Journal of Globe Scientific Reports

journal homepage: www.journal-gsr.online

Paper Type: Original Article

Research on Water Quality Prediction Based on Machine Learning

Xudong Chen[1#] Fengtian Pei[2#] Minghao Liu[3#] Zejun Chen[4#] Keqin Li[5#] Jingcheng Xie[6#]

1. College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China

2. School of Information and Intelligent Manufacturing, Jiangyang City Construction College, Sichuan, China

3. College of Water&Architectural Engineering, Shihezi University, Shihezi, China

4. Faculty of Social Sciences, University of Macau, Macao, China

5. Saint-Petersburg Institute for Shipbuilding & MarineTechnology of Guangdong Ocean University, Guangdong Ocean University, Zhanjiang, China

6. School of Software Engineering, Tongji University, Shanghai, China

#Co-first author

Abstract

At present, some urban water plants in China have started using chloramine disinfection. So how to determine whether the disinfected water is drinkable? This article collected a water quality prediction data, including indicators such as chloramine and trihalomethanes. Firstly, descriptive statistics and Pearson correlation analysis were conducted between the data of chloramine and trihalomethanes and the target variable (whether it is drinkable). It is known that water quality cannot be judged solely based on these two indicators, so more indicators such as pH value will be used. In order to establish a more accurate prediction model, the dataset is first preprocessed, including statistical analysis of missing values, determination of box plot outliers, and filling with KNN algorithm. Then, feature engineering is performed, including Yeo Johnson transformation, correlation analysis, and calculation of Shap values. Subsequently, the processed data was input into the established Stacking, Voting, and attention based CNN-LSTM classification prediction models. Random search and cross validation were used to train each model, resulting in the optimal hyperparameters for each model. The relevant evaluation indicators for each model were calculated to measure its accuracy.

Keywords: Chloramines, Trihalomethanes, Data preprocessing, Feature engineering, Stacking, Voting, Attention based CNN-LSTM.

1 | Introduction

With the global population increasing and industrial activities expanding, ensuring the safety of drinking water has become a critical issue in environmental science and public health. Water disinfection is essential for safe water supply, but traditional methods like chlorination have associated health risks due to by-products such as trihalomethanes (THMs), which negatively impact human health over the long term. Despite its widespread use due to low cost and simplicity, chlorination's potential carcinogenic and toxic by-products necessitate safer alternatives.

Chloramine disinfection has emerged as a promising solution, effectively reducing THM formation and minimizing the chlorine taste in water. While some urban water treatment plants have adopted chloramine, further research is needed to validate its safety and practicality comprehensively.

This study explores chloramine disinfection's effectiveness and safety by analyzing water quality data. Descriptive statistics and Pearson correlation analysis of key indicators like chloramine and THMs reveal that single indicators are insufficient for assessing water quality safety. Therefore, additional parameters such as pH are included to develop a comprehensive water quality prediction model.

2 | Establishment of Quality Prediction Model

The development of water quality prediction models is essential for determining the suitability of disinfected water for consumption. With advancements in technology and data processing capabilities, researchers can now use two main types of models—mechanistic and non-mechanistic prediction models—to forecast water quality, providing scientific guidance for water pollution control [1].

Mechanistic prediction models are based on the fundamental physical and chemical principles governing water quality changes. These models typically involve detailed environmental parameters such as temperature, pH, dissolved oxygen content, and organic load, along with their interactions. Researchers gather this data through laboratory experiments or field sampling and use statistical and environmental science principles to build mathematical models describing water quality changes. These models help scientists understand the behavior of various pollutants in water bodies, predict the impact of different treatment methods on water quality, and assess long-term trends. For example, dynamic simulations of nitrogen and phosphorus exchange in water can predict algal growth and the associated risk of eutrophication [2].

Non-mechanistic prediction models rely on extensive historical data and advanced data analysis techniques. This approach does not depend directly on the physical or chemical mechanisms of water quality changes but uses machine learning algorithms, statistical models, and other tools to identify patterns and relationships in the data [3]. By collecting water quality monitoring data from past years, researchers can employ time series analysis, regression analysis, or more complex machine learning models, such as support vector machines and neural networks, to forecast future water quality indicators. These models excel at handling large, variable datasets and can continuously improve prediction accuracy by adapting to new data [4], [5].

2.1| Descriptive statistics and correlation testing

Firstly, we calculate the relationship between the content of chloramine and trihalomethanes and their drinkability, and determine whether the drinkability of water quality can be judged solely based on these two characteristics [6], [7].

