Problem Chosen	2021 ShuWei Cup Summary Sheet	Team Control Number
B		202111107683

Extreme precipitation event prediction model based on XGBoost

Summary

In recent years, frequent heavy rainfall and snowfall in various parts of China have seriously threatened the lives, safety and property of local people. In the context of global warming, the quantity, intensity, frequency and type of future precipitation in China will be directly affected, and the probability of extreme precipitation will increase significantly. In this context, we have established prediction models of potential extreme precipitation events and quantitative analysis models of losses in different cities in order to make effective predictions of extreme precipitation events, minimize losses and protect people's lives.

For question 1, we first perform data pre-processing, such as dealing with missing time data, null values, etc. Then the daily precipitation data of Zhengzhou city are transformed into annual precipitation data. Immediately afterwards, we analyzed a series of characteristics such as the trend of annual precipitation, precipitation causes, abrupt variability, periodicity, smoothness, precipitation rate, and extreme precipitation rate in Zhengzhou. Finally, we conducted a comparative quantitative analysis of heavy rainfall events in Zhengzhou in two different time dimensions.

For question 2, we obtained daily precipitation data for Beijing, Suzhou, and Guangzhou from 1962 to 2021 from NOAA's National Center for Environmental Information (NCEI). We compared these cities using the relevant attributes from the first question.

For question 3, we want to build a model that predicts and analyzes cities where extreme rainfall is likely to occur in the future. We used three prediction models: BP neural network, ARIMA time series, and XGBoost. in the final model test, although the prediction of BP neural network is good, it is too slow in learning convergence, ARIMA time series is relatively weak in predicting extreme data, and on the whole, XGBoost has the best prediction effect. Finally, we made predictions for the other three cities and judged that Suzhou might have extreme rainfall in the future.

For question 4, we first compared the characteristics of the heavy precipitation in Zhengzhou with those in Shanxi, and then compared the damage they caused to people's property and lives. Based on these comparisons, we also derived several relationships between storm characteristics and losses.

For question 5, based on the conclusions drawn in this thesis and the relevant literature, we propose several long-term construction plans for cities under extreme precipitation conditions in the future.

***Key words*:** Mann-Kendall method；BP neural network；ARIMA；XGBoost；

Content

1. Introduction

1.1 Background

In recent years, heavy rainfall has been frequent, and Henan, Shaanxi and Hubei have suffered from rare heavy rainfall. Such catastrophic rainfall seriously threatens the safety of people's lives and property. In the context of global warming, the amount, intensity, frequency and type of future precipitation in China will be directly affected. Precipitation is expected to increase by about 10% by the end of this century, and the probability of extreme precipitation will increase significantly. Due to the large area of China and the combined influence of various types of topographical features and other factors, the precipitation characteristics of different cities show different features. Therefore, it is imperative to establish prediction models and quantitative loss analysis models for potential extreme precipitation events in different cities.

1.2 Work

Based on the three meteorological stations near Zhengzhou City given in the Appendix and the collected daily precipitation observation data of Shanxi, Beijing, Shanghai and Guangzhou for the past 70 years. In this paper, mathematical modeling is used to solve the following problems.

(1) The annual variation characteristics of precipitation in Zhengzhou are correlated and a number of years with higher precipitation are screened out. At the same time, a specific quantitative analysis of the 2021 Zhengzhou flood event is conducted.

(2) Precipitation data of Shanxi, Beijing, Shanghai, and Guangzhou over many years were collected and compiled to analyze the precipitation trends of these cities.

(3) A precipitation prediction model was developed based on different methods to predict the cities that may experience extreme precipitation in the future.

(4) Compare the characteristics of heavy precipitation in Zhengzhou in July 2021 with the characteristics and losses of heavy precipitation in Shanxi in October 2021.

(5) In-depth analysis of typical cities in China and propose long-term construction plans for cities under extreme precipitation conditions in the future.

2. Problem analysis

2.1 Data analysis

This problem provides the observation data of three meteorological stations near

Zhengzhou for the past 70 years. It includes wind direction, wind speed, gust, maximum temperature, minimum temperature, precipitation, snow depth, FRSHTT and other observations. We found that for the FRSHTT term we need to convert it to the data type first. The original data is a string of binary characters indicating whether one or more of fog, rain, snow, hail, thunder, or typhoon occurred that day. We combined these six items into one, and processed the FRSHTT item into a single binary character representing whether extreme weather occurred that day; then we found that many observations had data that were not observed that day, and we filled this part of the data with zeros. Finally, we found that the precipitation in Zhengzhou should be provided by the data from the three nearby weather stations, so we summed the FRSHTT and PRCP terms separately and then took the average of the three stations.

2.2 Analysis of question one

We used the processed climatic data for Zhengzhou for the annual variability feature correlation analysis. Pearson correlation analysis was used to derive the correlation coefficients between each observation and precipitation in terms of yearly variation, and then the results were analyzed. In order to obtain the variation of precipitation characteristics in Zhengzhou, we make a line graph for the analysis in annual units of variation. The problem requires a specific analysis of the 2021 flood event in Zhengzhou, so we chose to analyze the sudden changes in precipitation in Zhengzhou and the periodic distribution in the time domain using MK mutation and wavelet $analysis$[1].

2.3 Analysis of question two

We downloaded precipitation data from the National Oceanic and Atmospheric Administration (NOAA) for some other cities in China. Based on the geographical location of the cities, we selected precipitation data from three cities, Guangzhou, Beijing and Suzhou, for analysis. After pre-processing the data as described above, we used MK mutation and wavelet analysis to obtain the precipitation trends, mutations, and periodic distributions in the time domain for the three cities. Finally, in order to reflect the differences between them more visually, we summarized the precipitation trends of the three cities into one graph for comparison.

2.4 Analysis of question three

We used the data collected in Problem 2 and selected three methods, BP neural network, ARIMA and XGBoost model, to predict the future extreme rainfall weather in each city according to the requirements of the $problem$[2]. The data provided in this question were first preprocessed: the DATE term and PRCP term were kept after removing irrelevant observations, and then the data from station1 of the three weather stations near Zhengzhou were selected as the data set for these three methods. After

several training sessions, the effects of the three were compared according to the fitting effect. The one with the best effect is finally arrived at, and then the weather data of the cities collected in problem 2 are used to predict the cities with extreme rainfall.

2.5 Analysis of question four

In order to analyze the characteristics of heavy rainfall in Zhengzhou and Shanxi, we used the study of abrupt change points used in the previous problems to obtain the characteristics of the time when abrupt changes in rainfall occurred in Zhengzhou and Shanxi. Then we statistically analyzed the precipitation data from the two weather stations in Zhengzhou and Shanxi to obtain data on the maximum hourly precipitation, the maximum daily precipitation, and the duration of precipitation. Next, we searched to obtain data on the number of casualties, the extent of building damage, the area of crop loss, and the direct economic loss in both cities during the rainstorm. Finally, all the above data were compared by city to get the conclusion whether the characteristics of heavy rainfall and damages caused by Zhengzhou and Shanxi are the same.

2.6 Analysis of question five

Based on the conclusions obtained from question one to question four, we know the connection of storm characteristics on the damage caused by cities. Based on this conclusion, suggestions for long-term construction planning of the city are made.

3. Symbol and Assumptions

3. 1 Symbol Description

Tab.1 Explanation of symbols

Symbol	Meaning
r	Correlation coefficient
s_k	The order column of the time series x
p	The probability value corresponding to the t-statistic
$OBS - Precipitation$	Observed precipitation values Precipitation forecast

3.2 Fundamental assumptions

1、The precipitation data provided in the title is true and correct, that is, there is no large-scale data error at the observation point.

2、The three meteorological stations given in the question do not have high or low weights in the determination of precipitation in Zhengzhou.

3. There are various definitions of extreme rainfall in academia. In this paper, we adopt the judgment criteria as the data in the Chinese precipitation intensity classification standard.

4. Model Building and Model Testing

4.1 Question One

4.1.1 Correlation analysis of the annual variation characteristics of precipitation features in Zhengzhou area

Data pre-processing

Through NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION GLOBAL SURFACE SUMMARY OF DAY DATA (GSOD) (VERSION 7) (ftp://ftp.ncdc.noaa.gov/pub/data/gsod/readme.txt), we reviewed the meaning of each attribute and the rules for handling missing values, (e.g., MXSPD, which means the day the maximum sustained wind speed reported, with a missing value of 999.9). We first used excel's replacement function to replace 999.9 with a null value. Then we use python to perform a control test to get the number of null values for each attribute.

Tab.2 Variable Missing Value Summary

ATTRIBUTE	Number of null values
SNDP	19416
GUST	17671
STP	8734
SLP	22
MXSPD	18
WDSP	9
VISIB	9
TEMP	0
……	……

DEWP	0

Among them, the three attributes SNDP,GUST ,STP are significantly higher than the other items, and after visualizing the data (taking STATION1 as an example), we found that the number of valid data is small and unevenly distributed, so we removed them.

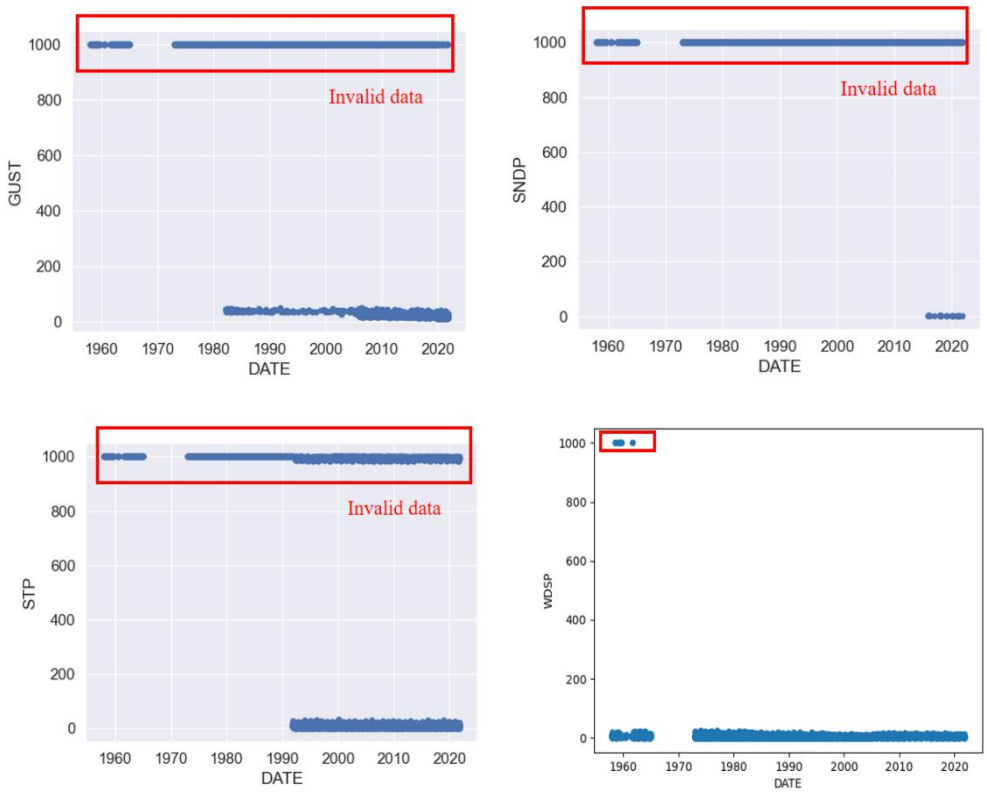


Fig.1 Missing value statistics

In addition, we note that in the early records of the observatory, the statistics are discontinuous (missing data for some time periods), and for ease of calculation, we only count the records of three observatories after January 1, 1962.

Transformation of daily precipitation data to annual precipitation data

First, we weighted the data from the three observation stations to average.

Second, we transform the daily precipitation data into annual precipitation data because we need to analyze the annual variation of precipitation characteristics related to the Zhengzhou area. Our method is to use excel's own pivot table for processing, and in the transformation, we handle each attribute differently according to the nature of the attribute itself.

Tab.3 Variable handling methods

ATTRIBUTE	Processing method
DEMP	averaging
FRSHTT	summation
MAX	maximizing
MIN	minimizing

PRCP	summation
SLP	averaging
TEMP	averaging
VISIB	averaging
WDSP	averaging
MAXSP	averaging

Correlation analysis of the annual variation characteristics of precipitation features

(1) Analysis of precipitation trends

Fig.2 Zhengzhou precipitation trend

The following conclusions can be drawn from the images.

1) Since 1960, the annual precipitation in Zhengzhou has basically ranged from 10cm - 40cm, fluctuating around 25cm or less.

2) Since 2000, there is a more obvious trend of increasing precipitation in Zhengzhou than before.

3) Zhengzhou had low precipitation in 1998 and high precipitation in 2003, 2016 and 2021.

