

## Research Article

# LightGBM Low-Temperature Prediction Model Based on LassoCV Feature Selection

Shangqi Duan,<sup>1</sup> Shuangde Huang,<sup>1</sup> Wei Bu,<sup>1</sup> Xingke Ge,<sup>1</sup> Haidong Chen,<sup>1</sup> Jing Liu,<sup>2</sup>  
and Jiqiang Luo <sup>3</sup>

<sup>1</sup>Kunming Power Supply Bureau of Yunnan Power Grid Co., Ltd., Kunming 650011, Yunnan, China

<sup>2</sup>School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China

<sup>3</sup>Beijing Institute of Spacecraft System Engineering, Beijing, China

Correspondence should be addressed to Jiqiang Luo; [luojiqiang@yeah.net](mailto:luojiqiang@yeah.net)

Received 17 June 2021; Revised 30 July 2021; Accepted 13 August 2021; Published 13 October 2021

Academic Editor: Sang-Bing Tsai

Copyright © 2021 Shangqi Duan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Icing disasters on power grid transmission lines can easily lead to major accidents, such as wire breakage and tower overturning, that endanger the safe operation of the power grid. Short-term prediction of transmission line icing relies to a large extent on accurate prediction of daily minimum temperature. This study therefore proposes a LightGBM low-temperature prediction model based on LassoCV feature selection. A data set comprising four meteorological variables was established, and time series autocorrelation coefficients were first used to determine the hysteresis characteristics in relation to the daily minimum temperature. Subsequently, the LassoCV feature selection method was used to select the meteorological elements that are highly related to minimum temperature, with their lag characteristics, as input variables, to eliminate noise in the original meteorological data set and reduce the complexity of the model. On this basis, the LightGBM low-temperature prediction model is established. The model was optimized through grid search and crossvalidation and validated using daily minimum surface temperature data from Yongshan County (station number 56489), Zhaotong City, Yunnan Province. The root mean square error, MAE, and MAPE of the model minimum temperature prediction after feature selection are shown to be 1.305, 0.999, and 0.112, respectively. These results indicate that the LightGBM prediction model is effective at predicting low temperatures and can be used to support short-term icing prediction.

## 1. Introduction

Evidence from power grid operation shows that wire breaks and tower toppling accidents, caused by transmission line icing, lead to great damage to the transmission lines themselves and also adversely affect the safe and stable operation of the power grid system more generally [1]. Most transmission line icing accidents occur in areas with high small-scale weather variability, which are strongly affected by factors such as temperature, humidity, cold and warm air convection, circulation, and wind [2]. Low temperature is an important cause of transmission line icing. Therefore, accurate prediction of minimum temperature can provide a good basis for short-term transmission line icing prediction. Minimum temperature data are generally based on time

series, and most traditional prediction methods use univariate time series modeling [3]. However, changes in temperature are affected by various meteorological factors, and those that correlate highly with temperature include wind direction, wind speed, and relative humidity [4]. Traditional time series temperature prediction models mainly include multiple linear regression, autoregressive integrated moving average (ARIMA) [3], and gray prediction methods. Accounting for dynamic changes in temperature is difficult because of the prediction effects of the aforementioned methods, and the prediction results generally tend towards average values. Tao et al. [4] proposed a temperature prediction method using a long short-term memory network based on a random forest approach. Niu and others proposed the use of principal component

analysis, a back propagation (BP) neural network, and a radial basis function (RBF) neural network, to establish a temperature prediction model [5]. Although this method considers the influence of multiple meteorological variables on temperature, it does not consider the time series characteristics of those meteorological variables. Jiang [6] proposed an application of a particle swarm-optimized RBF artificial neural network to temperature prediction. Although the structural parameters of the RBF model are optimized by the particle swarm optimization algorithm, problems associated with univariate time series prediction are still encountered, so the prediction accuracy is not high.

Aiming to address the difficulty of traditional prediction methods to learn from large amounts of data, and their inability to fully consider the impact of multiple meteorological variables and their own time correlations on temperature changes, this paper proposes a LightGBM low-temperature prediction model based on LassoCV feature selection. Daily minimum surface temperature data from Yongshan County, Zhaotong City, Yunnan Province (station number 56489) from 2015 to 2019 were selected as the experimental data for prediction. The model first uses autocorrelation to establish hysteresis characteristics. Subsequently, LassoCV is used to measure the importance of the individual variable features, and those that are highly related to minimum temperature are selected as input variables to LightGBM to model the minimum temperature time series data. The experimental prediction results show that this method has stronger learning ability and higher prediction accuracy than traditional methods and long short-term memory (LSTM) networks. Especially when trained on large-scale multivariable meteorological time series data, this model shows high accuracy and a fast training speed, which are beneficial to practical industrial applications. The accurate prediction of low temperature is a prerequisite to accurately predicting icing on power grid transmission lines. The use of this method can improve the accuracy and speed of prediction and provides a sound basis for supporting the production of icing prediction data.