Table 1 descriptive statistics

Water quality type	Chloramine content	trihalomethane content
The average content of drinkable water	7.09 mg/L	66.30 ug/L
Median of drinkable water	7.09 mg/L	66.54 ug/L
The average content of non drinkable water	7.17 mg/L	66.54 ug/L
Median of non drinkable water	7.22 mg/L	66.68 ug/L

From statistical data, the difference between chloramine and trihalomethanes in drinkable and non drinkable water samples is not significant, indicating that these two indicators alone are not sufficient to determine the drinkability of water. Next, we will further calculate the correlation between these two indicators and drinkability.

Pearson correlation coefficient is commonly used to measure the linear relationship between two variables X and Y. Pearson correlation coefficient. The calculation formula for r is as follows:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (1)$$

The correlation coefficient, which ranges from -1 to 1, quantifies the strength and direction of the linear relationship between two variables. A positive value indicates a direct relationship, while a negative value signifies an inverse relationship. In addition to calculating the correlation coefficient, a significance test (typically the p-value) assesses the reliability of the results. A p-value less than

0.05 indicates a statistically significant correlation, whereas a p-value greater than 0.05 suggests that the correlation may be due to chance.

In this study, the correlation coefficient between chloramine and drinkability is 0.024 with a p-value of 0.174, indicating a very weak and statistically insignificant relationship. Similarly, the coefficient for trihalomethanes (THMs) is 0.007 with a p-value of 0.691, also showing a weak and insignificant relationship with water drinkability. These findings suggest that the concentration levels of chloramine and THMs are not effective predictors of drinkability, indicating a need to consider additional chemical indicators or factors for a more accurate assessment of water quality.

2.2| Data preprocessing

2.2.1| Missing value statistics

We first examined the missing values of each field and then used appropriate methods to handle them. View missing values through Pandas library and missingno library, as shown in Figure 1.

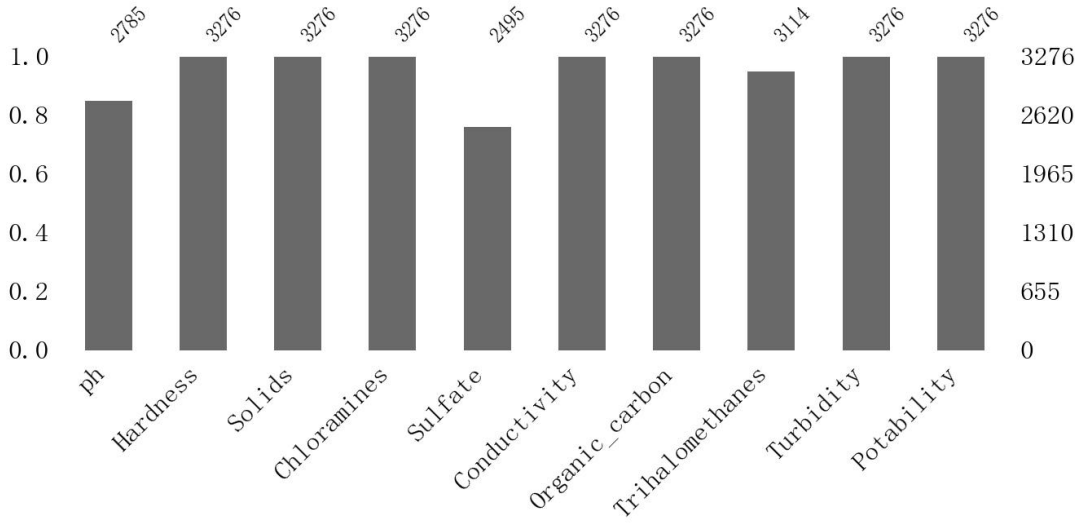


Figure 1 Missing Value Statistics Chart

The number of missing values for the characteristic "pH value" is 491, accounting for 14.99%. The number of missing values for the characteristic "sulfate (mg/L)" is 781, accounting for 23.84%. The number of missing values for the characteristic "trihalomethane (unit: μ g/L)" is 162, accounting for 4.95%.

2.2.2| KNN filling

The data analysis revealed some missing values. We decided to use the KNNImputer class to address this issue. This method uses the K-nearest neighbors (KNN) algorithm to estimate missing values based on the similarity between "neighbors."

We set the n_neighbors parameter to 5, meaning the nearest 5 non-missing values were used for each imputation. The KNN algorithm's ability to consider the distribution of other features enhances the reliability and accuracy of the imputation process. This approach is more effective than traditional methods like fixed value or mean imputation [8], [9].