(2) Analysis of precipitation causes

Correlation analysis refers to the analysis of two or more elements of variables with correlation, so as to measure the closeness of the correlation between two factors.

$$r = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}}$$

The correlation coefficient r takes values in the range $-1 \le r \le 1$.

$$\begin{cases} r > 0 \text{ Positive correlation}, r < 0 \text{ Negative correlation} \\ |r| = 0 \text{ no linear correlation} \\ |r| = 1 \text{ linear correlation} \end{cases}$$

$0 < |r| < 1$ indicates the presence of different degrees of linear correlation.

Correlation analysis was performed by python to generate correlation heat maps, and it was found that there were no attributes that showed strong correlation with precipitation.

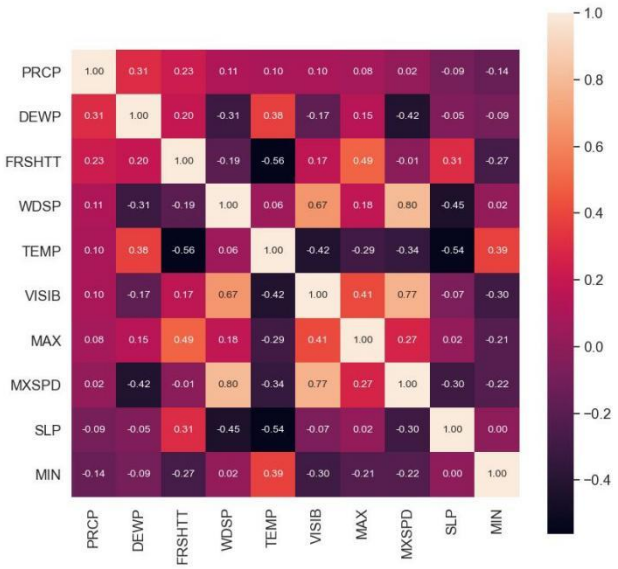


Fig.3 Variable correlation coefficient heat map

(3) Analysis of sudden variability of precipitation

The MK test is the Mann-Kendall method, which is a climate diagnosis and prediction *technique*[3]. The Mann-Kendall test can be applied to determine whether there is a sudden climate change in the climate series. The Mann-Kendall test is also often used to detect trends in the frequency of precipitation and drought under the influence of climate change.

For a time series X with n sample sizes, an order column is constructed.

$$s_k = \sum_{i=1}^{k} r_i \quad r_i = \begin{cases} 1 & x_i > x_j \\ 0 & \text{else} \end{cases} \quad j = 1, 2, \cdots, i$$

In Eq. $UF_1 = 0$, $E(s_k)$, $Var(s_k)$ are the mean and variance of the cumulative counts, and when $x_1, x_2, \ldots, x_n$ are independent of each other and having the same continuous distribution, they can be calculated by the following equation.

$$E(s_k) = \frac{n(n+1)}{4} \quad \mathrm{Var}(s_k) = \frac{n(n-1)(2n+5)}{72}$$

UF_i is the standard normal distribution, which is a sequence of statistics calculated in the order of the time *series* x[4]. Given the significance level α, check the normal distribution table, if $|UF_i| > U_a$, it indicates that there is a significant trend change in the series.

The above process is repeated again in the reverse order of the time series x, while making $UB_{k=}{}^{-}UF_{k}$, k=n,n-1,...,1),$UB_{1} = 0$. The advantage of this method is that it is not only easy to calculate, but also can specify the time when the mutation starts and indicate the mutation region. Therefore, it is a commonly used mutation detection method.

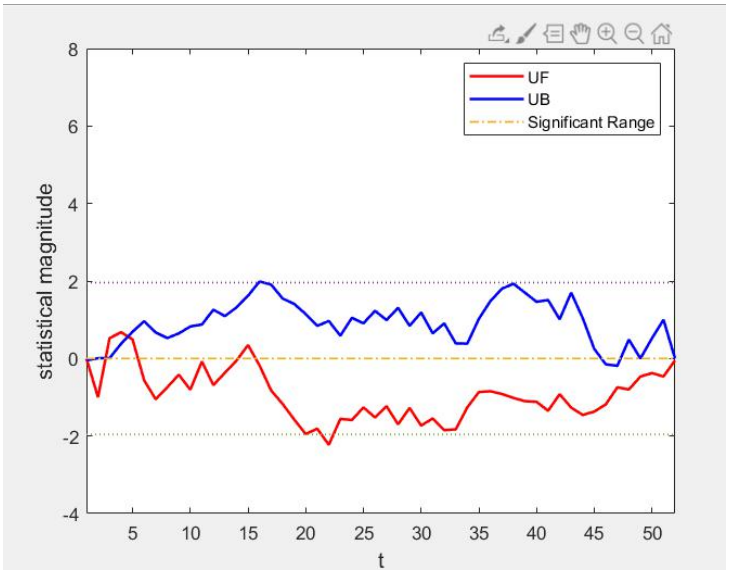


Fig.4 MK Inspection of Zhengzhou

We used matlab to perform the calculation and we can see that the mutation occurs in the SIGNIFICANT range at t=5(1967) and t=51(2013).

(4) Analysis of precipitation periodicity

A wavelet analysis with time-frequency multi-resolution function provides the possibility to better study the periodicity problem, which can clearly reveal the multiple cycles hidden in the time series, fully reflect the trend of the system in different time scales, and qualitatively estimate the future development trend of the system. We choose wavelet analysis to analyze the periodic variation of precipitation in Zhengzhou area in different time scales and its distribution in the time domain.

The wavelet analysis toolbox in matlab was first used to wavelet coefficientsize the precipitation. The real part, imaginary part, and variance were obtained as follows.

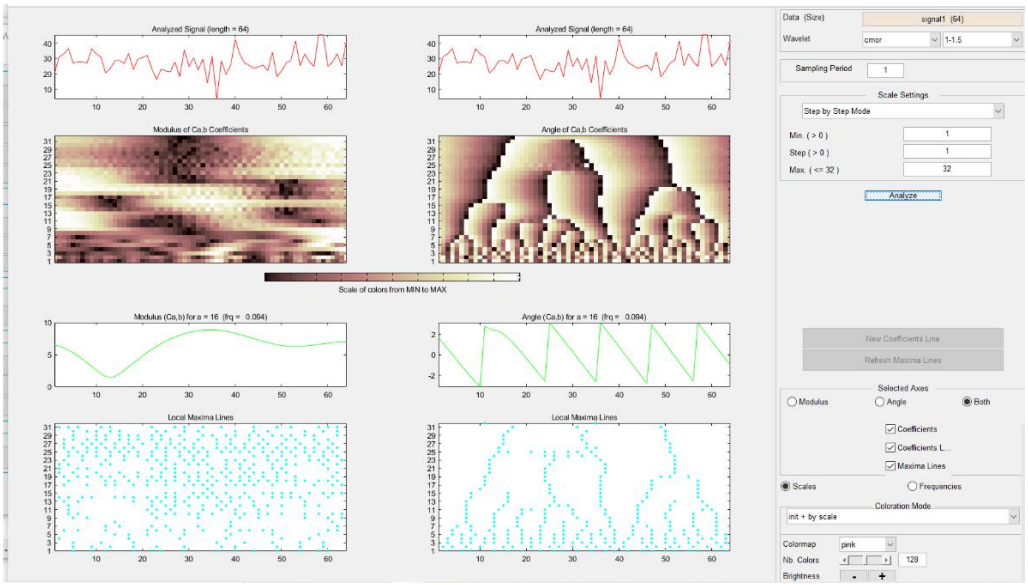


Fig.5 Wavelet analysis factorization

Because the wavelet coefficient real contour map can reflect the periodic variation of precipitation at different time scales and its distribution in the time domain, we can judge the future trend of precipitation at different time scales. Therefore, we exported the real wavelet coefficients to the software suffer and used the map style "contour map" to obtain the following contour map of the real wavelet coefficients.

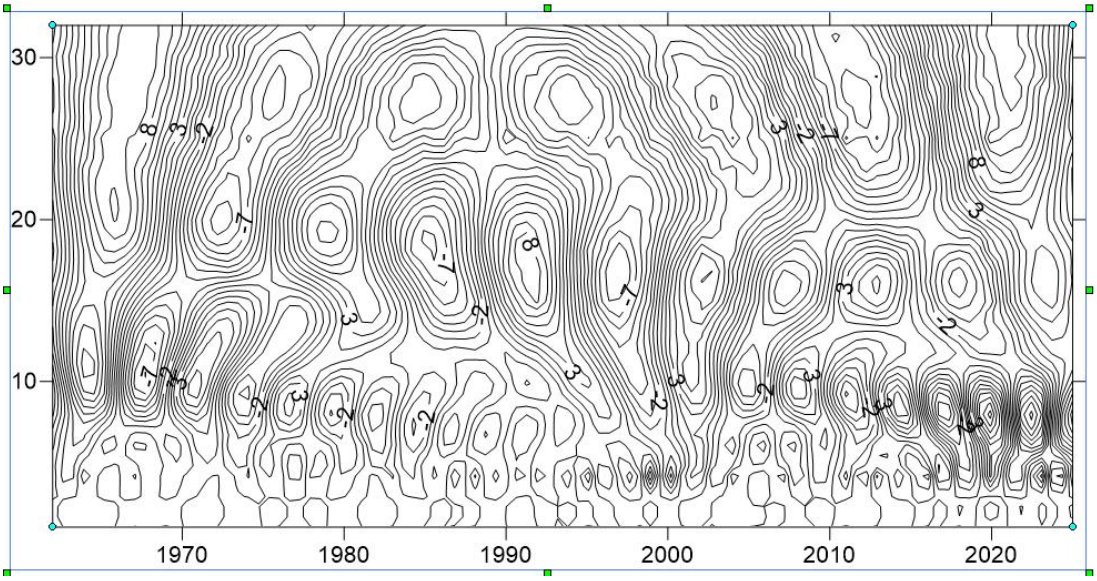


Fig.6 Wavelet analysis cycle

The vertical coordinate is the time scale, and the contour curve in the graph is the wavelet coefficient real part value. From the figure, we can see that there are three types of periodic patterns in the precipitation evolution: 25~32 years, 16~24 years, and 7~15 years. The 25~32 and 16~24 years time scale cycles are very stable throughout the analysis period and are global in nature, while the 7~15 years scale cycles are more stable during the period from 1962 to 1985 and after 2010.

（5）Analysis of Other characteristics

①Smoothness analysis

The ADF (Augmented Dickey-Fuller) test can be used to test the hypothesis of the smoothness of a time series. If the test result is statistically significant, the series can be considered to satisfy the smoothness. ADF test can be implemented in python using the tool statsmodels.

We performed the smoothness test on the annual precipitation data of Zhengzhou and got p=0.008<0.5, so we consider this curve as a smooth curve.

②Precipitation rate

We used matlab to count the number of days that precipitation (not zero after removing the null) occurred in Zhengzhou, and thus calculated the precipitation rate.

Number of precipitation days: 4639

Precipitation rate: 0.239

③Extreme precipitation rate

According to the Chinese precipitation intensity classification standard, we calculated the number of days of extreme rainfall using matlab.

Number of extreme rainfall days: 41

Dividing the total number of days by up, we can get the extreme rainfall in Zhengzhou from 1962 to 2021.

Extreme rainfall rate: 0.0021

Finally, we selected several time points with high annual precipitation and they are 1964, 1992, 2003, 2016, and 2021.

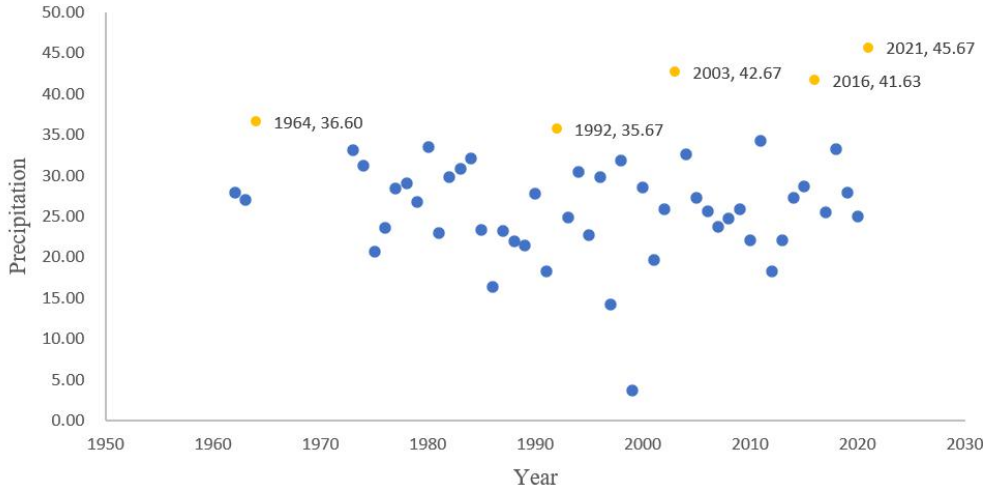


Fig.7 Scatterplot of precipitation

4.1.2 Specific quantitative analysis of the 2021 Zhengzhou flood event

We visualize the data in order to visualize the flood events in Zhengzhou compared to other time periods, and we compare them in two units: "days" and "years". We also use python's describe() function to perform a specific quantitative analysis of the flood events.

