization method

Traditional parameter optimization methods include grid search [10], random search, Newton's method, Bayesian optimization, etc. For large-scale temperature prediction problems, grid search and Newton's method have the same disadvantage of high computational costs. Therefore, in this paper, we select the random search algorithm and the Bayesian optimization algorithm, which are more flexible and efficient. then, apply them to three Boosting models for parameter optimization in order to obtain more accurate models.

Random search is a parameter optimization method based on random sampling. It find the optimal solution by randomly selecting combinations of parameters in the parameter space. Random search does not need traverse all parameter combinations, instead, it samples the parameter space randomly to identify better parameter combinations quickly.

Bayesian Optimization is a parameter optimization method based on probability models. It guides the search process by constructing a probability model of the objective function. It approximates the distribution of the objective function through construct conceptual models (usually a Gaussian Process) and select optimal parameter combinations through acquisition function (such as Expected Improvement).

3. Data preprocessing

Data preprocessing refers to a series of necessary processes on raw data, such as cleaning, transformation, and scaling, to make the data more suitable for machine learning algorithms. The preprocessing methods applied in this paper mainly includes handling missing values, dealing with outliers, normalizing the data, and feature selection [11].

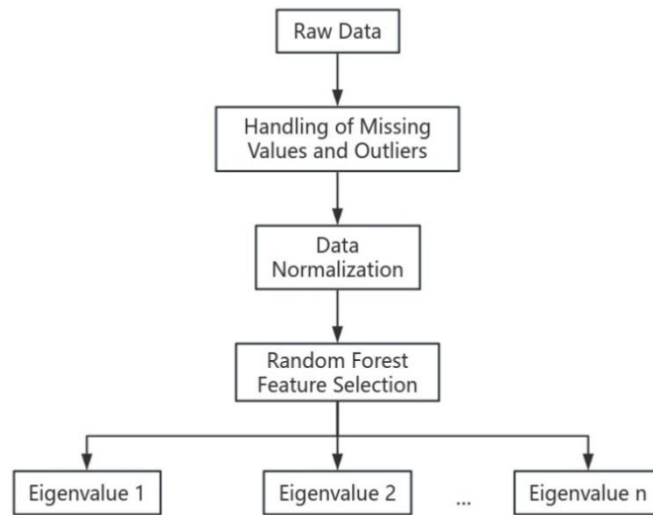


Figure 2: Data preprocessing

Missing Values Handling: In the dataset, five indicators—wind_speed, rain_1h, rain_3h, snow_1h, snow_3h—contain missing values. These five indicators are either completely missing or have large area missing values. Therefore, they can be directly removed from the dataset.

Outliers Handling: Outliers are observations that significantly deviate from the normal range or are extremely rare which may interfere model's training. In this paper, we use the mean imputation method to handle outliers. According to statistics, the highest recorded temperature in Nanjing is

40.5°C, and the lowest is -13.1°C. Any temperature outside this range is identified as an outlier which will be replaced with the average value of nearby data points.

Data Normalization [12]: Various meteorological indicators have different scales and significant differences. Using the raw data directly would emphasize the impact of high-value indicators in the model and neglect the impact of low-value indicators, which would increasing the error. Therefore, it is necessary to standardize the raw data. This paper adopt the Min-Max normalization method, with the expression as follows:

$$x' = \frac{x - MIN(X)}{MAX(X) - MIN(X)}$$

$MAX(X)$ represents the maximum value in the sample data, and $MIN(X)$ represents the minimum value in the sample data.

Feature Selection: In this paper, we use Random Forest model for feature selection, and optimize its hyperparameters through Bayesian algorithm. Continuous variables other than temperature are used as independent variables, with temperature values as the dependent variable. The feature importance ranking output by the Random Forest model is as follows:

Table 1: Importance ranking

Rank	Index	Significance Score
1	temp_min	0.211689
2	feels_like	0.194209
3	temp_max	0.181537
4	dew_point	0.140822
5	pressure	0.140321
6	humidity	0.012432
7	clouds_all	0.011601
8	wind_deg	0.008111
9	wind_speed	0.003505

In summary, the main indicators selected in this paper are temp_min, feels_like, temp_max, dew_point and pressure.

4. Experiments

The meteorological data from January 1, 2019, to December 31, 2023, were divided into training and testing sets, with the first 70% used as the training set and the remaining 30% as the testing set.

4.1. Evaluation metrics

4.1.1. Mean squared error(MSE)

MSE is the average of the squared differences between the predicted values and the actual values. The smaller the MSE value, the more accurate the model's predictions. The formula for calculating MSE is as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

y_i is the actual value, $\hat{y}_i$ is the predict value, n is the number of samples

4.1.2. Model training time

Training time refers to the duration required for a model to learn from the training dataset. The shorter the training time, the higher the efficiency of the model in practical applications.