2.2.3| Outlier test

In this article, the boxplot() function in the matplotlib library is used to draw a box plot for each column of data, and the number of outliers in each column of the dataset is calculated based on the results of the box plot. According to the box plot principle, data with IQR less than Q1-1.5 × IQR or greater than Q3+1.5 × IQR are defined as outliers. These data need to be filtered out and only valid data should be retained for subsequent analysis. Finally, draw another boxplot using the filtered data and save it to the output directory. Through this approach, outlier handling can be automated, reducing manual intervention during data processing, and enabling more objective identification of outliers in the data, as well as accurate calculation of the number of outliers. The following figure shows the original data box plots of all features and the data box plots after outlier processing[10], [11].As shown in Figure 2.

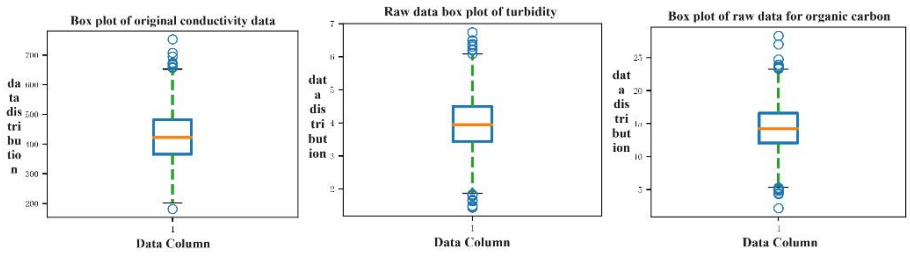


Figure 2 outlier test

2.2.4| Yeo-Johnson

To prevent model overfitting and enhance generalization, it is crucial to analyze the data distribution for consistency. We imported the data and used Python to assess whether each column followed a normal distribution, employing the stats.skew() function from the SciPy library to check for the need for Yeo-Johnson transformation. Skewness quantifies the asymmetry of the data distribution relative to the mean: a skewness of 0 indicates symmetry, while positive or negative values indicate right or left skewness, respectively. In this study, if the absolute skewness exceeded the threshold of 0.05, the data was deemed non-normally distributed and required Yeo-Johnson transformation. Figure 3 illustrates the original density histogram of Distancel alongside the histogram after the Yeo-Johnson transformation.

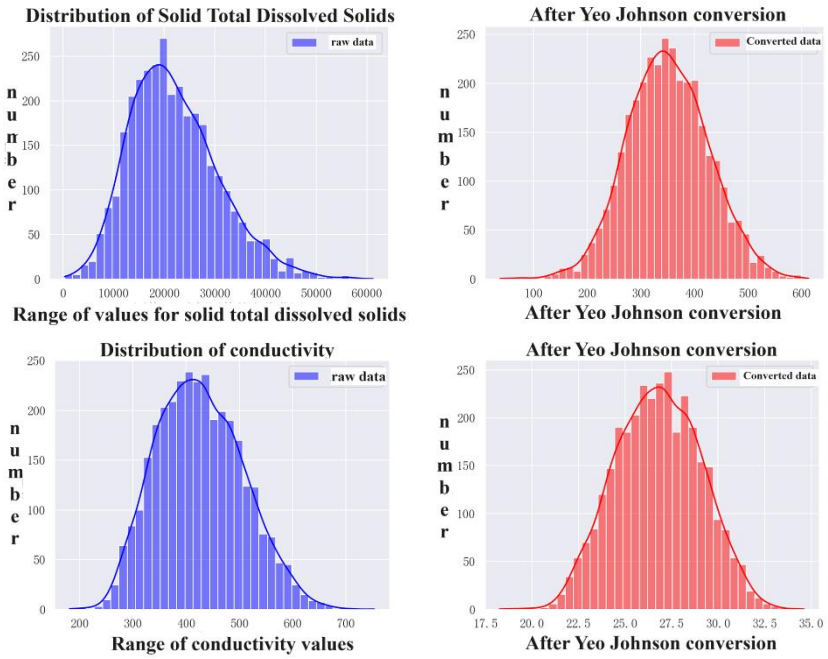


Figure 3 Yeo-Johnson

2.3| Feature engineering

2.3.1| Pearson correlation analysis

From Figure 4, it can be seen that the correlation between most features is not very high. So we don't need to delete the features.