(1) Comparing five days before and five days after

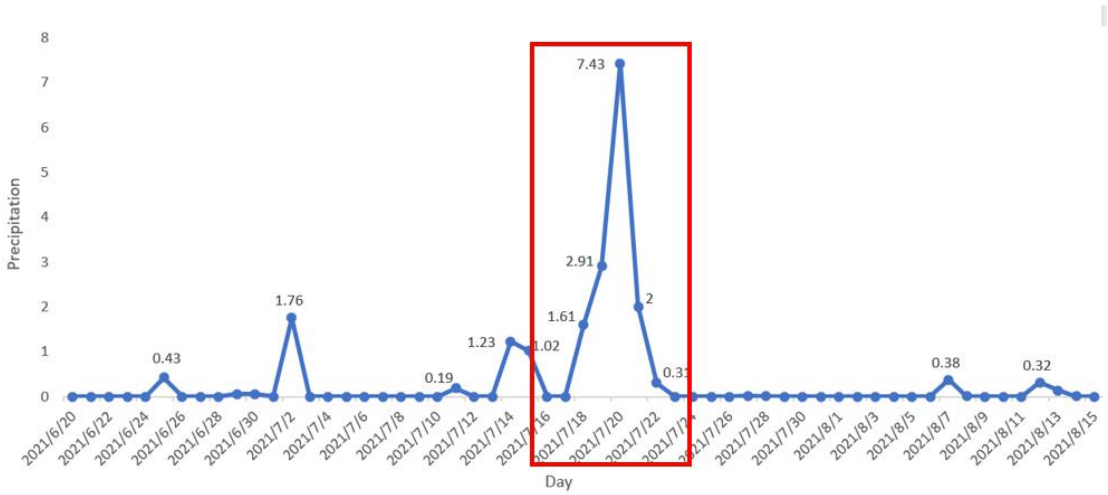


Fig.8 Precipitation folding line graph

Tab.4 Comparison of 5-day precipitation before and after extreme rainfall

ATTRIBUTE	7.13——7.17	7.18——7.22	7.23——7.27
COUNT	5	5	5
MEAN	0.490	2.850	0.004
STD	0.590	2.720	0.009
MIN	0.000	0.310	0.000
MAX	1.230	7.430	0.020

(2) Comparing 2019 and 2020

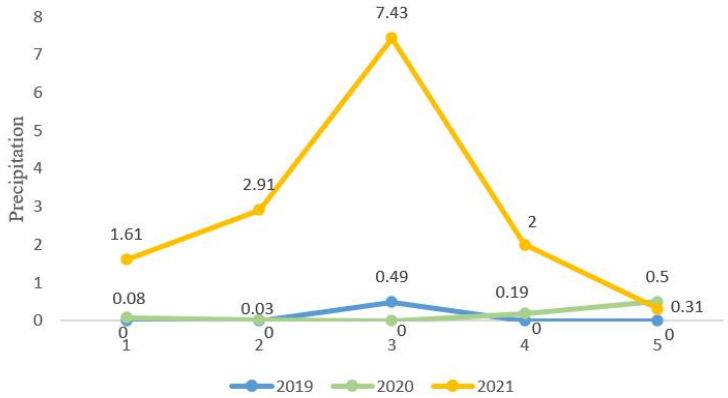


Fig.9 Three-year precipitation comparison chart

Tab.5 Three-year precipitation comparison table

ATTRIBUTE	2019	2020	2021
COUNT	5	5	5
MEAN	0.098	0.160	2.850
STD	0.219	0.203	2.720
MIN	0.000	0.000	0.310
MAX	0.490	0.500	7.430

4.2 Question Two

4.2.1 Data search method

NOAA's National Center for Environmental Information (NCEI) manages and makes available to the public one of the most important archives of environmental data on the planet. It provides more than 37 petabytes of comprehensive atmospheric, coastal, oceanic and geophysical data.

So, in order to obtain more data on precipitation in Chinese cities, we downloaded data from the official website of NCEI First, we selected the area where the weather station was located and set the start and end times of the data we were looking for.

(http://data.cma.cn/Market/Detail/code/A.0012.0001/type/1.htm).

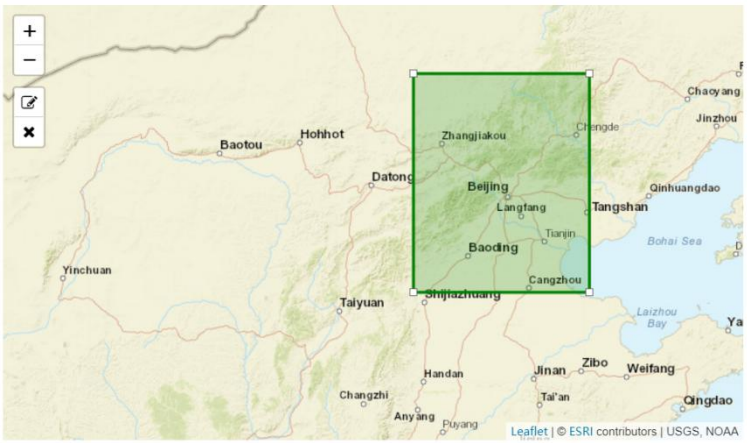


Fig.10 City data collection process diagram

In order to obtain data from 1962 to the present, we sent requests for data from several observation sites to NCEI. In this question, we will analyze the annual precipitation trends at Beijing, Suzhou, and Guangzhou.

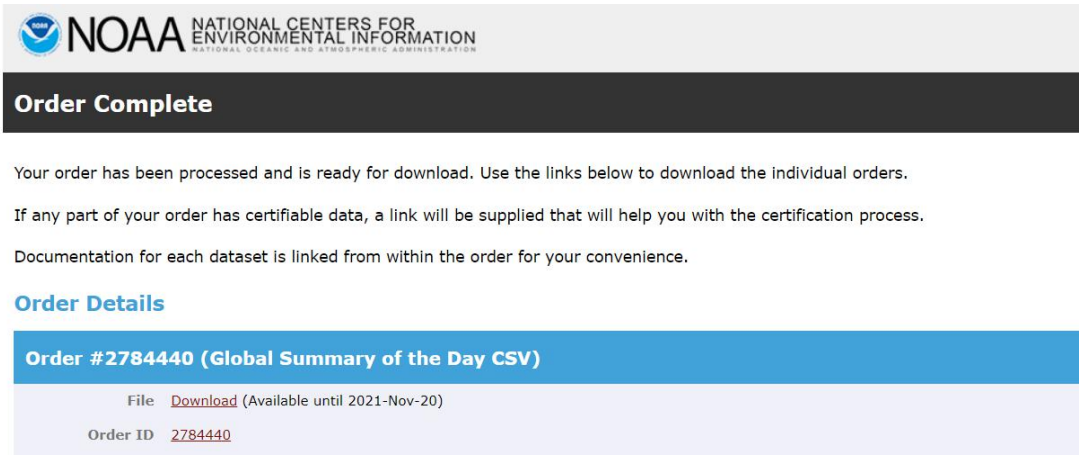


Fig.11 Data download process diagram

4.2.2 Analysis of precipitation trends in Beijing, Suzhou and Guangzhou

（1）Analysis of precipitation trends, number of days of rainfall, and extreme precipitation

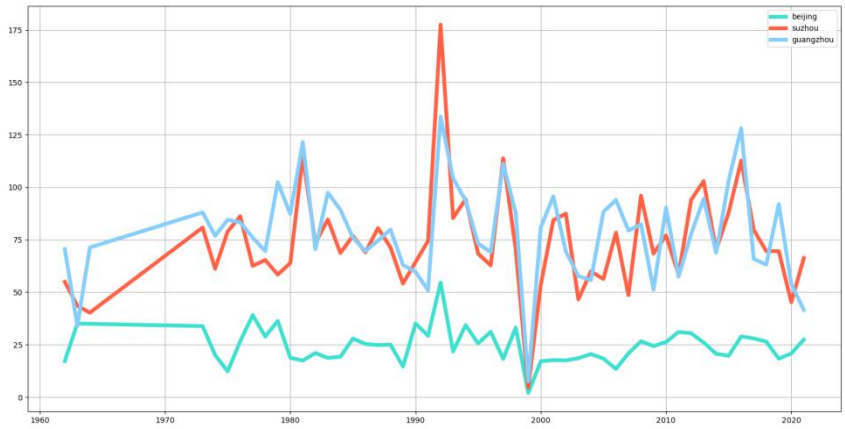


Fig.12 Precipitation trends in three cities

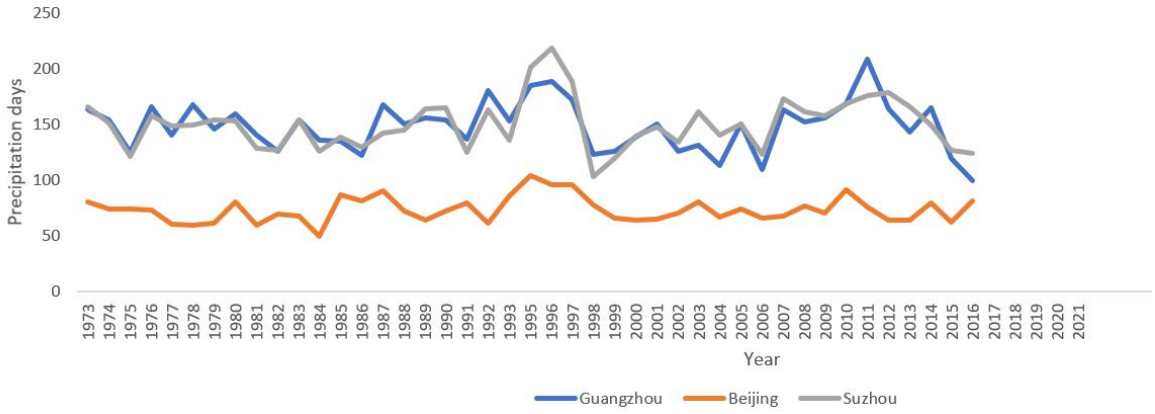


Fig.13 Number of precipitation days in three cities

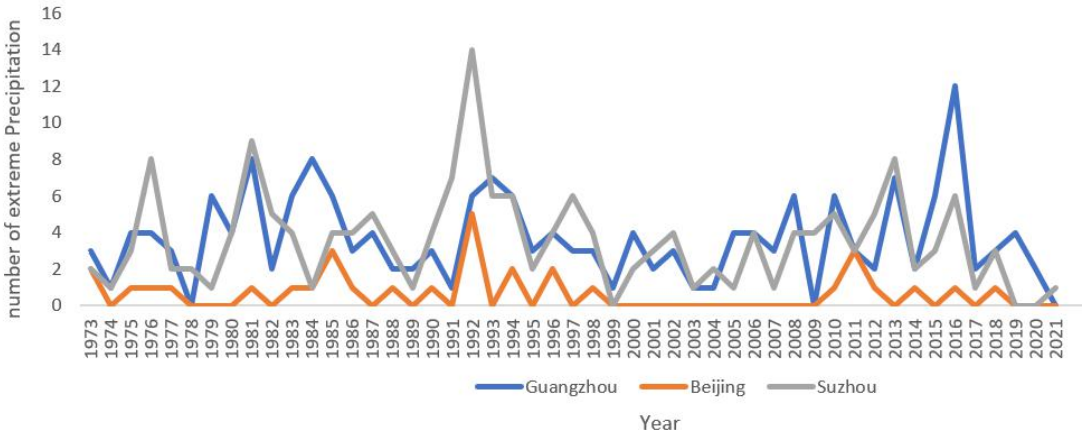


Fig.14 Extreme precipitation in three cities

From the above figure, it can be seen that:

(1) Suzhou and Guangzhou have similar precipitation, with annual precipitation all mainly distributed in the interval [50cm, 125cm], and Beijing has less precipitation overall than Suzhou and Guangzhou, with annual precipitation mainly distributed in the interval [15cm, 45cm].

(2) From 2000 onwards, the annual precipitation in Suzhou and Guangzhou tends to rise significantly, while that in Beijing rises slowly.

(3) All three places have sudden change values in 1992, 1999 and 2016 at the same time.

4) The number of annual extreme precipitation and the number of days of annual precipitation in all three places have similar trends of change - "rising and falling at the same time".