## 2. Principles of the LightGBM

LightGBM is derived from Reference [7] and related open source tools. The light gradient boosting (LGB) model is an efficient implementation of the classic gradient boosting decision tree (GBDT) model. The LGB model handles the classification, regression, and ranking problems in machine learning. GBDT obtains the final answer by combining multiple decision trees and by adding up the results of all the decision trees. This process has been improved to obtain extreme gradient boosting (XGB). The difference between XGB and GBDT is in the way the tree is split and the way the value of the leaf node is determined. The core idea is to conduct a second-order Taylor expansion of the loss function to be fitted by GBDT and to introduce the regular term of the tree intelligently, so that the formula of the second-order Taylor expansion can be simplified and solved analytically. Thus, a new tree splitting method and leaf node value determination method are derived. LGB is further

optimized on the basis of XGB's improvement of GBDT formula. Figure 1 presents a flow chart of the transition from the gradient boosting method to the LGB model.

*2.1. GBDT Model.* The forward addition model is written as

$$f(x) = \sum_{i=1}^M \beta_i h(x; \theta_i), \quad (1)$$

where  $x$  represents the sample and  $h(x; \theta_i)$  represents the  $i$ th base model. Thus, the entire base model is a weighted sum of  $M$  models, so this model is called the forward addition model. The temperature prediction model solves the regression problem. The trained model  $f$  minimizes the loss function  $L$ :

$$\min \sum_{j=1}^N L(f(x_j), y_j) = \min \sum_{j=1}^N L\left(\sum_{i=1}^M \beta_i h(x_j; \theta_i), y_j\right), \quad (2)$$

where  $\{(x_j, y_j)\}_{j=1}^N$  is the training sample and  $\sum_{i=1}^{M-1} \beta_i h(x; \theta_i)$  is assumed to be known during training.  $\beta_M h(x; \theta_M)$  minimizes equation (2); the idea is to let  $\beta_M h(x; \theta_M), y_j$  fit the negative gradient  $g_j$  through the gradient descent algorithm, where the negative gradient  $g_j$  is

$$g_j = -\frac{\partial L(f_{M-1}(x_j), y_j)}{\partial f_{M-1}(x_j)}, \quad (3)$$

and  $g_j$  exists for every sample. Thus, the way a new sample fits  $\{(x_j, g_j)\}$  is converted into a regression problem:

$$\theta_M, \beta_M = \arg \min_{\theta, \beta} \sum_{j=1}^N \left\| g_j - \beta h(x_j; \theta) \right\|^2. \quad (4)$$

After fitting the negative gradient, the final step size must be determined.

$$\rho_m = \arg \min_{\rho} \sum_{j=1}^N L(f_{M-1}(x_j) + \rho h(x_j), y_j). \quad (5)$$

Thus, the final model is

$$f(x) = f_{M-1} + \rho_M \beta_M h(x; \theta_M). \quad (6)$$

When the base learner  $h$  is determined as the cart decision tree, this becomes the GBDT method.

*2.2. From GBDT to XGB.* The lowest-level XGB model is still a forward addition model, but the difference is that it is not optimized by simple gradient descent, but the regular term of the tree and the second-order Taylor expansion are introduced in a way that simplifies the splitting of GBDT and leaf nodes in a single tree value determination. GBDT requires the gradient that corresponds to each sample  $\{(x_j, g_j)\}_{j=1}^N$ , and XGB requires the gradient and Hess value (two steps) that correspond to each sample  $\{(x_j, g_j, h_j)\}_{j=1}^N$ . The splitting of the XGB tree and the determination of the

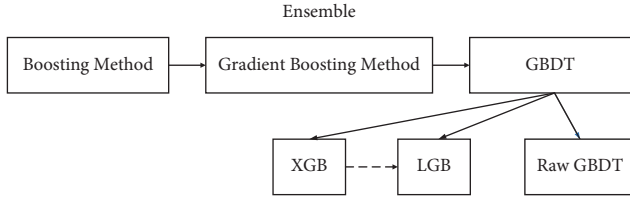


FIGURE 1: Flow chart of the boosting-LGB relationship.

value of the leaf nodes are related to  $(g_j, h_j)$ , which are expressed as follows:

$$\text{Gain} = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma, \quad (7)$$

$$W_j = -\frac{G_j}{H_j + \lambda},$$

where  $\gamma$  and  $\lambda$  are two hyperparameters used to measure the complexity of the tree by adjusting the number of leaf nodes and the weight of the sum of the squares of leaf node values, respectively.