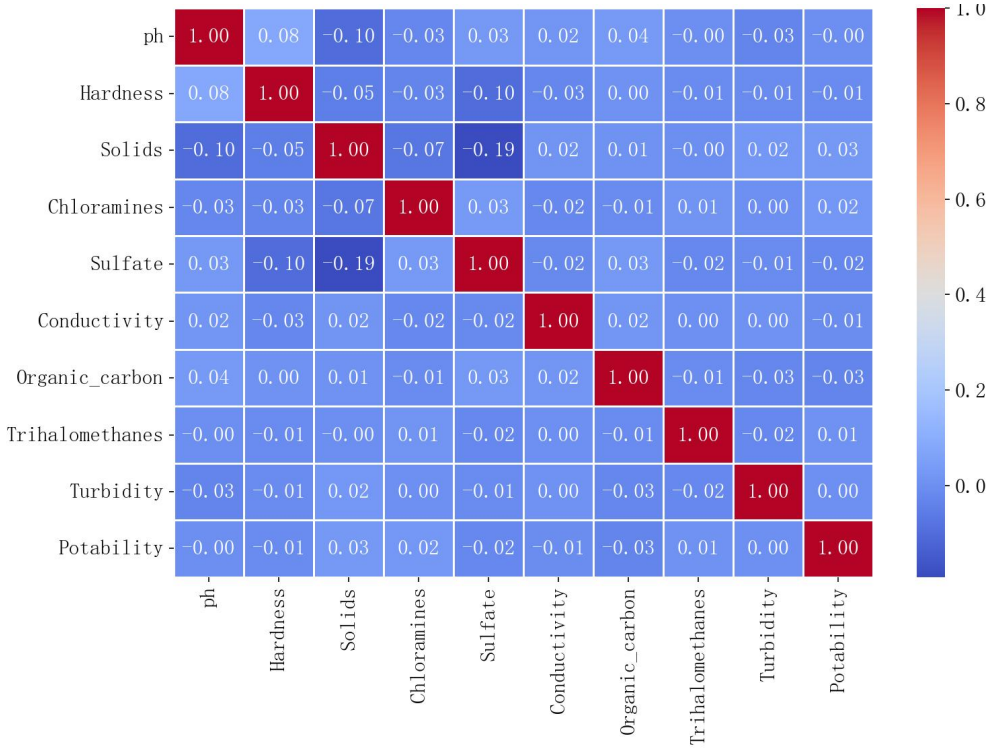


Figure 4 Pearson correlation

2.3.2| Shap value analysis

Next, we will calculate the importance of the features. SHAP values are based on Shapley values in cooperative game theory, which can provide explanations for the contribution of each feature to model predictions. As shown in Figure 5.

The importance of all features is relatively average, with no particularly prominent or low ones.

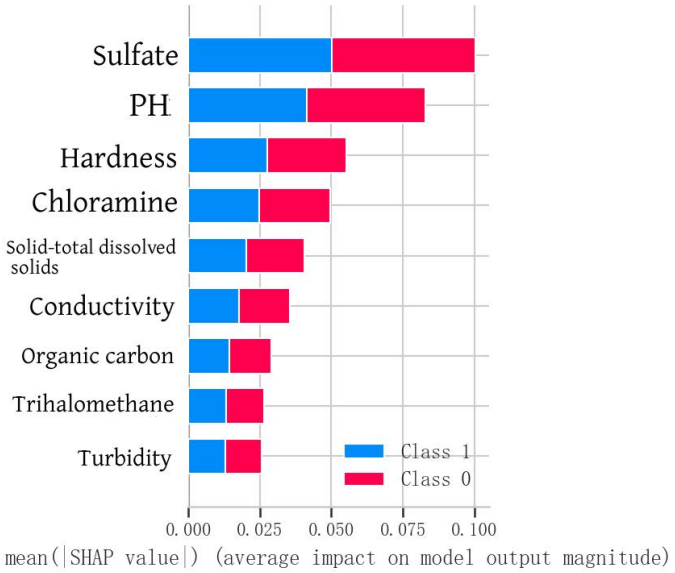


Figure 5 Importance of Features

Combining correlation analysis and feature importance:

Correlation analysis: There is no high or low correlation between features, so from a correlation perspective, there are no obvious candidate features that need to be removed.

Feature importance: The importance of all features is relatively balanced, with no particularly low feature importance values, indicating that each feature provides certain information to the model.

Therefore, from the current analysis, there is not enough reason to remove any feature. All features have their unique contributions and are not highly correlated with each other, indicating that they provide relatively independent information. It is recommended to retain all features for model training without further business understanding or other external information.

2.4| Division of training and testing sets

When processing the dataset to prepare a water quality prediction model, MinMaxScaler was first applied for feature normalization. Normalization is the process of scaling the range of data to a

given minimum and maximum value, typically between 0 and 1. For each feature, the normalization calculation formula is:

$$x_{\text{scaled}} = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

Where x is the original data value, min (x) and max (x) are the minimum and maximum values of the feature in the entire dataset, respectively. In this way, all eigenvalues are rescaled to the range of 0 to 1, which helps to handle eigenvalues of different magnitudes and can accelerate the convergence speed of algorithms such as gradient descent, as they are not affected by certain extreme values during the calculation process.