（2）Mutation test analysis

Beijing：

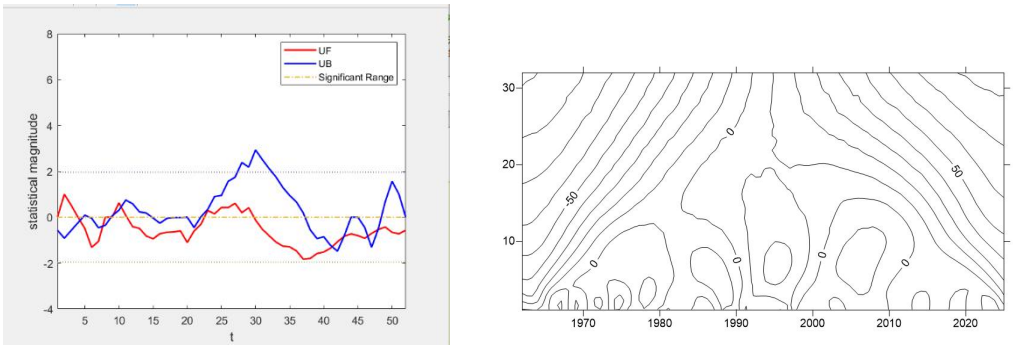


Fig.15 Beijing MK test chart, wavelet cycle chart

Suzhou：

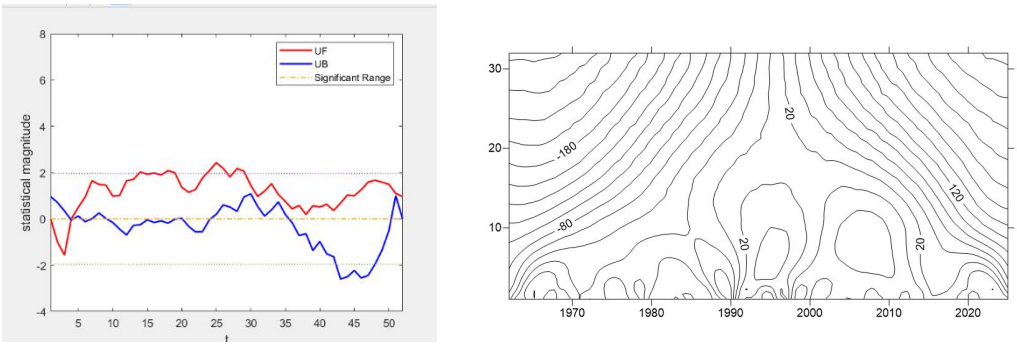


Fig.16 Suzhou MK test chart, wavelet cycle chart

Guangzhou：

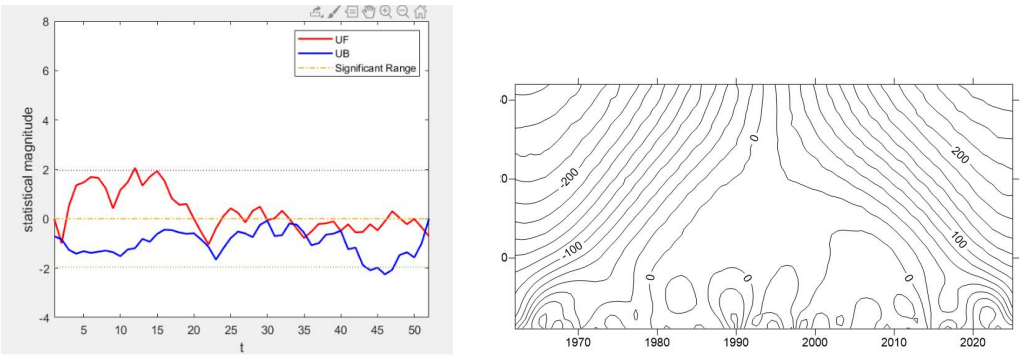


Fig.17 Guangzhou MK test chart, wavelet cycle chart

From the above figure, it can be seen that:

precipitation in Guangzhou has abrupt changes in the significant range of t=1.5, t=22, t=32.5, t=41, t=51, and there are periodic variations in the evolution of precipitation on the 3-8 year scale;

precipitation in Beijing is at t=3, t=7, t=9, t=10, t=41. 5, t=43.5, t=46, t=47.5; the evolution of precipitation on the 3-10 year scale with periodic variation; the evolution of precipitation on the 3-10 year scale has periodic variation.

precipitation in Suzhou: abrupt changes in the significant range of t=4.5; there is a cyclic pattern in precipitation evolution on the 3-10 year scale . In the evolution of precipitation, there is a cyclical variation pattern on the scale of 3 to 10 years, and this scale is relatively stable from 1990 to 2010.

4.3 Question Three

In this question, we used three prediction methods and chose the optimal solution based on the fitting effect, thus predicting the cities where extreme rainfall is likely to occur.

4.3.1 Forecasting based on BP neural networks.

BP neural network is a multilayer feedforward neural network trained according to the error back propagation $algorithm$[5] . It consists of two processes: forward propagation of the signal and backward propagation of the error, i.e., calculating the error output in the direction from the input to the output, while adjusting the weights and thresholds in the direction from the output to the input. In forward propagation, the input signal acts on the output node through the implied layer and undergoes a nonlinear transformation to produce the output signal; if the actual output does not match the desired output, it is transferred to the backward propagation process of the error. The error back-propagation is to back-propagate the output error through the hidden layer to the input layer layer by layer, and to apportion the error to all units in each layer, using the error signal obtained from each layer as the basis for adjusting the weights of each $unit$[6]. By adjusting the connection strength of the input nodes

to the hidden layer nodes and the connection strength of the hidden layer nodes to the output nodes and the threshold value, the error decreases in the gradient direction, and after repeated learning training, the network parameters (weights and thresholds) corresponding to the minimum error are determined and the training is stopped. At this point, the trained neural network is able to process the input information of similar samples and output the non-linear transformed information with the minimum error by itself.

In this question, since the prediction of precipitation in a city requires the prediction of the future based on existing historical data. And through the study of the first question, we know that the randomness of precipitation is very strong and the inherent law is difficult to grasp. So we choose dynamic time series based on BP neural network for prediction.

The specific process is as follows: we choose the BP neural network with memory function.

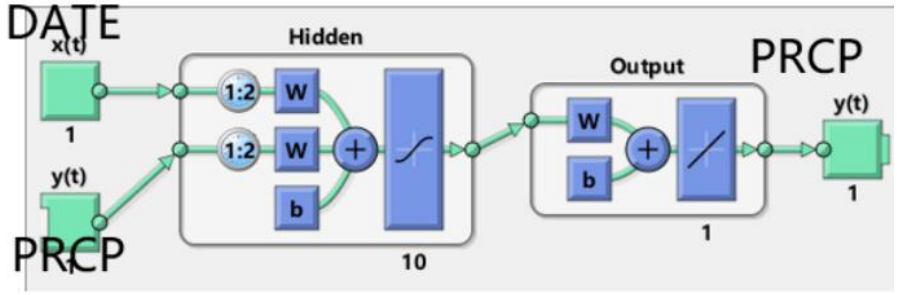
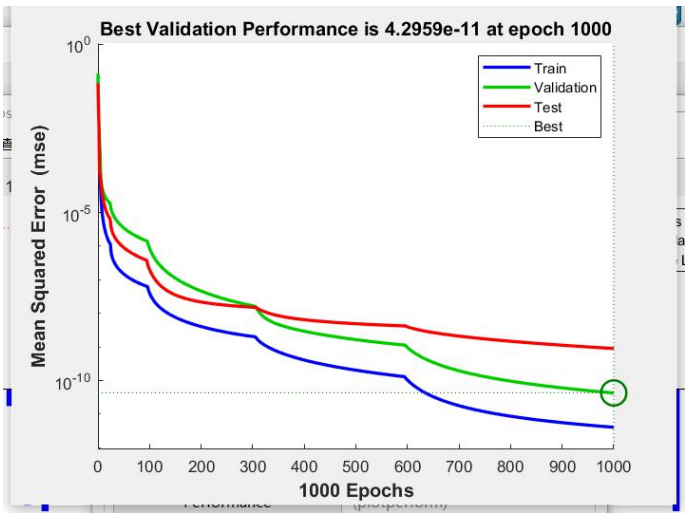


Fig.18 Neural network structure diagram

After several times of debugging and modifying parameters, the effect is as follows.



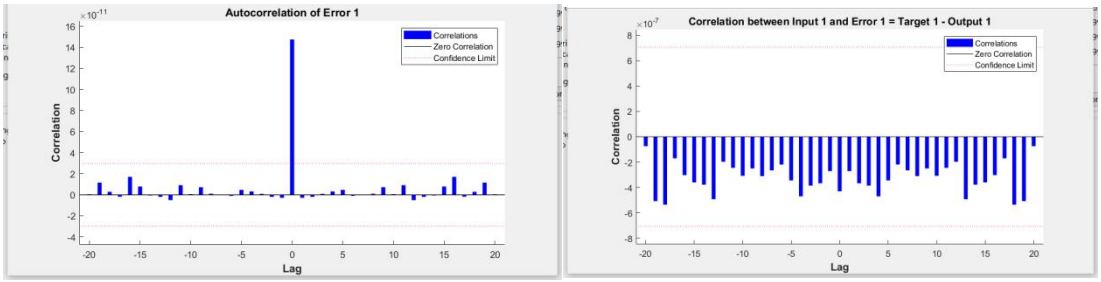


Fig.19 Neural network rendering

It can be seen that in the autocorrelation of error and in the correlation graph, correlations are in the confidence limit, indicating that this neural network is used to predict better.

4.3.2 Forecasting based on ARIMA Time Series

The ARIMA model, known as the Autoregressive Integrated Sliding Average model, is the most commonly used model for time series forecasting. The annual precipitation is a random series that varies over time, and the BIC criterion is used to determine the optimal value of (p, q) in the ARIMA model to construct the precipitation regression model, which mainly combines the trivariate method for regression forecasting.

The ARIMA model uses a random sample data series to construct a regression model for water loss, which has been used in many regional water loss forecasting applications with good results.

This paper uses the ARIMA model to simulate the historical precipitation trends in Zhengzhou, to build an optimal precipitation forecast model, and to forecast the precipitation trend in 2021 and compare it with the actual precipitation to verify the feasibility and applicability of the model.

Step 1: Plotting the series of Zhengzhou historical precipitation data, it is found that the series fluctuates within a fixed range on both sides of the mean, and the smoothness is good.

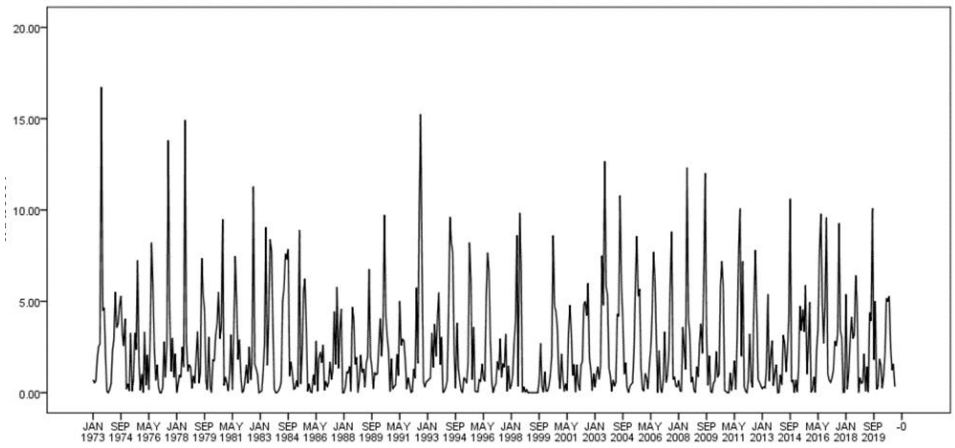


Fig.20 Precipitation sequence diagram

Step 2： Determine the parameters of the model, the autoregressive order p, the

number of differences d, and the moving average term q. The autocorrelation and bi-correlation of the monthly precipitation are analyzed. As can be seen from the figure, the autocorrelation coefficients of the annual precipitation data series of Zhengzhou city decrease slowly to zero after the first order of difference processing. According to the criterion of minimum BIC as the ideal order, the better model ARIMA (1,1,1) was finally obtained after several simulations.

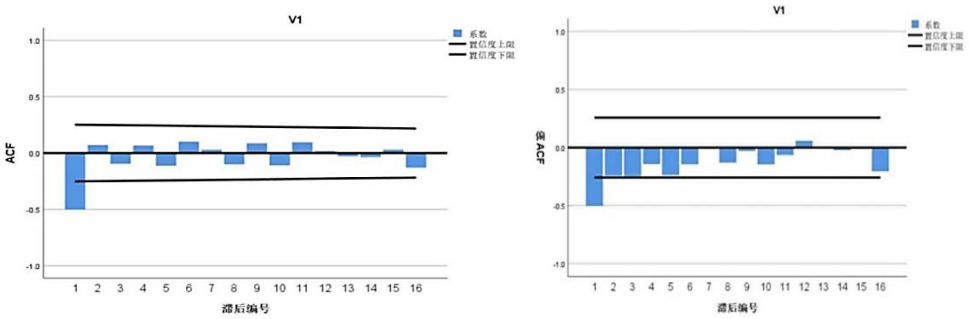


Fig.21 BIC coefficient detection

Step 3: Simulate the annual precipitation in Zhengzhou in 2021, and fit the simulated values to the real values and analyze the residual series.From the images, it can be found that the model prediction is good, but the prediction for the mutation point is not accurate.