- $G_L$ : sum of  $g_j$  of all samples divided on the left
- $H_L$ : sum of  $h_j$  of all samples divided on the left
- $G_R$ : sum of  $g_j$  of all samples divided on the right
- $H_R$ : sum of  $h_j$  of all samples divided on the right
- $W_j$ : value corresponding to the leaf node
- $G_j$ : sum of  $g_j$  of all samples that are divided in the area of leaf node  $j$
- $H_j$ : sum of  $h_j$  of all samples that are divided in the area of leaf node  $j$

**2.3. LGB Model.** The LGB model is further optimized on the basis of the XGB derivation described above. These optimizations are performed to reduce the computational cost, but they can also play a role in preventing overfitting (because the original data are noisy, some rough processing may increase the generalization ability of the model). The computational cost of each leaf node split is

$$\text{cost}_{\text{time}} = \text{feature}_{\text{num}} \times \text{sample}_{\text{num}} \times \text{point}_{\text{num}}, \quad (8)$$

where  $\text{cost}_{\text{time}}$  represents the time (s) consumed by calculation,  $\text{feature}_{\text{num}}$  represents the number of features,  $\text{sample}_{\text{num}}$  represents the number of samples, and  $\text{point}_{\text{num}}$  indicates the number of candidate points.

Showing that the cost of the tree model is mainly based on three factors:

- (1) Sample size (the cost of calculating  $h_i$  is closely related to the sample size)
- (2) Number of features (features need to be traversed when the tree is split)
- (3) Number of candidate points (the candidate points under this feature need to be traversed when the tree is split)

Therefore, the core of the LGB approach is to minimize the computational cost for each of these factors. The corresponding three technologies are

- (1) Gradient unilateral sampling (GOSS): reduce the number of samples
- (2) Feature binding technology (EFB): reduce the number of features
- (3) Histogram algorithm (Hist): reduce the number of selected points

### 3. Establishing the LassoCV-LightGBM Low-Temperature Prediction Model

**3.1. Data Collection.** This study uses meteorological observation data from Yongshan County (Station No. 56489), Zhaotong City, Yunnan Province, which comes from an Institute of Geographic Sciences and Natural Resources (<https://www.resdc.cn/>) data set, collected daily from January 1, 2015, to December 31, 2019. Part of the original meteorological data is shown in Table 1.

Table 1 confirms that the relevant meteorological variables at Yongshan Station are updated once per day. Data elements include the collected station number (Station\_Id), year (Year), month (Month), day (Day), and eight meteorological variables. They are as follows: minimum temperature (Min\_TEM), average relative humidity (Aver\_RHU), minimum relative humidity (Min\_RHU), average wind speed (Aver\_WIN), maximum wind speed (Max\_WIN), wind direction of maximum wind speed (Max\_WIN\_D), extreme wind speed (Extreme\_WIN), and wind direction of extreme wind speed (Extreme\_WIN\_D). The units and precision of the eight meteorological elements are as follows, respectively: 0.1°C, percentage 1%, percentage 1%, 0.1 m/s, 0.1 m/s, 16 directions (degrees), 0.1 m/s, and 16 directions (degrees).

**3.2. Establishing Hysteresis Characteristics.** With time series, the establishment of lag features is a key step in mining the autocorrelation information of the data. The autocorrelation coefficient (ACF) [8] is usually used to measure the correlation between the current moment  $y_t$  and the  $k$ -order lag  $y_{t-k}$ . The correlation coefficient measures the linear correlation between these two variables as follows:

$$r_k = \frac{\sum (y_t - y_{\text{mean}})(y_{t-k} - y_{\text{mean}})}{\sum (y_t - y_{\text{mean}})^2}, \quad (9)$$

where  $r_k$  represents the correlation between  $y_t$  and its  $k$ -order lag, representative autocorrelation coefficient. The autocorrelation represents the relationship between the values of a time series at different points in time. The ACFs of various meteorological variables used in this study are shown in Figures 2–7.

Variables that are highly correlated with temperature include wind direction, wind speed, and relative humidity [4]. When the ACF is  $r_k \geq 0.3$ , then a medium-strength correlation between the variables exists [9], and this is used as the selection criterion for meteorological variables.

TABLE 1: Original meteorological observation data.