Next, the data is divided into a training set and a testing set. The segmentation ratio is 20% for the test set and 80% for the training set. This segmentation is achieved through the train_test_stplit function, where test_stize=0.2 is set to specify the proportion of the test set. In order to ensure that the segmented data can still maintain the proportion of each category in the original data, a hierarchical sampling strategy (strategy=y) was adopted. Hierarchical sampling ensures that the proportion of each category in the training and testing sets is similar to that in the original dataset, which is particularly important for handling imbalanced datasets and can avoid bias caused by too few samples in a certain category during model training.

2.5| Establishment of Water Quality Prediction Model

2.5.1| Model training and parameter optimization

In this article, three popular gradient boosting machine models were used: XGBoost, LightGBM, and CatBoost. These models are efficient tools for classification tasks in machine learning, which improve prediction accuracy by constructing a series of decision trees and making each new tree improve the errors of the previous one.

In order to find the optimal configuration for each model, this paper adopts the Randomized Search CV method for random search optimization of hyperparameters. This method is different from Grid Search, as it randomly selects parameter combinations within a specified hyperparameter space for model training and evaluation. This randomness helps to explore a wider parameter space while reducing computational costs. In this process, StratifiedKFold was used for cross validation to ensure the effectiveness of the evaluation and reduce the risk of model overfitting. This method divides the data into k subsets, with each subset alternately serving as the test set and the rest as the training set during model training, while maintaining the same proportion of categories in each subset as in the complete dataset, thus ensuring the balance between training and validation.

After completing the hyperparameter search, select the best performing model instance from the search results. These best model instances were subsequently used to construct BaggingClassifier,

which is an ensemble learning method. Ensemble learning improves prediction accuracy and stability by combining multiple models, with the core idea of bagging (Bootstrap Aggregating) being to resample the original dataset multiple times to form multiple different training data subsets, and then train a base model for each subset. In this example, each type of model (XGBoost, LightGBM, and CatBoost) uses the best parameters found by RandomSearchCV to train multiple independent models, and integrates these models through BaggingClassifier. The final prediction result of the integrated model is based on the average or majority vote of all individual model predictions. This method can significantly improve the model's generalization ability to new data and reduce the risk of overfitting on specific samples. Figure 6 is a schematic diagram of hyperparameters for three models in this article, corresponding to different seed models.

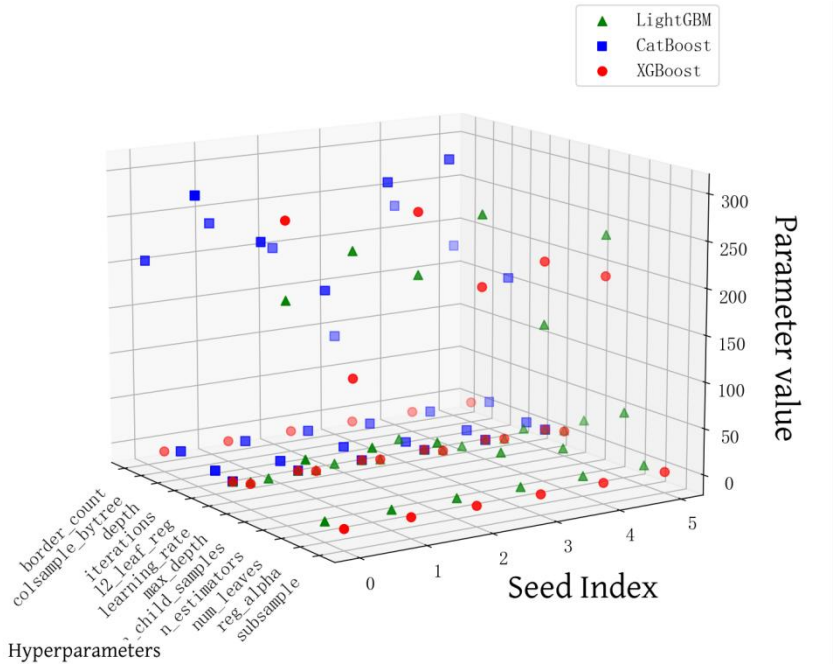


Figure 6 The optimal hyperparameters for gradient boosting algorithm

2.5.2| Model fusion

In this article, an advanced model fusion technique called stacking is employed to improve the prediction accuracy and robustness of the final model. Stacking is an integration technique that works by training another model (called a meta model) on the outputs of multiple base models in the first layer. In this method, several different models are first trained (based on different ensemble BaggingClassifiers in this article), and then the outputs of these models (usually probability predictions of categories) are used as a new feature set to train the meta model. In this article, the meta model is logistic regression, which makes the final decision by considering the probability of the base model output. Using probability rather than category labels allows the meta model to capture more information about uncertainty, thereby making more refined decisions. When implementing stacking, it is first necessary to ensure that different base models are

sufficiently diverse, as model diversity is the key to improving stacking performance. Then, the meta model is trained by using the predicted results of the base model as input. This approach enables the meta model to learn which base models are more reliable in specific situations, thereby optimizing the overall prediction performance.