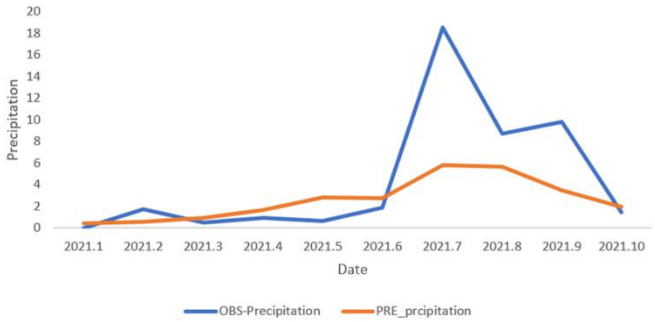


Fig.22 Fitting between observed and predicted precipitation in 2021

4.3.3 Forecasting based on XGBoost.

XGBoost is a Boosting algorithm that belongs to the Integrated Learning Model and has the most efficient performance in supervised learning tasks such as classification, regression and $ranking$[7]. The algorithm has become a preferred tool for machine learning, first and foremost because of its excellent predictive performance, highly optimized multi-core processing and distributed machine implementation, and ability to handle sparse data. It is essentially a class of boosted tree models that integrate many weak classifiers to form a strong classifier in an iterative fashion. One of the decision tree training process of XGBoost is shown in Fig.

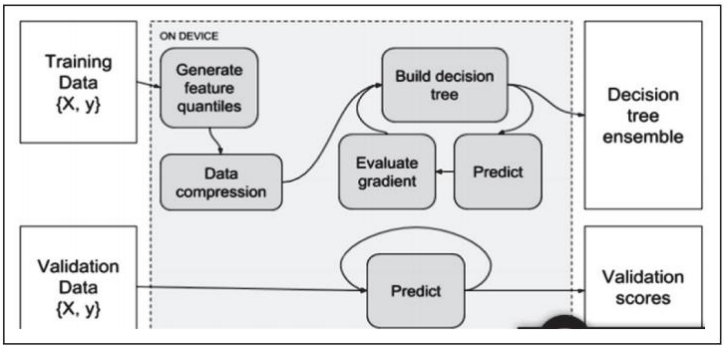


Fig.23 XGBoost training process

The core idea of the algorithm is to keep adding trees to a tree and performing feature splitting to make the tree grow. Each tree that is added corresponds to the model is learning a new function to fit the residuals of the previous prediction. After we have trained N trees, the value we want to predict is essentially is the sum of the corresponding scores of each tree multiplied by the corresponding weight, which is the sample predicted $value$.[8] For the objective function, we should use the second-order Taylor expansion to optimize the The objective function is as follows

$$J(f_i) = \sum_{i=1}^{n} L(y_i, \widehat{y_i^{t-1}} + f_t(x_i)) + \Omega(f_t)$$

n is the total number of samples, $\widehat{y_i^{t-1}}$ is the prediction of the t-1th learner for sample i $f_t(x_i)$ is the newly added t th learner, $\Omega(f_t)$ is the canonical term, and L is the loss function, i.e., the error of the model.

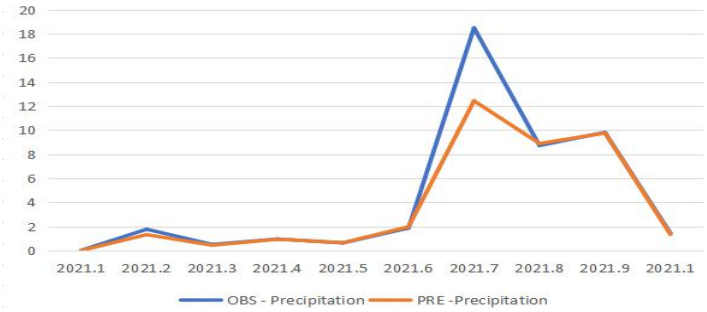


Fig.24 Fitting between observed and predicted precipitation in 2021

Comparing the prediction dataset generated after training the XGBoost model with the observed values, we can find a very good fit of the data. In addition, thanks to the cache-compressed sensing algorithm designed by xgboost, the efficiency and accuracy of prediction are greatly improved.

4.3.4 Model selection and prediction of extreme precipitation cities.

The BP neural network algorithm has good prediction effect, but when we use it,

we find that it converges slowly, especially in the prediction of a large amount of data: because the BP neural network algorithm is essentially a gradient descent method, the objective function it wants to optimize is very complex, so it is bound to appear the "sawtooth phenomenon", which makes the BP algorithm inefficient. This makes the BP algorithm inefficient; and because the optimized objective function is very complex, it will inevitably appear some flat areas where the neuron output is close to 0 or 1. In these areas, the weight error changes very little, making the training process almost stop. Therefore, in fact the BP neural network algorithm is not suitable for making long-term predictions of extreme precipitation.

The ARIMA model predictions work well for overall trends, but poorly for extreme values, which in application may lead to small predictions of precipitation magnitudes, and thus protection against large precipitation may not be ideal.

In summary, the XGBoost algorithm achieves satisfactory training results in terms of prediction accuracy, arithmetic power, and approximation of extreme values. Thus, we use XGBoost to predict extreme precipitation data for Beijing, Guangzhou, and Suzhou for the next five years.When we get the rainfall data of the three places for the next five years. We find the probability of their extreme rainfall occurrence.

Beijing	Guangzhou	Suzhou
3.132	15.285	34.429

4.4 Question Four

4.4.1 Comparison of the characteristics of the two rainstorms

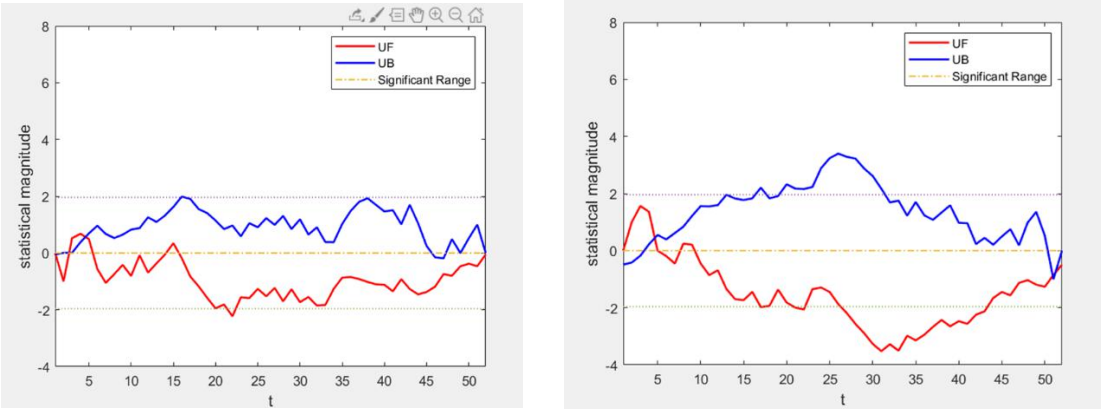


Fig.25 MK test mutation points in two places

During the statistical year, Zhengzhou experienced a total of two sudden changes in annual rainfall, the most recent of which was a major rainstorm in July this year.

Maximum precipitation per hour 201.9 mm

Maximum precipitation of 552.5 mm in a single day

Precipitation lasted for more than 40 hours

4.4.2 Comparison of damage

During the statistical year, there were two sudden changes in the annual rainfall in Shanxi. The most recent one was a major rainstorm in October this year.

Maximum precipitation per hour 57.1 mm

Maximum cumulative urban precipitation of 285.2 mm

Precipitation lasted for more than 90 hours

4.4.2 Comparison of damage from two rainstorms

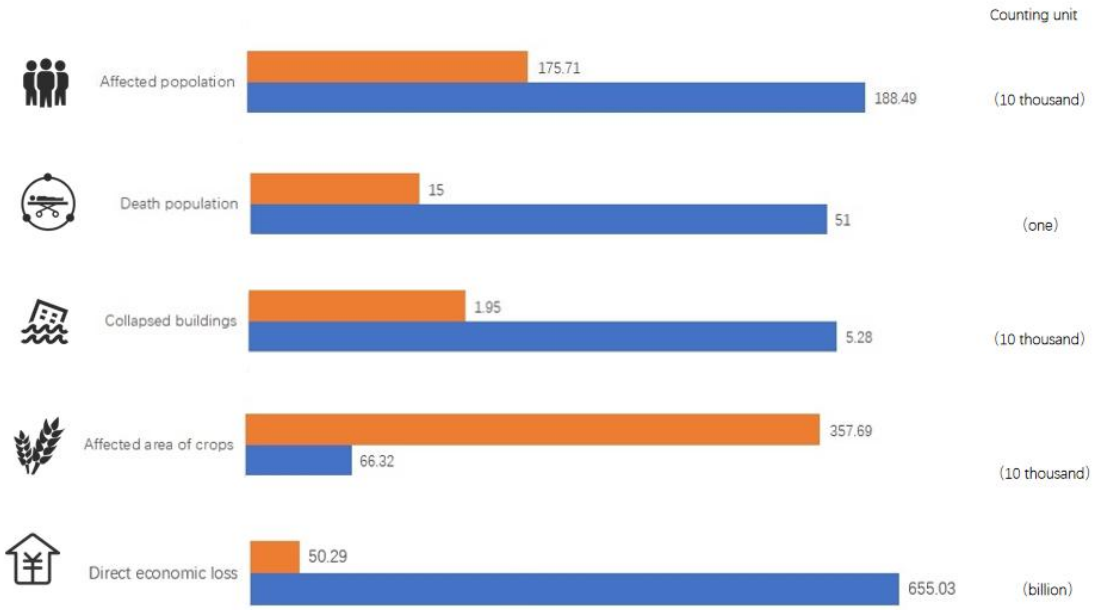


Fig.26 Comparison of losses between the two places

Data from People's Daily, Sina Weibo, etc.

4.4.3 Summary

The maximum precipitation per hour and the maximum precipitation per day in Zhengzhou are much larger than those in Shanxi, but the precipitation time in Shanxi is twice as long as that in Zhengzhou. Therefore, the heavy rainfall in Zhengzhou is a

short period of heavy rainfall compared to Shanxi. The rainstorm in Shanxi is a long time rainstorm compared to Zhengzhou. From this, we find that the damage is not the same between the two. For the number of casualties, the number of damaged buildings, the area of damaged crops and the direct economic loss, the rainstorm in Zhengzhou
is significantly more destructive than that in Shanxi. In summary, we believe that the damage caused by short duration and high intensity rainstorms is much greater and more difficult for the society to bear.

4.5 Question Five

Extreme precipitation is defined as an extreme weather event when the amount of precipitation exceeds a threshold value. If a sudden heavy rainstorm occurs in an urban area, we can learn from the study of question four: strong rainfall in a short period of time and a large amount of precipitation that cannot penetrate quickly can lead to these serious consequences of casualties, building destruction, crop area loss and direct economic loss. Take the heavy rainfall in Zhengzhou in July 2021 as an example, it has the following characteristics: 1. The process accumulated rainfall is large. According to statistics, the maximum process cumulative rainfall observed in Zhengzhou was 993.1mm. 2. Strong local intensity. The maximum hourly precipitation amount reached 201.9mm. 3. Wide range of heavy rainfall. It affected the cities in north-central Henan.

In order to make suggestions on these issues, we reviewed many literature sources and found the most direct consequence of extreme precipitation: a sudden increase in runoff. Based on this, we obtained a complete line of thought: the sudden increase of runoff due to extreme precipitation causes the failure of urban pipes or the failure of expanded pipe diameters, so that urban water cannot be discharged quickly. So how to deal with the subsequent series of chain reactions caused by the increased runoff is the fundamental challenge we have to face. The following recommendations are finally made.

1. Build sponge cities. Sponge city is a new generation of urban stormwater management concept, refers to the city can be like a sponge, in adapting to environmental changes and respond to natural disasters brought about by rainwater has good resilience, can also be called "water resilient city".

The international common term is "low impact development rainwater system construction", which absorbs, stores, infiltrates and purifies water when it rains, and releases and uses the stored water when needed to realize the free migration of rainwater in the city.

Mainly from the following aspects of construction: old neighborhood sponge - the establishment of sunken green space and planting ditches, impervious surface construction, elimination of urban easy to fish points - road replacement permeable asphalt as well as the installation of rainwater barrels, so that rainwater in residential areas after treatment and storage infiltrate and discharge into water storage devices; so that rainwater on the road can quickly enter the river and thus solve the flooding

problem.