Station_Id	Year	Month	Day	Min_TEM	Aver_RHU	Min_RHU	Aver_WIN	Max_WIN	Max_WIN_D	Extreme_WIN	Extreme_WIN_D
56489	2015	1	1	8.4	59	50	15	27	5	63	5
56489	2015	1	2	7.6	59	46	16	28	5	64	7
56489	2015	1	3	4.2	71	45	8	21	5	52	5
56489	2015	1	4	7.4	72	58	13	32	5	78	5
56489	2015	1	5	7.2	65	53	7	33	6	69	6
56489	2015	1	6	4.3	71	44	9	39	9	76	9
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
56489	2019	12	31	5	68	42	23	54	6	95	5

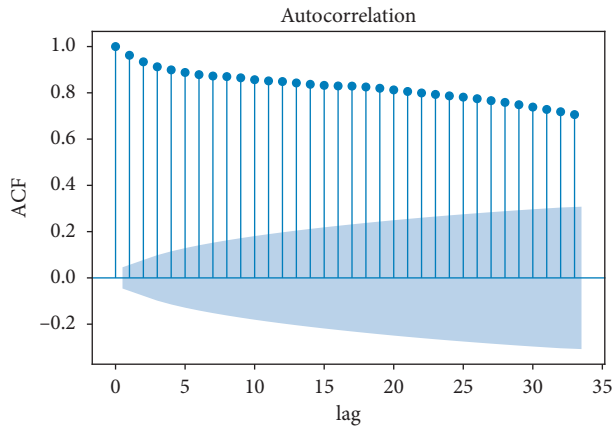


FIGURE 2: ACF of minimum temperature.

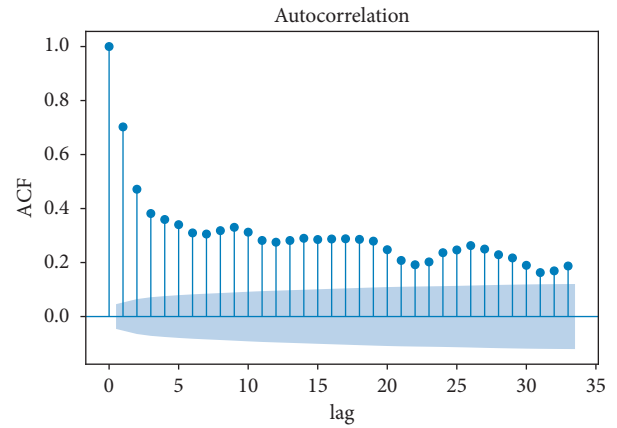


FIGURE 4: ACF of average relative humidity.

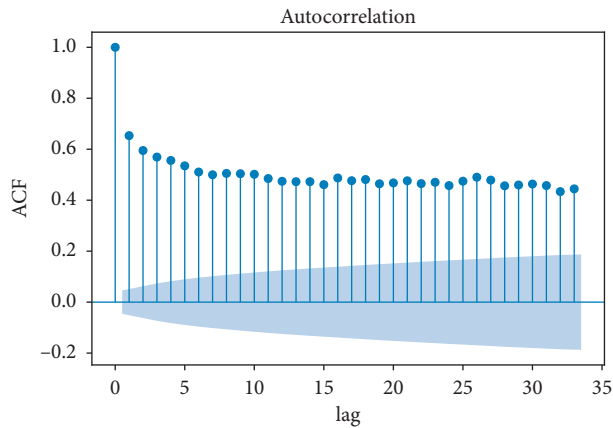


FIGURE 3: ACF of average wind speed.

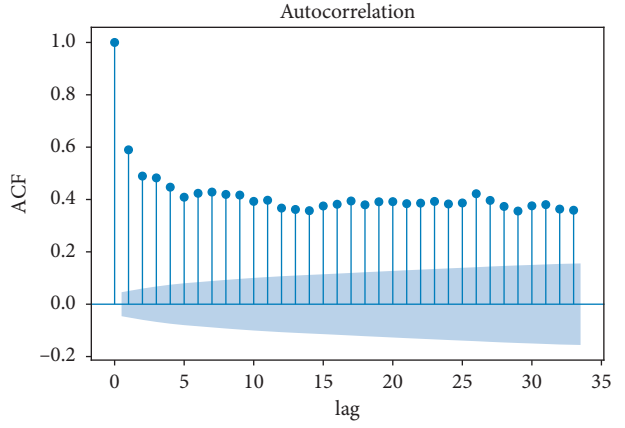


FIGURE 5: ACF of maximum wind speed.

**3.3. Data Preprocessing and Model Construction.** First, the data are cleaned. This study uses the Python environment to process and model the data, using the Pandas fillna function to replace missing and empty values with the average value. The time variables (year, month, and day) are then integrated into a date that can be used as a Pandas index.

Second, the input data are normalized. Table 1 shows that the input meteorological variables have different dimensions and units, so they need to be normalized in order to be able to compare them with each other. The normalization method selected in this paper is min-max scaling, using

$$\mathbf{x}_i^* = \frac{\mathbf{x}_i - \min}{\max - \min}, \quad (10)$$

where  $\mathbf{x}_i$  ( $i = 1, 2, 3, \dots, \mathbf{n}$ ) is the input value,  $\mathbf{x}_i^*$  ( $i = 1, 2, 3, \dots, \mathbf{n}$ ) is the normalized value, max is the maximum value of the variable, and min is the minimum value of the variable.