4.1.3. Model prediction time

Prediction time refers to the duration required for a model to make predictions on new data. The shorter the prediction time, the higher the efficiency of the model in practical applications.

4.2. Model comparison and evaluation

Table 2: AdaBoost optimized parameter settings

Parameter	Parameter Settings		
	Original Model	Random Search	Bayesian
<i>n_estimators</i>	50	150	61
<i>learning_rate</i>	1	0.5	0.241
<i>max_depth</i>	3	5	9
<i>min_samples_split</i>	2	2	16
<i>min_samples_leaf</i>	1	2	3

Table 3: XGBoost optimized parameter settings

Parameter	Parameter Settings		
	Original Model	Random Search	Bayesian
<i>n_estimators</i>	100	249	486
<i>learning_rate</i>	0.1	0.15	0.026
<i>max_depth</i>	6	9	9
<i>subsample</i>	1	0.84	0.77
<i>colsample_bytree</i>	1	0.83	0.78

Table 4: GBDT optimized parameter settings

Parameter	Parameter Settings		
	Original Model	Random Search	Bayesian
<i>n_estimators</i>	100	306	500
<i>learning_rate</i>	0.1	0.18	0.12
<i>max_depth</i>	3	9	10
<i>min_samples_split</i>	2	10	20
<i>min_samples_leaf</i>	2	7	10