2. Do climate survey and analysis. In recent years, affected by global warming, surface evaporation as well as transpiration is increasing, prompting the water cycle to accelerate, making the occurrence of extreme rainfall events showing frequent characteristics, increasing the chances of various disasters, threatening urban security as well as development. Before building sponge cities, we should organize climate surveys, collect information and data related to urban rainfall, and analyze the laws and characteristics of extreme urban precipitation, so as to maximize the ability to cope with extreme urban precipitation events.

5. Sensitivity Analysis

In this paper, we use ARIMA time series, BP neural network and XGBoost algorithm to build three different prediction models to predict the precipitation of Zhengzhou, Guangzhou, Beijing, Suzhou, and Shanxi in 2021 simultaneously. The predicted values of the three models are compared with the actual observed values, and the model with the best fit - XGBoost. The stability of the best solution is obtained by using sensitivity analysis in the optimization model, and the stability of the optimal solution is obtained by sensitivity analysis of the model. In this paper, R^2 $\alpha \leq 0.1$ which meets the conditions for model fitting, proves the high reliability of the prediction results of the XGBoost model.

6.Strengths and Weakness

6.1 Independent Models

6.1.1 ARIMA

(1) Strengths: Concise model, only endogenous variables are required; wide range of application; good short-term prediction; good prediction when the trend of the series is obvious.

(2) Weakness: High requirement for series smoothness; can only capture linear relationships, not nonlinear relationships; large bias in long-term prediction.

6.1.2 BP neural network

(1) Strengths: Highly fault tolerant; self-learning and adaptable; good performance and trainability.

(2) Weakness: High data requirements, need to have a large amount of data for training; slow and difficult to compute.

6.1.3 XGBoost

(1) Strengths: High accuracy, high flexibility, good prediction; wide range of applications, can solve various problems such as prediction, classification, regression, etc.; high intelligence, can automatically handle missing and abnormal data.

(2) Weakness: Only applicable to structured data; complex internal logic and large amount of time consuming.

6.2 Overall Model

6.2.1 Strengths

(1) Mann-Kendall test and wavelet analysis are used to analyze the abrupt variability and periodicity of the series, and find the abrupt variation points and periods of annual precipitation in various places in the past years, which well describe the trend and characteristics of annual precipitation in various places.

(2) Three prediction models built by ARIMA time series, BP neural network and XGBoost algorithm were pre-trained, and the model predictions were fitted with the observed values using the data of 2021 to evaluate the model strengths and weaknesses, and the best model, XGBoost, was selected to predict the future daily precipitation of each place. The best model, XGBoost, was selected to forecast future daily precipitation in each region. By comparing and evaluating the models, the most suitable model for the series can be selected, which makes the prediction better.

6.2.2 Weakness

(1) Lack of endogenous analysis of mutation points, resulting in poor prediction of mutation points.

(2) The amount of data is not large enough, which leads to the model training degree is not high enough, and the accuracy of prediction effect is not good enough.

7. Conclusion

Based on the analysis of the models developed for each problem, we conclude the following.

for correlation analysis of the annual variation characteristics of each observation in Zhengzhou, we used Pearson correlation coefficients and found that there is basically no correlation between each observation and precipitation provided in this question. Then, in order to

quantitatively analyze the flood events in Zhengzhou, we conducted MK mutation analysis, wavelet analysis, precipitation trend analysis, and smoothness analysis on the precipitation observations in Zhengzhou for the past 70 years, and found that Zhengzhou had 2 sudden changes in precipitation between 1962 and 2021; there are 3 types of cycles in the precipitation evolution in Zhengzhou with scales of 25 to 32 years, 16 to 24 years, and 7 to 15 years There are three types of cycles in the precipitation evolution of Zhengzhou, namely 25~32 years, 16~24 years and 7~15 years. The 25-32 and 16-24 year time scales are very stable throughout the whole analysis period and are global in nature, while the 7-15 year scales are more stable from 1962 to 1985 and from 2010 onwards; since 1960, the annual precipitation in Zhengzhou has basically been between 10 cm and 40 cm. The fluctuation is 25 cm or less, and the trend of increasing precipitation in Zhengzhou has become more obvious since 2000.3) Zhengzhou had low precipitation in 1998 and high precipitation in 2003, 2016 and 2021; several time points with high annual precipitation in Zhengzhou are 1964, 1992, 2003, 2016 and 2021.

We collected precipitation data from three cities, Guangzhou, Beijing and Suzhou. They were subjected to MK mutation analysis, wavelet analysis, precipitation trend analysis and smoothness analysis and compared by summarizing them. It is found that four abrupt changes in precipitation occurred in Guangzhou between 1962 and 2021; there is a cyclical pattern of 3-8 years in the precipitation evolution; two abrupt changes in precipitation occurred in Shanxi between 1962 and 2021; there is a cyclical pattern of 5-20 years in the precipitation evolution, which is more stable in the period of 1980-2010; the precipitation in Beijing is more stable in the period of 1980-2010. The precipitation in Beijing has 8 abrupt changes between 1962 and 2021; the precipitation evolution has a cyclical pattern of 3-10 years; the precipitation in Suzhou has 1 abrupt change between 1962 and 2021; the precipitation evolution has a cyclical pattern of 3-10 years, and this type of scale is stable between 1990 and 2010. The precipitation in Suzhou and Guangzhou was found to be relatively stable in the period 1990-2010. In comparison, we found that the precipitation in Suzhou and Guangzhou is similar; since 2000, the precipitation in Suzhou and Guangzhou has been increasing significantly, and the precipitation in Beijing has been increasing slowly; all three places have sudden change values in 1992, 1999 and 2016; the number of extreme precipitation and days of precipitation in the three places have similar changing trends.

We chose three approaches, BP neural network, ARIMA and XGBoost model, to predict the future extreme rainfall weather in the cities. By comparing the fitted effect plots produced during the training process, we finally chose the XGBoost model to predict the future extreme rainfall weather for the city of Suzhou and obtained the conclusion that extreme rainfall weather is likely to occur in Suzhou in the future.

References

[1] Ma Jun,Zhu Guoping,Li Yan. Trend analysis of spatial and temporal distribution of precipitation in Fangshan District based on Mann-Kendall method[J]. Water Resources Development and Management,2021(01):47-51+56.

[2] Liu Jian. Application of ARIMA-based model in precipitation trend analysis and prediction[J]. Water Science and Technology and Economics,2018,24(09):67-69.

[3] Xu XC,Zhang XZ,Dell F,Song W. Trends in precipitation intensity and its impact on precipitation in China from 1961-2010[J]. Geographical Research,2014,33(07):1335-1347.

[4] Lu WX, Liu BJ, Chen JF, Chen XH. Analysis of precipitation trends in the Pearl River Basin over the past 50a[J]. Journal of Natural Resources,2014,29(01):80-90.

[5] Li Yuejun,Guo Pinwen. Application of BP neural network-based summer downscaled precipitation forecasting method in northern Zhejiang[J]. Journal of Atmospheric Sciences,2017,40(03):425-432.

[6] Sun Deliang. Research on machine learning-based landslide susceptibility zoning and rainfall-induced landslide forecasting and warning[D]. East China Normal University,2019.

[7] Xu L, Wang TANL, Liu SONGG, Li D, Li W, Tan LCC. A prediction model of precipitation change trend based on SSA-XGBoost method[j]. Journal of Earth Environment,2020,11(05):475-485.

[8] Zhu Yan,Zhai Danhua,Wu Zhipeng,Zhang Yan. Short-time intense precipitation forecasting method based on Xgboost algorithm[J]. Meteorological Science and Technology,2021,49(03):406-418.