Given the many noise problems in established methods for evaluating lag characteristics of multielement meteorological time series, and the insufficient prediction accuracy of the traditional ARIMA model, this study proposes a LightGBM low-temperature prediction model based on LassoCV feature selection. First, the ACF is used to establish

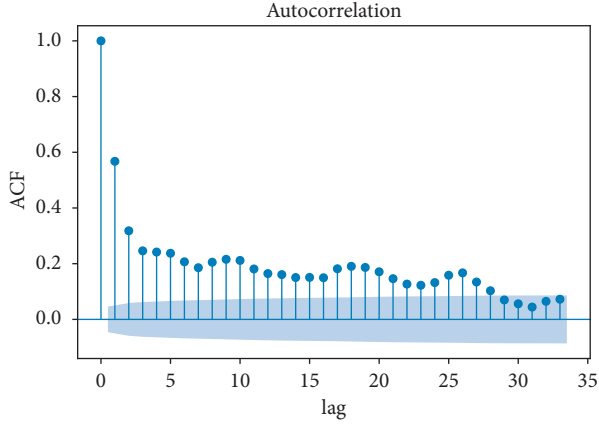


FIGURE 6: ACF of minimum relative humidity.

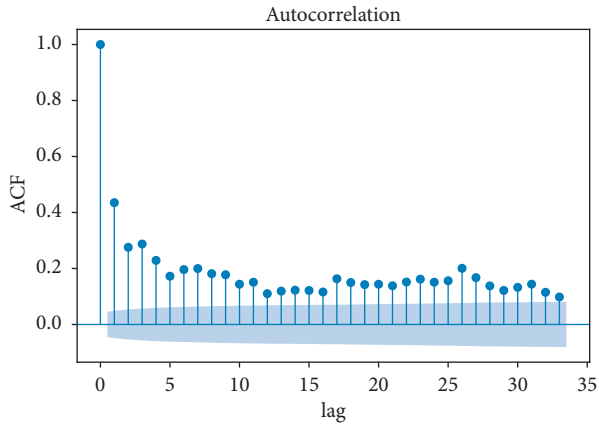


FIGURE 7: ACF of extreme wind speed.

the lag characteristics, and the value of each meteorological element is normalized, that is, the dimensional differences in the multivariate time series data are eliminated, and the processed multivariate data set can be evaluated using a supervised learning method. LassoCV feature selection is then used to measure the importance of each lag feature. The LGB model is next trained on the training data set, and finally the model is optimized through grid search and crossvalidation. Figure 8 presents a flow chart of the model processes.

**3.4. LassoCV Feature Selection.** Selecting features is necessary here because there are many variables and a large amount of noise but some variables have very little effect on minimum temperature. Feature selection is used to select variables that are highly correlated with minimum temperature. The Lasso algorithm performs feature screening to effectively reduce the dimension of multi-dimensional data. The Lasso algorithm [10] is based on linear regression: a threshold is predetermined for the absolute value of the model regression coefficient, and the residual square sum of the model is minimized by adding a normal form function [11]. This algorithm compresses and eliminates the variables whose correlation is less than the threshold by optimizing

the objective function. The remaining variables are then output as the characteristic variables. The results of using LassoCV to select characteristic variables that are highly correlated with minimum temperature are shown in Figure 9.

Set the linear regression model to

$$Y = X^T \beta + \varepsilon, \quad (11)$$

where  $X = [x_1, x_2, \dots, x_i, \dots, x_n]^T$ ,  $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}] \in R^{1 \times m}$  is the low-temperature characteristic data preprocessed by autocorrelation,  $Y = [y_1, y_2, \dots, y_n]^T \in R^{n \times 1}$  is the response variable,  $\beta = [\beta_1, \beta_2, \dots, \beta_m]^T \in R^{1 \times m}$  is the model coefficient, and  $\varepsilon = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]^T \in R^{n \times 1}$  is the error vector. The ordinary least squares method of the linear regression model is estimated to be  $\min[\sum_{i=1}^n (y_i - \sum_{j=1}^m x_{i,j} \beta_j)^2]$ ,  $\hat{\beta}_{ols} = (X^T X)^{-1} X^T Y$ , when the constraint function is added, that is, LASSO, which is specifically expressed as

$$\arg \min_{\beta} \left[ \sum_{i=1}^n \left( y_i - \sum_{j=1}^m x_{i,j} \beta_j \right)^2 + \lambda \sum_{j=1}^m |\beta_j| \right]. \quad (12)$$

In the formula, the parameter  $\lambda$  is the penalty coefficient of parameter estimation, and its size is verified by ten-fold cross-validation, and the parameter  $\alpha$  is determined in the same way. This paper uses the minimum regression method [12] to solve the LASSO regression algorithm. The minimum regression method is a variable selection algorithm based on the forward selection algorithm and the forward gradient algorithm, which can obtain more accurate eigenvectors, which are described in detail as follows:

- (1) The calculation process of the forward selection algorithm is as follows: select the independent variable  $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]$  that is closest to the target variable  $y_k$  in  $X = [x_1, x_2, \dots, x_i, \dots, x_n]^T$ , and there are

$$\bar{y}_k = \mathbf{x}_k \beta_k. \quad (13)$$

Among them, the coefficient  $\beta_k$  is determined by using the following equation:

$$\beta_k = \frac{\langle \mathbf{x}_k, y_k \rangle}{\mathbf{x}_k^2}. \quad (14)$$

The variable residual is

$$y_{res,k} = y_k - \bar{y}_k. \quad (15)$$

The variable residual is defined as a new target variable, and the set  $X$  without  $x_k$  is used as the new independent variable set. Repeat the above process until the residual is less than the set range or the number of independent variable sets is zero, and the algorithm terminates.

- (2) The forward gradient algorithm selects a feature variable  $x_k$  with the most correlation each time to approximate the target variable  $y_k$ . Unlike the forward selection algorithm, the residual is defined as

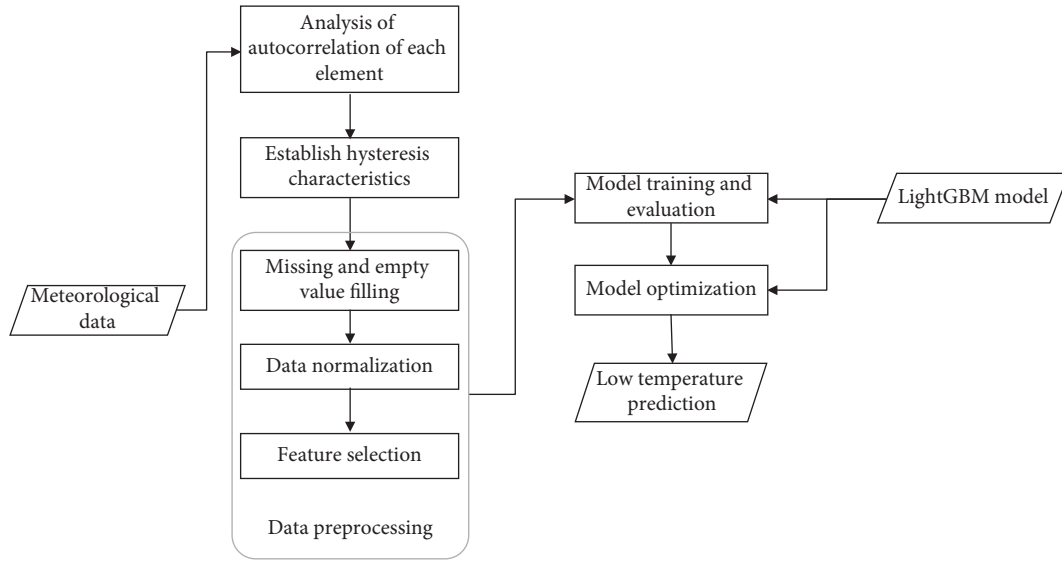


FIGURE 8: Model construction flow chart.

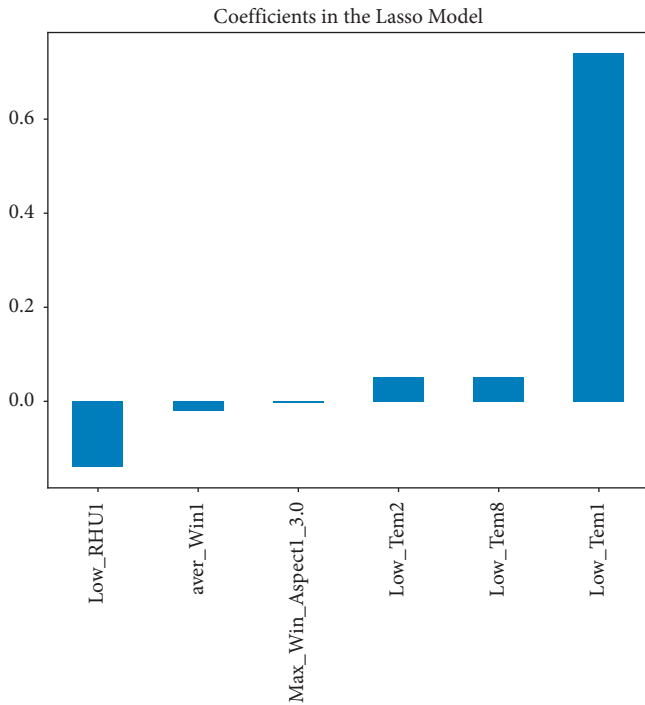


FIGURE 9: LassoCV feature selection results.

$$y_{\text{res},k} = y_k - \mathbf{x}_k \boldsymbol{\beta}_k. \quad (16)$$

Regard the residual as the new objective function, the original variable set  $X = [x_1, x_2, \dots, x_i, \dots, x_n]^T$  as the variable set, and recalculate according to formula (16), until the residual  $y_{\text{res},k}$  less than the set threshold range; the optimal solution is obtained.