2.5.3| Performance evaluation

In this article, the performance evaluation phase focuses on using multiple metrics to comprehensively evaluate the performance of the model, ensuring that the model's performance is understood from different perspectives.

Firstly, the following four commonly used performance evaluation metrics were adopted:

Accuracy is the proportion of correctly classified predictions (true cases and true negative cases) to the total sample size. The formula is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Accuracy is the proportion of observations correctly predicted as positive to the total number of observations predicted as positive. The formula is:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

The recall rate is the proportion of observations correctly predicted as positive classes to the total number of actual positive classes. The formula is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

The F1 score is the harmonic mean of precision and recall, representing a balance between these two indicators. The formula is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

The calculation results are shown in Table 2.

Table 2 Model comparison chart

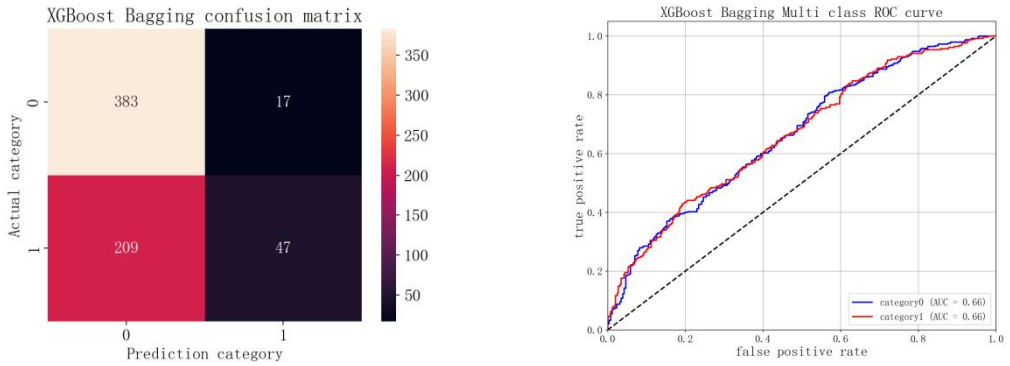
	Accuracy	precision	recall	F1 score
XGBoost Bagging	0.655488	0. 690667	0.570547	0.53296
LightGBM Bagging	0.663110	0.654275	0.600000	0.591284

CatBoost Bagging	0.660061	0.654275	0.596094	0.586298
Stacking	0.650915	0.627307	0.611094	0.612312
Voting	0.664634	0.668060	0.594219	0.578898
Attention based CNN-LSTM	0.68598	0.679705	0.685976	0.667777

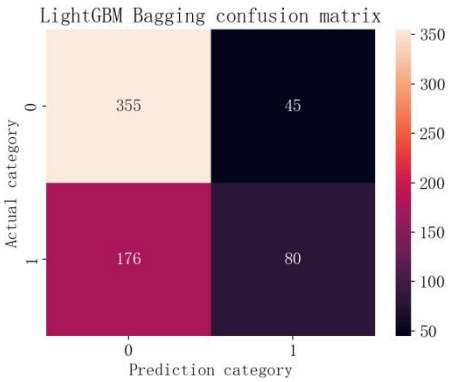
Secondly, use confusion matrix to intuitively understand prediction accuracy. Confusion matrix is a very useful tool that demonstrates the performance of the model in each category, including true positives, false positives, true negatives, and false negatives. By using the confusion matrix, it is possible to intuitively see which categories the model performs well in and which categories have problems.

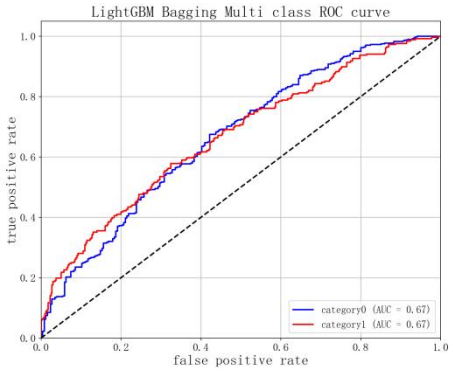
Then draw the ROC curve, which is an important tool for evaluating the classification performance of the model, especially when the dataset is imbalanced, along with the receiver operating characteristic curve (ROC curve) and AUC (area under the curve). The ROC curve is plotted by calculating the true positive rate (recall) and false positive rate (false positive rate) at different thresholds. The AUC value provides the ability of the model to distinguish between positive and negative classes, and the higher the AUC value, the better the performance of the model. The calculation formula for AUC is the area under the ROC curve.