Appendix

The MATLAB code section

```
%% Function Section
function [temp1]=dealdata1(n)
% Pre-processing of the FRSHTT items of the raw data in the first question
 temp1=[];
    [r,c]=size(n);
    for i=1:r
        if n(i,6) ==0
            temp1(i,1)=0;
        else
            temp1(i,1)=1;
        end
    end
end
function [UFk,UBk] = MKTest(y)
%UFk:顺序结果;UBk:逆序结果
n=length(y);
r=zeros(n,1);
for i=2:n
    for j=1:i
        if y(i)>y(j)
            r(i)=r(i)+1;
        end
    end
end
sk=zeros(n,1);
for i=2:n
    for j=1:i
        sk(i)=sk(i)+r(j);
    end
end
UFk=zeros(n,1);
for i=2:n
    E=i*(i-1)/4;
    % Sk(i)的均值
    Var=i*(i-1)*(2*i+5)/72;
    % Sk(i)的方差
    UFk(i)=(sk(i)-E)/sqrt(Var);
end
y2=flipud(y);
r2=zeros(n,1);
```

```
for i=2:n
    for j=1:i
        if y2(i)>y2(j)
            r2(i)=r2(i)+1;
        end
    end
end
sk2=zeros(n,1);
for i=2:n
    for j=1:i
        sk2(i)=sk2(i)+r2(j);
    end
end
UFk2=zeros(n,1);
for i=2:n
    E=i*(i-1)/4;
    % Sk(i)的均值
    Var=i*(i-1)*(2*i+5)/72;
    % Sk(i)的方差
    UFk2(i)=(sk2(i)-E)/sqrt(Var);
end
UBk=zeros(n,1);
UBk=flipud(-UFk2);
plot(1:n,UFk,'r-','linewidth',1.5);
hold on;
plot(1:n,UBk,'b-','linewidth',1.5);
plot(1:n,0*ones(n,1),'-.','linewidth',1); %绘制横轴 0 值线
plot(1:n,1.96*ones(n,1),':','linewidth',1); %绘制 95%置信区间界限
plot(1:n,-1.96*ones(n,1),':','linewidth',1);%绘制 95%置信区间界限
axis([1,n,-4,8]);%设置横坐标范围和纵坐标范围
legend('UF','UB','Significant Range'); %设置图例
xlabel('t','FontName','TimesNewRoman','FontSize',12);%设置 x 轴标签
ylabel('statistical magnitude','FontName','TimesNewRoman','Fontsize',12);%设
置 y 轴标签
xlswrite('E:\0.0\matlabb\bin.xlsx',UFk,'Sheet1','A1');
xlswrite('E:\0.0\matlabb\bin.xlsx',UBk,'Sheet1','B1');
end
%% 对第一问中的原始数据的 FRSHTT 项进行预处理
temp11=dealdata1(AstationCopy);
temp2=dealdata1(BstationCopy);
temp3=dealdata1(CstationCopy);
%% 对第一问的突变进行研究
[num1,ans1,ans2]=xlsread('data.xlsx');
[ufk,ubk]=MKTest(num1(:,6));
```

```
%% 对第一问数据进行小波分析
[num2,~,~]=xlsread('data3.xlsx');
num3=num2(:,2);
shibu=real(coefs);
mo=abs(coefs);
mofang=(mo).^2;
fangcha=sum(abs(coefs).^2,2);
shibu1=[];
year=1962;
num=1;
for j=1:64
    for i=1:32
        shibu1((j-1)*32+i,3)=shibu(i,j);
        shibu1((j-1)*32+i,1)=year;
        shibu1((j-1)*32+i,2)=num;
        num=num+1;
         end
         num=1;
         year=year+1;
    end

    %% 对第二问需要处理的数据的突变进行研究
    [num3,~,~]=xlsread('guangzhounew2.xlsx');
    [ufk3,ubk3]=MKTest(num3(:,2));
    [num4,~,~]=xlsread('shanxinew2.xlsx');
    [ufk4,ubk4]=MKTest(num4(:,2));
    [num5,~,~]=xlsread('beijingnew2.xlsx');
    [ufk5,ubk5]=MKTest(num5(:,2));
    [num6,~,~]=xlsread('suzhounew2.xlsx');
    [ufk6,ubk6]=MKTest(num6(:,2));

    %% 对第二问找到的数据进行小波分析
    signal1=num3(:,2);
    signal2=num4(:,2);
    signal3=num5(:,2);
    signal4=num6(:,2);
    %广州
    shibu11=real(coefs);
    mo1=abs(coefs);
    mofang1=(mo).^2;
    fangcha1=sum(abs(coefs).^2,2);
    shibu111=[];
    year=1962;
    num=1;
```

```
for j=1:64
    for i=1:32
        shibu111((j-1)*32+i,3)=shibu11(i,j);
        shibu111((j-1)*32+i,1)=year;
        shibu111((j-1)*32+i,2)=num;
        num=num+1;
    end
    num=1;
    year=year+1;
end
%山西
shibu2=real(coefs);
mo2=abs(coefs);
mofang2=(mo).^2;
fangcha2=sum(abs(coefs).^2,2);
shibu22=[];
year=1962;
num=1;
for j=1:64
    for i=1:32
        shibu22((j-1)*32+i,3)=shibu2(i,j);
        shibu22((j-1)*32+i,1)=year;
        shibu22((j-1)*32+i,2)=num;
        num=num+1;
    end
    num=1;
    year=year+1;
end
%北京
shibu3=real(coefs);
mo3=abs(coefs);
mofang3=(mo).^2;
fangcha3=sum(abs(coefs).^2,2);
shibu33=[];
year=1962;
num=1;
for j=1:64
    for i=1:32
        shibu33((j-1)*32+i,3)=shibu3(i,j);
        shibu33((j-1)*32+i,1)=year;
        shibu33((j-1)*32+i,2)=num;
        num=num+1;
    end
    num=1;
```

```
    year=year+1;
end
%苏州
shibu4=real(coefs);
mo4=abs(coefs);
mofang4=(mo).^2;
fangcha4=sum(abs(coefs).^2,2);
shibu44=[];
year=1962;
num=1;
for j=1:64
    for i=1:32
        shibu44((j-1)*32+i,3)=shibu4(i,j);
        shibu44((j-1)*32+i,1)=year;
        shibu44((j-1)*32+i,2)=num;
        num=num+1;
    end
    num=1;
    year=year+1;
end
%% 采用百分位法将第 90 个百分位值的 70 年平均值定义为极端降水事件的阈值
%根据统计，24h 降水总量超过 2.99cm 则算作特大暴雨，基于此设置阈值为 2.99
zhengzhou=0;
guangzhou=0;
shanxi=0;
beijing=0;
suzhou=0;
for i=1:19447
    if AstationCopy(i,11)>=2.99
        if AstationCopy(i,11)~=99.99
            zhengzhou=zhengzhou+1;
        end
    end
end
for i=1:18929
    if guangzhou1(i,1)>=2.99
        if guangzhou1(i,1)~=99.99
            guangzhou=guangzhou+1;
        end
    end
end
for i=1:18910
```

```
    if shanxi1(i,1)>=2.99
        if shanxi1(i,1)~=99.99
            shanxi=shanxi+1;
        end
    end
end
for i=1:18913
    if beijing1(i,1)>=2.99
        if beijing1(i,1)~=99.99
            beijing=beijing+1;
        end
    end
end
for i=1:18919
    if suzhou1(i,1)>=2.99
        if suzhou1(i,1)~=99.99
            suzhou=suzhou+1;
        end
    end
end
%% 总结出 70 年各城市极端降雨天数后，再对各城市每年极端降雨天数进行统计
%郑州
zhengzhounum(2021-1957+1,2)=[0];
for i=1957:2021
    zhengzhounum(i-1956,1)=i;
end
for i=1:19447
    if AstationCopy(i,11)>=2.99
        if AstationCopy(i,11)~=99.99
            zhengzhounum(Astation(i,2)-1956,2)=zhengzhounum(Astation(i,2)-1956,2)+1;
        end
    end
end
%广州
guangzhounum(2021-1962+1,2)=[0];
for i=1962:2021
    guangzhounum(i-1961,1)=i;
end
for i=1:18929
    if guangzhou2(i,4)>=2.99
        if guangzhou2(i,4)~=99.99
```

```
            guangzhounum(guangzhou2(i,1)-1961,2)=guangzhounum(guangzhou2(i,1)-1961,2)+1;
        end
    end
end
%山西
shanxinum(2021-1962+1,2)=[0];
for i=1962:2021
    shanxinum(i-1961,1)=i;
end
for i=1:18910
    if shanxi2(i,4)>=2.99
        if shanxi2(i,4)~=99.99
            shanxinum(shanxi2(i,1)-1961,2)=shanxinum(shanxi2(i,1)-1961,2)+1;
        end
    end
end
%北京
beijingnum(2021-1962+1,2)=[0];
for i=1962:2021
    beijingnum(i-1961,1)=i;
end
for i=1:18913
    if beijing2(i,4)>=2.99
        if beijing2(i,4)~=99.99
            beijingnum(beijing2(i,1)-1961,2)=beijingnum(beijing2(i,1)-1961,2)+1;
        end
    end
end
%苏州
suzhounum(2021-1962+1,2)=[0];
for i=1962:2021
    suzhounum(i-1961,1)=i;
end
for i=1:18919
    if suzhou2(i,4)>=2.99
        if suzhou2(i,4)~=99.99
            suzhounum(suzhou2(i,1)-1961,2)=suzhounum(suzhou2(i,1)-1961,2)+1;
        end
    end
end
%% 总结出各城市 70 年来降水总天数
zhengzhouall=0;
```

```
guangzhouall=0;
shanxiall=0;
beijingall=0;
suzhouall=0;
for i=1:19447
    if AstationCopy(i,11)~=0
        if AstationCopy(i,11)~=99.99
            zhengzhouall=zhengzhouall+1;
        end
    end
end
for i=1:18929
    if guangzhou1(i,1)~=0
        if guangzhou1(i,1)~=99.99
            guangzhouall=guangzhouall+1;
        end
    end
end
for i=1:18910
    if shanxi1(i,1)~=0
        if shanxi1(i,1)~=99.99
            shanxiall=shanxiall+1;
        end
    end
end
for i=1:18913
    if beijing1(i,1)~=0
        if beijing1(i,1)~=99.99
            beijingall=beijingall+1;
        end
    end
end
for i=1:18919
    if suzhou1(i,1)~=0
        if suzhou1(i,1)~=99.99
            suzhouall=suzhouall+1;
        end
    end
end
%% 总结出各城市 70 年来每年降水总天数
%郑州
zhengzhouyear(2021-1957+1,2)=[0];
for i=1957:2021
    zhengzhouyear(i-1956,1)=i;
```

```
341.    end
342.    for i=1:19447
343.        if AstationCopy(i,11)~=0
344.            if AstationCopy(i,11)~=99.99
345.                zhengzhouyear(AstationCopy(i,2)-1956,2)=zhengzhouyear(AstationCopy(i,2)-1956,2)+1;
346.            end
347.        end
348.    end
349.    %广州
350.    guangzhouyear(2021-1962+1,2)=[0];
351.    for i=1962:2021
352.        guangzhouyear(i-1961,1)=i;
353.    end
354.    for i=1:18929
355.        if guangzhou2(i,4)~=0
356.            if guangzhou2(i,4)~=99.99
357.                guangzhouyear(guangzhou2(i,1)-1961,2)=guangzhouyear(guangzhou2(i,1)-1961,2)+1;
358.            end
359.        end
360.    end
361.    %山西
362.    shanxiyear(2021-1962+1,2)=[0];
363.    for i=1962:2021
364.        shanxiyear(i-1961,1)=i;
365.    end
366.    for i=1:18910
367.        if shanxi2(i,4)~=0
368.            if shanxi2(i,4)~=99.99
369.                shanxiyear(shanxi2(i,1)-1961,2)=shanxiyear(shanxi2(i,1)-1961,2)+1;
370.            end
371.        end
372.    end
373.    %北京
374.    beijingyear(2021-1962+1,2)=[0];
375.    for i=1962:2021
376.        beijingyear(i-1961,1)=i;
377.    end
378.    for i=1:18913
379.        if beijing2(i,4)~=0
380.            if beijing2(i,4)~=99.99
381.                beijingyear(beijing2(i,1)-1961,2)=beijingyear(beijing2(i,1)-1961,2)+1;
382.            end
```

```
383.        end
384.    end
385.    %苏州
386.    suzhouyear(2021-1962+1,2)=[0];
387.    for i=1962:2021
388.        suzhouyear(i-1961,1)=i;
389.    end
390.    for i=1:18919
391.        if suzhou2(i,4)~=0
392.            if suzhou2(i,4)~=99.99
393.                suzhouyear(suzhou2(i,1)-1961,2)=suzhouyear(suzhou2(i,1)-1961,2)+1
    ;
394.            end
395.        end
396.    end
397.
398.    %% BP 神经网络代码
399.    function [Y,Xf,Af] = myNeuralNetworkFunction4(X,Xi,~)
400.    %MYNEURALNETWORKFUNCTION neural network simulation function.
401.    %
402.    % Auto-generated by MATLAB, 14-Nov-2021 14:37:10.
403.    %
404.    % [Y,Xf,Af] = myNeuralNetworkFunction(X,Xi,~) takes these arguments:
405.    %
406.    %   X = 2xTS cell, 2 inputs over TS timesteps
407.    %   Each X{1,ts} = 1xQ matrix, input #1 at timestep ts.
408.    %   Each X{2,ts} = 1xQ matrix, input #2 at timestep ts.
409.    %
410.    %   Xi = 2x2 cell 2, initial 2 input delay states.
411.    %   Each Xi{1,ts} = 1xQ matrix, initial states for input #1.
412.    %   Each Xi{2,ts} = 1xQ matrix, initial states for input #2.
413.    %
414.    %   Ai = 2x0 cell 2, initial 2 layer delay states.
415.    %   Each Ai{1,ts} = 10xQ matrix, initial states for layer #1.
416.    %   Each Ai{2,ts} = 1xQ matrix, initial states for layer #2.
417.    %
418.    % and returns:
419.    %   Y = 1xTS cell of 2 outputs over TS timesteps.
420.    %   Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
421.    %
422.    %   Xf = 2x2 cell 2, final 2 input delay states.
423.    %   Each Xf{1,ts} = 1xQ matrix, final states for input #1.
424.    %   Each Xf{2,ts} = 1xQ matrix, final states for input #2.
425.    %
```

```
%   Af = 2x0 cell 2, final 0 layer delay states.
%   Each Af{1ts} = 10xQ matrix, final states for layer #1.
%   Each Af{2ts} = 1xQ matrix, final states for layer #2.
%
% where Q is number of samples (or series) and TS is the number of timesteps
.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = 0.00463422413406744;
x1_step1.gain = 2.01714473683379;
x1_step1.ymin = -1;

% Input 2
x2_step1.xoffset = 0.457546892419602;
x2_step1.gain = 1.42508284962891;
x2_step1.ymin = -1;

% Layer 1
b1 = [2.1942387783588217509;1.434141512509035099;-1.169776148038160
99;-0.80799414377658629327;0.1546021367610952435;0.11209797046570944
834;-0.11002928019124812165;1.5420988798096573191;-1.9405954615645575
334;-2.7387762486976212628];
IW1_1 = [-1.2406288237734595103 -1.6491780111320972768;-0.556280222
87934841827 -0.21554852079700193013;0.026009067761125530954 0.7639977
9450906724065;1.2717806993211804301 1.4064521217717820267;0.00531146
65770457453187 0.14330568265600385214;2.0312531278066350282 0.615542
24413187874632;-0.17334681070226404254 -0.067493417029236335569;1.862
064827058531602 -0.30660729426574845347;-1.0607968892786416859 -0.414
12957508389519612;-0.87936837974182913147 1.5559978675449888108];
IW1_2 = [-0.94236763763777164904 1.2881648405941326097;0.146424861
50009492718 -0.24358228922704758257;-1.0689806592362920412 -0.2354858
3490530991535;-0.09129764082395248892 -1.5842426296473877123;-0.20048
879407595424085 -0.043972644549794312474;-0.51056457774493457791 -1.5
17968445609461536 3;0.046060635256925712422 -0.076044180537447436663;
1.1166483421755615257 -0.67163888600826981978;0.2837066873123327837 -
0.46420553436390432633;0.049856559337638460538 1.4429250575525001121]
;

% Layer 2
b2 = 0.099124773252133174806;
```

```
453.    LW2_1 = [-5.3893377988970488755e-06 -0.28183395246219516972 -0.0382