The specific steps of the Lasso algorithm are as follows:

**Step 1 (target variable):** According to formulae (13) and (14), solve the variable  $x_k$  with the highest correlation with the objective function, remove it from the variable

set sum, and determine the new target variable according to formula (16).

**Step 2 (related variable):** repeat Step 1 until the correlation between the new variable  $x_l$  and the target variable  $y_{\text{res},k}$  is the same as the correlation between the variables  $x_k$  and  $y_{\text{res},k}$ .

**Step 3 (characteristic variable):** on the angular bisector of  $x_k$  and  $x_l$ , use equation (16) to reapproximate the variable  $x_t$ , so that the correlation between  $x_t$  and  $y_{\text{res},k}$  is the same as  $x_k$ ,  $x_l$ , and  $y_{\text{res},k}$ . The correlation degree is the same, add the variable  $x_t$  to the feature set, and use the common angle bisector of the set as the new approach direction.

**Step 4 (loop):** loop the above process until  $y_{\text{res},k}$  is small enough or the variable set is empty, and the final feature set is the desired feature variable.

The results of using LassoCV to select characteristic variables that are highly correlated with low temperature changes are shown in Figure 9.

Figure 9 shows that, among the meteorological elements and the established time series features, Low\_Tem1, Low\_Tem2, Low\_Tem8, Max\_Win\_Aspect1\_3.0, aver\_Win1, and Low\_RHU1 have higher correlation with minimum temperature than other features; hence, these six feature sets are used as the sample data set.

**3.5. LightGBM Training and Model Tuning.** Low temperature prediction is defined as using a historical meteorological variable sequence  $\{\dots, x_{t-1}, x_t\}$  to predict a future minimum temperature sequence  $\{x_{t+1}, x_{t+2}, \dots\}$ . The preprocessed and feature-selected data sets are input to the LightGBM for training, and then a grid search combined with ten-folds crossvalidation is selected to optimize the main model hyperparameters, including iterations, learning\_rate, max\_depth, and criterion, thereby improving the accuracy of the low-temperature prediction model. After

optimization, the best hyperparameters of the low-temperature prediction model are as in Table 2.

#### 4. Experimental Results and Analysis

To evaluate the accuracy and practicability of the low-temperature prediction model based on the proposed LassoCV-LightGBM combination described in Section 2, this work collected meteorological observation data from Yongshan Station in Zhaotong City, Yunnan Province, to train and test the model. Here, we select the data from the first 1790 days as the training set and the data from the last 56 days as the verification set.

*4.1. Model Evaluation Indicators.* After training the model, the entire verification set is predicted, and then the predicted and observed temperature data are denormalized. To evaluate the performance of the model, this study compares its accuracy with that of the traditional prediction model ARIMA [13] and the LSTM [14], by comparing observed and predicted minimum temperature values. The root mean square error (RMSE), average absolute error (MAE), and average absolute percentage error (MAPE) are selected as the model evaluation indicators. They are calculated as follows:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (y'_i - y_i)^2}, \\ \text{MAE} &= \frac{1}{N} \sum_{i=1}^N |y'_i - y_i|, \\ \text{MAPE} &= \frac{1}{N} \sum_{i=1}^N \left| \frac{y'_i - y_i}{y_i} \right|, \end{aligned} \quad (17)$$

where  $y'_i$  is the predicted minimum temperature value,  $y_i$  is the observed value, and  $N$  is the number of data elements. Smaller values of RMSE, MAE, and MAPE mean a smaller error in the minimum temperature prediction, and therefore a better prediction model performance. Table 3 shows the RMSE, MAE, and MAPE values of the three models.

To further test the superiority and practicability of the proposed LassoCV-LightGBM model, this study randomly selected data from five other sites in Yunnan Province for testing. The verification results are shown in Table 4.

Table 3 shows that the RMSE, MAE, and MAPE values of the LassoCV-LightGBM model are all smaller than those of the LSTM and ARIMA models, indicating that the LassoCV-LightGBM model has higher prediction accuracy and smaller error between the predicted minimum temperature value and the actual observed value.

In summary, the use of the LassoCV-LightGBM model for multivariate time series data with a large amount of data improves not only the prediction accuracy but also the speed of the model training.

TABLE 2: Best hyperparameters of the low-temperature prediction model.

Iterations	Learning_rate	Max_depth	Criterion
780	0.003	9	Gini

TABLE 3: Comparison of the test errors of ARIMA, LSTM, and LassoCV-LightGBM low-temperature prediction models.