In this article, ROC curves were plotted for each model and corresponding AUC values were calculated, which helps evaluate the model's classification ability for different categories. By drawing ROC curves and calculating AUC separately for each category, we can gain a detailed understanding of the model's performance on each category, providing guidance for further optimizing the model.As shown in Figure 7.

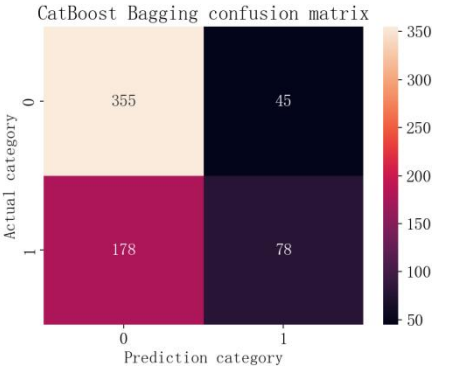


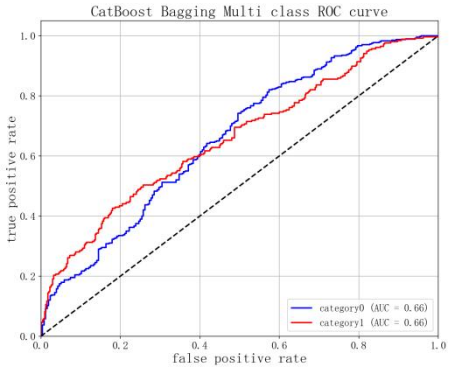
XGBoost Bagging



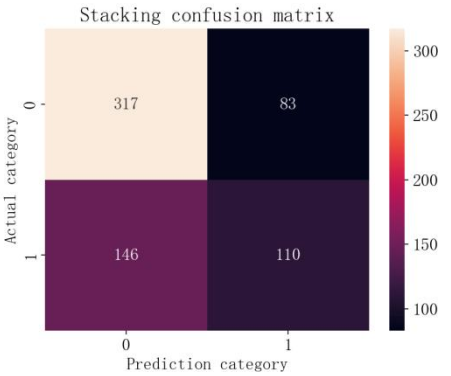


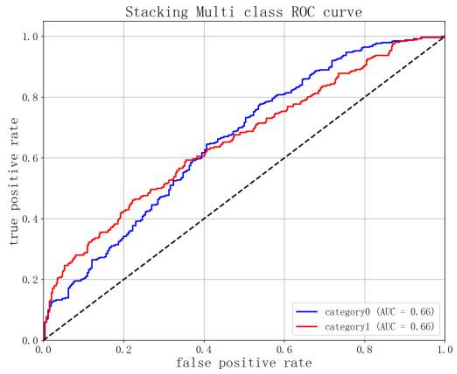
LightGBM Bagging



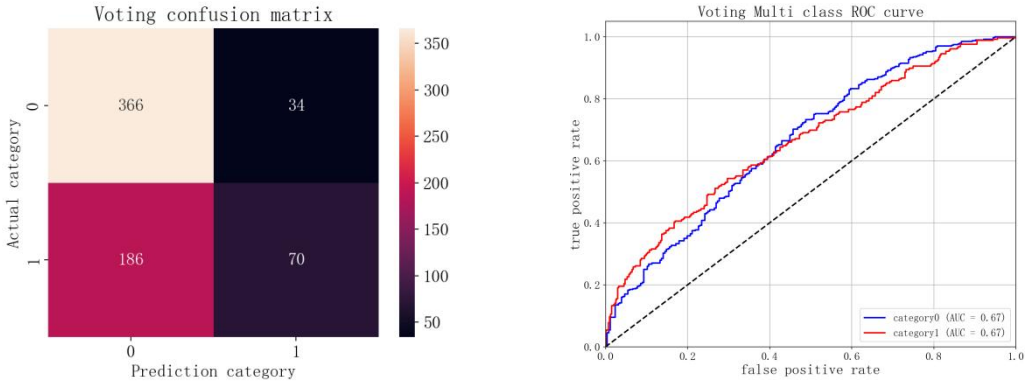


CatBoost Bagging

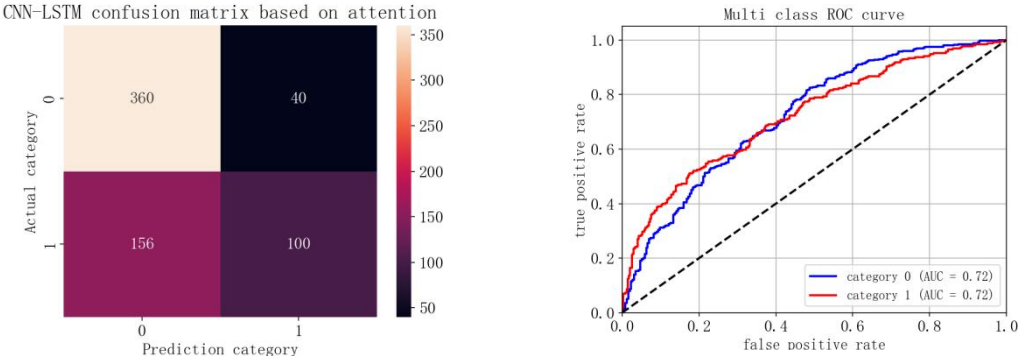




Stacking



Voting



Attention based CNN-LSTM

Figure 7 The confusion matrix and ROC curve of each model

5| Conclusion

This article analyzes the advantages and disadvantages of chlorine gas and chloramine disinfection methods, especially the hazards of trihalomethanes produced by chlorine gas disinfection, and clarifies the advantages of chloramine disinfection in reducing by-product generation and alleviating chlorine odor. Through the collection of water quality prediction data, this article conducted an in-depth analysis of the relationship between indicators such as chloramine and trihalomethanes and the drinkability of water quality. It was found that relying solely on these two indicators cannot accurately determine water quality. Therefore, this article introduces more indicator data and conducts comprehensive preprocessing and feature engineering on the data. Based on this, Stacking, Voting, and attention based CNN-LSTM classification prediction models were constructed, and the hyperparameters of each model were optimized through random search and cross validation. Finally, the performance of each model in water quality prediction was evaluated, verifying the effectiveness and accuracy of the combination of multiple indicators and