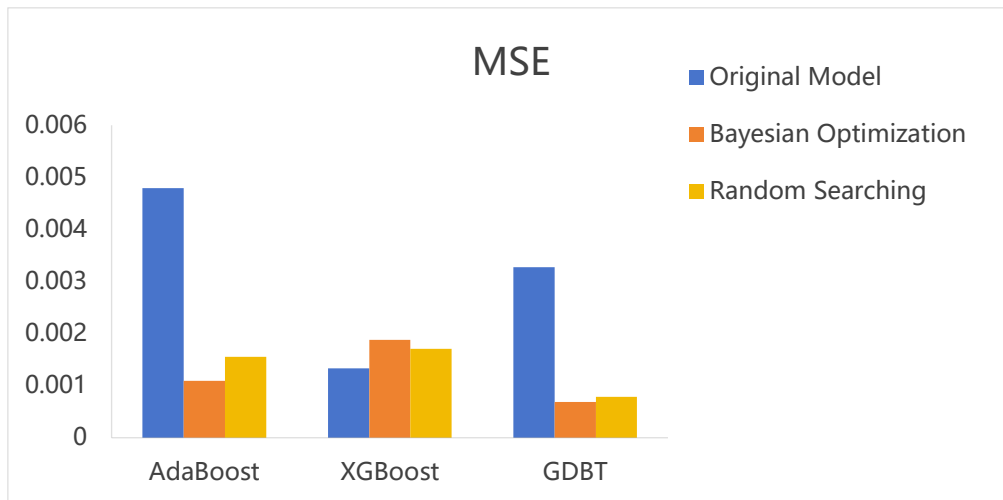


Figure 3: MSE predicted by boosting model

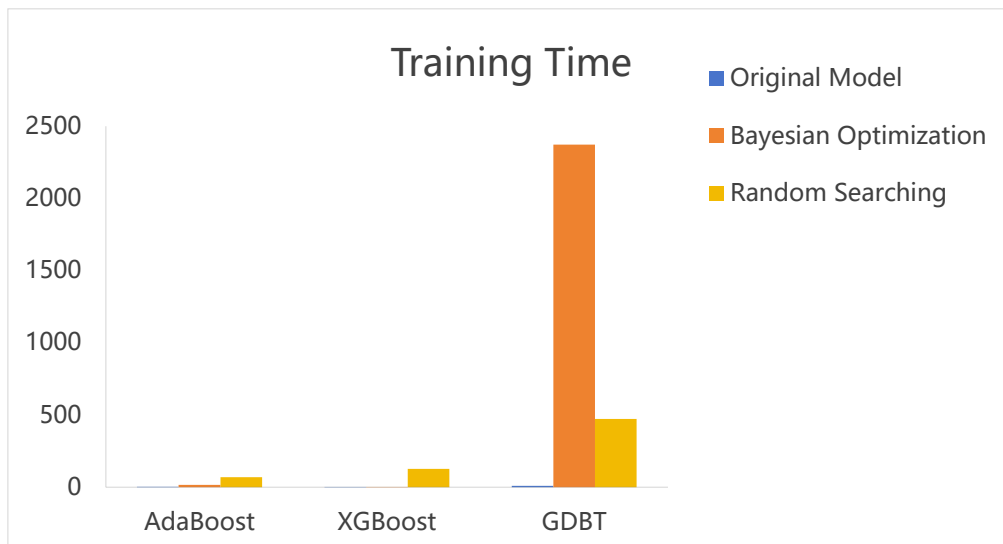


Figure 4: The training time of boosting model

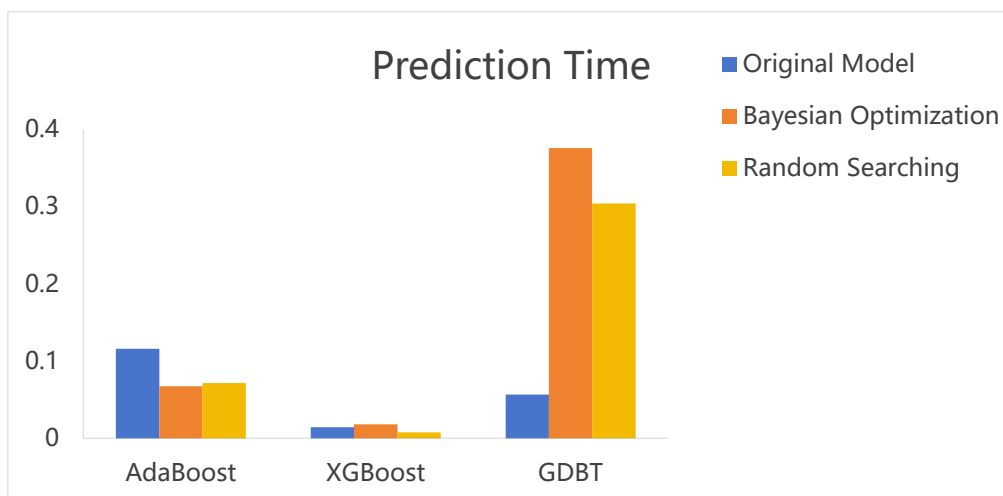


Figure 5: The prediction time of boosting model

By comparing Figure 3, Figure 4, and Figure 5, it can be observed that the GBDT model optimized using Bayesian optimization has the smallest MSE and RMSE. It indicates the best performance in temperature prediction. However, during the experiments, it was found that the GBDT model with Bayesian optimization has long training and prediction times, which may reveal a disadvantage of lower efficiency in practical temperature forecasting.