Low-temperature prediction model	RMSE	MAE	MAPE
ARIMA	2.132	1.564	0.258
LSTM	1.913	1.452	0.223
LassoCV-LightGBM	1.305	0.999	0.112

TABLE 4: Comparison of test errors of LassoCV-LightGBM low-temperature prediction models at other sites.

Station_Id	RMSE	MAE	MAPE
56959	1.614	1.211	0.117
56952	1.538	1.404	0.134
56836	1.651	1.507	0.152
56641	1.443	1.101	0.124
56483	1.476	1.156	0.125

#### 5. Conclusions

The LightGBM gradient boosting tree integration model is suitable for modeling multivariate time series data. Compared with traditional time series forecasting methods, the unique GOSS, EFB, and Hist of the LightGBM gradient boosting tree integration model can address the problems of high dimensionality, nonlinearity, and local minima more effectively, and it has stronger data learning and generalization capabilities. Moreover, LassoCV can analyze the importance of features and adds regular terms to prevent overfitting. This study establishes lag features through ACFs and then uses LassoCV to select features from multivariate meteorological time series data, which provide more effective and accurate data for model construction and reduce the complexity of the model. Taking minimum temperature data as a specific example, a prediction model based on LassoCV-LightGBM was constructed, and meteorological observation data from the Yongshan site were used for prediction and analysis. The experimental results show that the Lasso-LightGBM model performs better than the ARIMA and LSTM models, with improved low-temperature prediction accuracy, indicating that the Lasso-LightGBM model has superior capability in analyzing multisource time series data. It has particular applicability in predicting low temperatures and is clearly a useful tool for supporting power grid icing prediction.

#### Data Availability

No data were used to support this study.

#### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] L. Qing-Feng, Z. Fan, Q. Wu, J. Gao, Z. Y. Su, and W. J. Zhou, "Investigation of ice-covered transmission lines and analysis on transmission line failures caused by ice-coating in China," *Power System Technology*, vol. 32, no. 9, pp. 33–36, 2008.
- [2] S. Huang and X.-C. Guo, "Research and prospect in micro-meteorological area of transmission lines," *Electric Power Survey & Design*, vol. 38, pp. 1671–9913, 2019.
- [3] C. Narendra Babu and B. E. Reddy, "Predictive data mining on average global temperature using variants of ARIMA models," in *Proceedings of the IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM-2012)*, pp. 75–80, IEEE, Nagapattinam, India, March 2012.
- [4] T. Ye and J. L. Du, "Temperature prediction using long short-term memory network based on random forest," *Computer Engineering and Design*, vol. 151, pp. 1000–7024, 2019.
- [5] Z. Niu and H. Hu, "Prediction model of monthly mean air temperature based on principal component analysis and BP neural network," *High School Science Journal*, vol. 35, no. 11, pp. 6–8, 2015, in Chinese.
- [6] G. Jiang, "The application of PSO-RBF-ANN in temperature prediction," *China Computer & Communication*, vol. 36, pp. 1003–9767, 2020.
- [7] K. Guolin, M. Qi, T. Finley et al., "LightGBM: a highly efficient gradient boosting decision tree," in *Proceedings of the Advances in Neural Information Processing Systems*, Long Beach, CA, USA, December 2017.
- [8] H. G. Sun and H. Deng, "Study on the sample autocorrelation coefficient and partial autocorrelation coefficient," *Journal of Bengbu University*, vol. 49, 2016.
- [9] D. Wang, Z. Cao, and B. Chen, "Multivariate time series local support vector regression forecast methods for daily temperature," *Journal of System Simulation*, vol. 1004-731X, 2016.
- [10] L. Yu-Qiang, P. Tian-Hong, and L. Hao-Ran, "NIR spectral feature selection using lasso method and its application in the classification analysis," *Spectroscopy and Spectral Analysis*, vol. 1000-0593, 2019.
- [11] X. Wu, B. Wu, and J. Sun, "Addition of the Chubak extract and egg white on biophysical properties of grape juice during evaporation process," *Journal of Food Process Engineering*, vol. 40, no. 2, p. 23, 2017.
- [12] A. F. Frank, C. Hlutkowsky, and L. Bemis, "Effect of ultrasound and storage time on quality attributes of strawberry juice," *NeuroImage*, vol. 184, p. 68, 2019.
- [13] L. Sha and H. Lin, "Spatio-temporal modelling and prediction combined with MLR and ARIMA model," *Computer Engineering and Applications*, vol. 183, pp. 1002–8331, 2004.
- [14] M. Zhao, D. Wang, and J. Fang, "Prediction of subway station temperature based on LSTM neural network," *Journal of Beijing Jiaotong University*, vol. 33, 2019.