advanced models in water quality prediction, providing scientific basis and technical support for water quality safety monitoring.

References

[1] Huang Wei, Yang Wanli, Mei Yuqin, etc Control analysis of disinfection byproduct trihalomethanes in chloramine disinfection [J] Occupational and Health, 2019, 35 (22): 3071-3074.

[2] Julie Jie, He Kai, Huang Sheng, etc Development and Future Prospects of Water Quality Monitoring Technology Driven by Data Model Coupling [J/OL] People's the Pearl River: 1-15 [2014-06-10].

[3] Shang Xudong, Duan Zhongxing, Chen Bingsheng, etc Water quality prediction based on bidirectional long short-term memory network combination model [J/OL] Journal of Environmental Science: 1-10 [June 10, 2024].

[4] Xiao Yanglan, Shen Huirou, Xu Yihan, etc Water quality prediction of Minjiang River Basin based on GBDT-LSTM [J] Journal of Ecological Environment, 2024, 33 (04): 597-606.

[5] Fu Dunkai, Zhang Yunhui, Xu Xiaojun, etc Progress and Trends in Water Quality Prediction Research Based on Bibliometrics [J] East China Geology, 2024, 45 (01): 88-100.

[6] Zhang Shuyan, Chen Qibing, Cai Yijie Research on ARIMA Model Prediction of River Water Quality Based on Exponential Smoothing [J] Guangdong Chemical Industry, 2024, 51 (06): 95-98.

[7] Xiang Xinjian, Xu Honghui, Xie Jianli, etc Research on Water Quality Prediction Based on VMD-TCN-GRU Model [J] People's Yellow River, 2024, 46 (03): 92-97.

[8] Yang Zhenjian, Pang Ying Water quality prediction model based on GAT-BILSTM Res [J] Journal of Tianjin Chengjian University, 2024, 30 (01): 60-65.

[9] Xiao Mingjun, Zhu Yichun, Gao Wenyuan, etc Comparison of Water Quality Prediction Methods Based on Different Artificial Neural Networks [J/OL] Environmental Science: 1-10 [June 10, 2024].

[10] Xu Shengqiang Research on Water Quality Evaluation and Prediction of Handan Yuecheng Reservoir Based on BP Neural Network [J] Shaanxi Water Resources, 2024 (02): 104-105+108.

[11] Niu Jinghui Key data prediction algorithm for industrial wastewater quality based on GWO XGBoost [J] Industrial Water Treatment, 2024, 44 (01): 184-190.