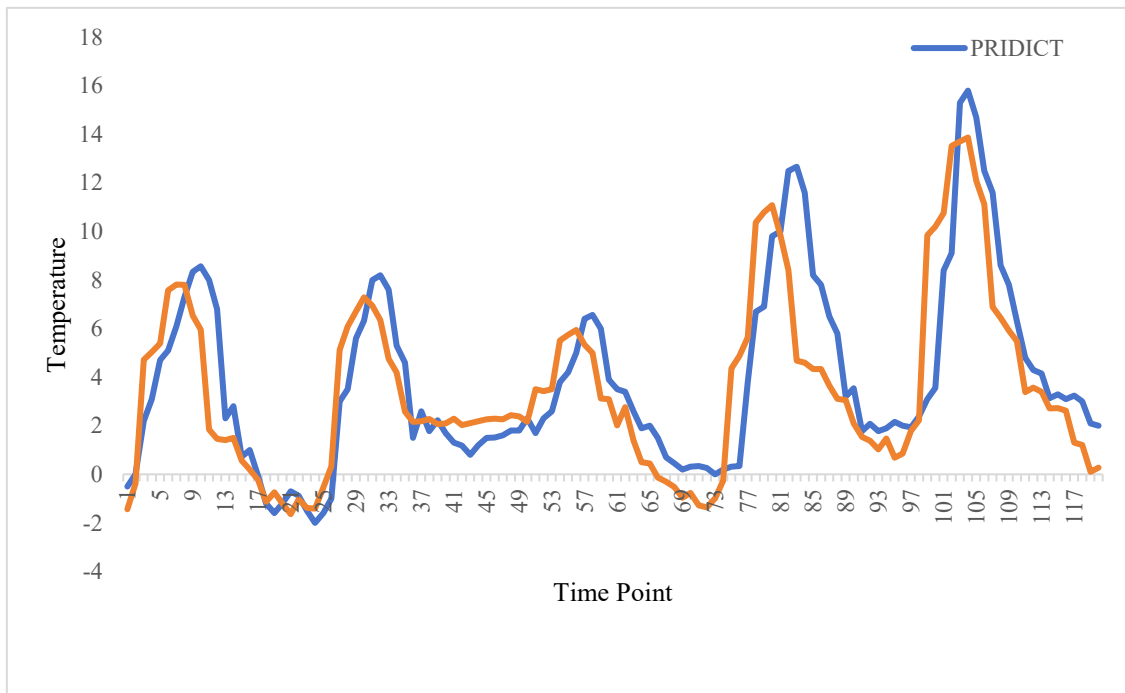


Figure 6: Fitting diagram of the temperature prediction for GBDT

5. Conclusion

By comparing the prediction results on the test set, all three models demonstrated good robustness. The GBDT model optimized using Bayesian optimization achieved the best performance in terms of Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), while AdaBoost and XGBoost had relatively shorter training times. We use cross-validation [13] during training to split the training set and validation set, which effectively avoiding overfitting caused by unreasonable data splitting.

To address the long training time of GBDT, more efficient gradient boosting algorithms such as LightGBM [14] or CatBoost could be explored in the future. These algorithms can provide faster training speeds while maintaining accuracy. Additionally, to further enhance the model's ability to predict extreme events, data augmentation techniques [15] could be employed to generate more training samples targeting rare climate patterns.

Through these measures, we expect to further improve the performance of temperature prediction models in future research. This will help better address the challenges posed by climate change and provide strong support for research and practice in the field of temperature prediction.

References

- [1] Bai Xue. *Temperature Prediction and Heatwave Warning Analysis Based on Time Series Models and XGBoost* [D]. Nanjing University of Information Science and Technology, 2021. DOI:10.27248/d.cnki.gnjqc.2021.000581.
- [2] Lu Hairong. *Diabetes Prediction Model Based on Boosting Algorithms* [D]. Southwest University, 2023. DOI:10.27684/d.cnki.gxndx.2023.003014.

- [3] Guo Qingchun, Li Li, Zhang Ran, et al. *Neural Network Prediction Model for Temperature Changes in China* [J]. *Journal of Tropical Meteorology*, 2009, 25(04): 483-487.
- [4] Li Ziqin, Du Jianqiang, Nie Bin, et al. *A Survey of Feature Selection Methods* [J]. *Computer Engineering and Applications*, 2019, 55(24): 10-19.
- [5] Breiman L. *Random forests*[J]. *Machine learning*, 2001, 45(1):5-32.
- [6] Li Yaru, Zhang Yulai, Wang Jiachen. *A Survey of Bayesian Optimization Methods for Hyperparameter Estimation* [J]. *Computer Science*, 2022, 49(S1): 86-92.
- [7] Cao Ying, Miao Qiguang, Liu Jiachen, et al. *Research Progress and Prospects of the AdaBoost Algorithm* [J]. *Acta Automatica Sinica*, 2013, 39(06): 745-758.
- [8] Chen T, Guestrin C. *Xgboost: A scalable tree boosting system*[C]//*Proceedings of the 22nd acm sigkddinternational conference on knowledge discovery and data mining*. 2016:785-794.
- [9] Zhou Jiaying, He Pengfei, Qiu Rongfa, et al. *Intrusion Detection Research Based on Random Forest and Gradient Boosting Tree Fusion* [J]. *Journal of Software*, 2021, 32(10): 3254-3265. DOI:10.13328/j.cnki.jos.006062.
- [10] Sun Lili, Fang Hongbin, Zhu Xingxing, et al. *Stock Prediction Based on XGBoost Model Optimized by Grid Search* [J]. *Journal of Fuyang Normal University (Natural Science Edition)*, 2021, 38(02): 97-101. DOI:10.14096/j.cnki.cn34-1069/n/2096-9341(2021)02-0097-05.
- [11] Yu Tianying. *Short-term Temperature Prediction Model Based on XGBoost and Improved LSTM* [D]. *Huazhong University of Science and Technology*, 2023. DOI:10.27157/d.cnki.ghzku.2023.000307.
- [12] Zhang Minmin. *Research on Temperature Prediction Based on Deep Learning* [D]. *Hebei University of Economics and Business*, 2023. DOI:10.27106/d.cnki.ghbj.2023.000377.
- [13] Fan Yongdong. *A Survey of Cross-Validation Methods in Model Selection* [D]. *Shanxi University*, 2013.
- [14] Li Zhanshan, Yao Xin, Liu Zhaogeng, et al. *Feature Selection Algorithm Based on LightGBM* [J]. *Journal of Northeastern University (Natural Science Edition)*, 2021, 42(12): 1688-1695.
- [15] Chen Wenbing, Guan Zhengxiong, Chen Yunjie. *Data Augmentation Method Based on Conditional Generative Adversarial Networks* [J]. *Journal of Computer Applications*, 2018, 38(11): 3305-3311.