        18875339870096719 1.33449973758469775e-05 -1.9268860313177920141 3.57
        72816328797943874e-06 -3.9825358588920574121 4.7628397099268607003e-0
        5 -0.033624540032292107861 -7.0409406604083854021e-06];
454.
455.    % Output 1
456.    y1_step1.ymin = -1;
457.    y1_step1.gain = 1.42508284962891;
458.    y1_step1.xoffset = 0.457546892419602;
459.
460.    % ===== SIMULATION ========
461.
462.    % Format Input Arguments
463.    isCellX = iscell(X);
464.    if ~isCellX
465.        X = {X};
466.    end
467.    if (nargin < 2), error('Initial input states Xi argument needed.'); end
468.
469.    % Dimensions
470.    TS = size(X,2); % timesteps
471.    if ~isempty(X)
472.        Q = size(X{1},2); % samples/series
473.    elseif ~isempty(Xi)
474.        Q = size(Xi{1},2);
475.    else
476.        Q = 0;
477.    end
478.
479.    % Input 1 Delay States
480.    Xd1 = cell(1,3);
481.    for ts=1:2
482.        Xd1{ts} = mapminmax_apply(Xi{1,ts},x1_step1);
483.    end
484.
485.    % Input 2 Delay States
486.    Xd2 = cell(1,3);
487.    for ts=1:2
488.        Xd2{ts} = mapminmax_apply(Xi{2,ts},x2_step1);
489.    end
490.
491.    % Allocate Outputs
492.    Y = cell(1,TS);
493.
```

```
% Time loop
for ts=1:TS

    % Rotating delay state position
    xdts = mod(ts+1,3)+1;

    % Input 1
    Xd1{xdts} = mapminmax_apply(X{1,ts},x1_step1);

    % Input 2
    Xd2{xdts} = mapminmax_apply(X{2,ts},x2_step1);

    % Layer 1
    tapdelay1 = cat(1,Xd1{mod(xdts-[1 2]-1,3)+1});
    tapdelay2 = cat(1,Xd2{mod(xdts-[1 2]-1,3)+1});
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*tapdelay1 + IW1_2*tapdelay2)
;

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
end

% Final Delay States
finalxts = TS+(1: 2);
xits = finalxts(finalxts<=2);
xts = finalxts(finalxts>2)-2;
Xf = [Xi(:,xits) X(:,xts)];
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS ========
```

```
The Python code section
# 第一问画图
import pandas as pd
import numpy as np
```

```
from sklearn import preprocessing
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
data = pd.read_excel(&apos;1.xlsx&apos;)
from sklearn import preprocessing
data = pd.read_excel(&apos;test1.xlsx&apos;)
total = data.isnull().sum().sort_values(ascending=False)
print(total)


var = &apos;DATE&apos;
data = pd.concat([data[&apos;WDSP&apos;], data[var]], axis=1)
data.plot.scatter(x=var, y=&apos;WDSP&apos;);

df_train = pd.read_excel(&apos;STATION1.xlsx&apos;)
corrmat = df_train.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True);
k = 10 #number of variables for heatmap

cols = corrmat.nlargest(k, &apos;PRCP&apos;)[&apos;PRCP&apos;].index
cm = np.corrcoef(df_train[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt=&apos;.2f&apos;, annot_kws={&apos;size&apos;: 10}, yticklabels=cols.values, xticklabels=cols.values)
plt.show()

data = pd.read_excel(r&apos;3.xlsx&apos;,index_col = u&apos;DATE&apos;)
plt.rcParams[&apos;font.sans-serif&apos;] = [&apos;SimHei&apos;]
plt.rcParams[&apos;axes.unicode_minus&apos;] = False
data.plot()
plt.show()
plot_acf(data)
plt.show()
from statsmodels.tsa.stattools import adfuller
print(&apos;result：&apos;,adfuller(data[u&apos;PRCP&apos;]))


a = [0,1.23,1.02,0,0.2]
b = [1.61,2.91,7.43,2,0.31]
c = [0,0,0,0,0.02]
num1=pd.Series(a)
```

```
num2=pd.Series(b)
num3=pd.Series(c)
print(num1.describe())
print(num2.describe())
print(num3.describe())

a1 = [0,0,0.49,0,0]
b1 = [0.08,0.03,0,0.19,0.5]
c1 = [1.61,2.91,7.43,2,0.31]
num4=pd.Series(a1)
num5=pd.Series(b1)
num6=pd.Series(c1)
print(num4.describe())
print(num5.describe())
print(num6.describe())
np.var(b)
np.std(b)

data = pd.read_excel(&apos;suzhounew2.xlsx&apos;)
var = &apos;Year&apos;
data = pd.concat([data[&apos;PRCP&apos;], data[var]], axis=1)
data.plot.scatter(x=var, y=&apos;PRCP&apos;);
data = pd.read_excel(r&apos;shanxinew2.xlsx&apos;,index_col = u&apos;Year&apos;)
plt.rcParams[&apos;font.sans-serif&apos;] = [&apos;SimHei&apos;]
plt.rcParams[&apos;axes.unicode_minus&apos;] = False
data.plot()
plt.show()
plot_acf(data)
plt.show()
from statsmodels.tsa.stattools import adfuller
print(&apos;result：&apos;,adfuller(data[u&apos;PRCP&apos;]))

# 第二问画图
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt

data1 = pd.read_excel(&apos;beijingnew2.xlsx&apos;)
data2 = pd.read_excel(&apos;suzhounew2.xlsx&apos;)
# data3 = pd.read_excel(&apos;shanxinew2.xlsx&apos;)
data4 = pd.read_excel(&apos;guangzhounew2.xlsx&apos;)
var = data1[[&apos;Year&apos;]]
```

```
90.
91.
92.  sub_axix = filter(lambda x:x%200 == 0, var)
93.  plt.plot(var, data1[[&apos;PRCP&apos;]],color=&apos;turquoise&apos;, label=&
     apos;beijing&apos;,linewidth=5)
94.  plt.plot(var, data2[[&apos;PRCP&apos;]], color=&apos;tomato&apos;, label=&a
     pos;suzhou&apos;,linewidth=5)
95.  # plt.plot(var, data3[[&apos;PRCP&apos;]], color=&apos;orange&apos;, label
     =&apos;shanxi&apos;,linewidth=5)
96.  plt.plot(var, data4[[&apos;PRCP&apos;]], color=&apos;lightskyblue&apos;, labe
     l=&apos;guangzhou&apos;,linewidth=5)
97.  plt.legend() # 显示图例
98.  plt.grid()
99.
100.     plt.xlabel(&apos;Year&apos;)
101.     plt.ylabel(&apos;PRCP&apos;)
102.     plt.show()
103.
104.     # 第三问 XGBoost 代码
105.     import numpy as np
106.     import pandas as pd
107.     import xgboost as xgb
108.     import datetime
109.     from sklearn.linear_model import LinearRegression # 线性回归的类
110.     from sklearn.preprocessing import StandardScaler # 数据标准化
111.     from sklearn.model_selection import train_test_split # 数据划分的类
112.     import matplotlib as mpl
113.     import matplotlib.pyplot as plt
114.     import lightgbm as lgb
115.
116.
117.     def fill(train):
118.         start_2_end_data = pd.date_range(start=train.loc[0, &apos;DATE&apos;], e
     nd=train.loc[len(train)-1, &apos;DATE&apos;])\
119.             .strftime("%Y-%m-%d").to_list()
120.         existing_data = train[&apos;DATE&apos;].map(lambda x: x.strftime("%Y-
     %m-%d")).to_list()
121.         # 日期缺失的数据
122.         lost_data = [i for i in start_2_end_data if i not in existing_data]
123.         train.index = pd.DatetimeIndex(train[&apos;DATE&apos;])
124.         for i in lost_data:
125.             train.loc[pd.to_datetime(i), &apos;PRCP&apos;] = 0
126.         train[&apos;DATE&apos;] = pd.date_range(start=start_2_end_data[0], end
     =start_2_end_data[-1])
```

```
127.         return train.sort_index()
128.
129.
130.     # def train_validate_test(off_train, off_test):
131.     #     train_history_field = off_train[
132.     #         off_train[&apos;DATE&apos;].isin(pd.date_range(&apos;1983/7/1&apos;, periods=4000))]  # [1983-07-01,1994-06-12)
133.     #     train_label_field = off_train[
134.     #         off_train[&apos;DATE&apos;].isin(pd.date_range(&apos;1994/6/13&apos;, periods=1000))]  # [1994-06-13,1997-03-08)
135.     #     # 验证集历史区间、标签区间
136.     #     validate_history_field = off_train[
137.     #         off_train[&apos;DATE&apos;].isin(pd.date_range(&apos;1997/3/8&apos;, periods=4000))]  # [1997-03-08,2008-02-18)
138.     #     validate_label_field = off_train[
139.     #         off_train[&apos;DATE&apos;].isin(pd.date_range(&apos;2008/2/19&apos;, periods=1000))]  # [2008-02-19,2010-11-14)
140.     #     # 测试集历史区间、中间区间、标签区间
141.     #     test_history_field = off_train[
142.     #         off_train[&apos;DATE&apos;].isin(pd.date_range(&apos;2010/11/15&apos;, periods=4000))]  # [2010-11-15,2021-10-27)
143.     #     test_label_field = off_test  # [2021-10-28,2024-07-23)
144.     #
145.     #     return train_history_field, train_label_field, validate_history_field, \
146.     #         validate_label_field, test_history_field, test_label_field
147.
148.
149.     def get_feature(df):
150.         df.index = range(len(df))
151.         df[&apos;DATE_split&apos;] = df[&apos;DATE&apos;].map(lambda x: x.strftime("%Y-%m-%d").split(&apos;-&apos;))
152.         for i in range(len(df)):
153.             df.loc[i, &apos;year&apos;], df.loc[i, &apos;month&apos;], df.loc[i, &apos;date&apos;] = \
154.                 int(df.loc[i, &apos;DATE_split&apos;][0]),  int(df.loc[i, &apos;DATE_split&apos;][1]), int(df.loc[i, &apos;DATE_split&apos;][2])
155.         del df[&apos;DATE_split&apos;]
156.         return df
157.
158.
159.     def train_model(train):
160.         X_train = train.drop([&apos;PRCP&apos;], axis=1)
161.         Y_train = train[&apos;PRCP&apos;]
```

```
162.        X2_train, X2_test, Y2_train, Y2_test = train_test_split(X_train, Y_train, tes
    t_size=0.2, random_state=0)
163.
164.        Y2_predict = xgb_(train, X2_test)[0]
165.        Y2_predict1 = lgb_(train, X2_test)[0]
166.        t = np.arange(len(X2_test))
167.        plt.figure(facecolor=&apos;w&apos;)
168.        plt.plot(t, Y2_test, &apos;b.&apos;, label=u&apos;真实值&apos;)
169.        plt.plot(t, Y2_predict, &apos;r-&apos;, label=u&apos;xgb 预测值&apos;)
170.        plt.plot(t, Y2_predict1, &apos;g-&apos;, label=u&apos;lgb 预测值&apos;)
171.        plt.legend(loc=&apos;lower right&apos;)
172.        plt.title(u"降雨量与年份", fontsize=20)
173.        plt.grid(b=True)
174.        plt.show()
175.
176.
177.    def xgb_(train, test):
178.        X_train = train.drop([&apos;PRCP&apos;, &apos;DATE&apos;], axis=1)
179.        Y_train=train[&apos;PRCP&apos;]
180.        if &apos;PRCP&apos; in test.columns:
181.            X_test = test.drop([&apos;PRCP&apos;, &apos;DATE&apos;], axis=1)
182.        else:
183.            X_test = test.drop([&apos;DATE&apos;], axis=1)
184.        scaler = StandardScaler()
185.        X_train = scaler.fit_transform(X_train)  # 训练并转换
186.        X_test = scaler.transform(X_test)  ## 直接使用在模型构建数据上进行一
    个数据标准化操作
187.
188.        model = xgb.XGBRegressor(max_depth=3,
189.                        learning_rate=0.1,
190.                        n_estimators=100,
191.                        objective=&apos;reg:linear&apos;,
192.                        n_jobs=-1)
193.        # 预测
194.        model.fit(X_train, Y_train,
195.                    eval_set=[(X_train, Y_train)],
196.                    eval_metric=&apos;logloss&apos;,
197.                    verbose=100)
198.        predict = model.predict(X_test)
199.        # 处理结果
200.        pred = pd.DataFrame(predict, columns=[&apos;PRCP&apos;])
201.        result = pd.concat([test[[&apos;DATE&apos;]], pred], axis=1)
202.        return predict, result
203.
```

```
204.
205.    def lgb_(train, test):
206.        X_train = train.drop([&apos;PRCP&apos;, &apos;DATE&apos;], axis=1)
207.        Y_train = train[&apos;PRCP&apos;]
208.        if &apos;PRCP&apos; in test.columns:
209.            X_test = test.drop([&apos;PRCP&apos;, &apos;DATE&apos;], axis=1)
210.        else:
211.            X_test = test.drop([&apos;DATE&apos;], axis=1)
212.
213.        scaler = StandardScaler()
214.        X_train = scaler.fit_transform(X_train)  # 训练并转换
215.        X_test = scaler.transform(X_test)  ## 直接使用在模型构建数据上进行一
    个数据标准化操作
216.
217.        gbm = lgb.LGBMRegressor(num_leaves=31,
218.                                learning_rate=0.05,
219.                                n_estimators=20)
220.        gbm.fit(X_train, Y_train,
221.                eval_set=[(X_train, Y_train)],
222.                eval_metric=&apos;logloss&apos;,
223.                verbose=100)
224.        predict = gbm.predict(X_test)
225.        pred = pd.DataFrame(predict, columns=[&apos;PRCP&apos;])
226.        result = pd.concat([test[[&apos;DATE&apos;]], pred], axis=1)
227.        return predict, result
228.
229.
230.    if __name__ == &apos;__main__&apos;:
231.        train_data = pd.read_excel(&apos;4.xlsx&apos;)
232.        train_data = fill(train_data)
233.        test_data = pd.read_excel(&apos;5.xlsx&apos;)
234.
235.        train_data = get_feature(train_data)
236.        test_data = get_feature(test_data)
237.
238.        # show = train_model(train_data)
239.        pred = xgb_(train_data, test_data)[1]
240.        # pred1 = lgb_(train_data, test_data)[1]
241.        pred.to_excel(&apos;6.xlsx&apos;)
```