applied sciences

MDPI

Article

High-Dimensional, Small-Sample Product Quality Prediction Method Based on MIC-Stacking Ensemble Learning

Jiahao Yu [1], Rongshun Pan [1] and Yongman Zhao [1,2,*]

[1] Department of Industrial Engineering, College of Mechanical and Electrical, Shihezi University, Shihezi 832003, China; 20202009046@stu.shzu.edu.cn (J.Y.); panrongshun@stu.shzu.edu.cn (R.P.)
[2] Department of Data Science and Big Data Technology, College of Information Science and Technology, Shihezi University, Shihezi 832003, China
* Correspondence: zhrym@shzu.edu.cn

Abstract: Accurate quality prediction can find and eliminate quality hazards. It is difficult to construct an accurate quality mathematical model for the production of small samples with high dimensionality due to the influence of quality characteristics and the complex mechanism of action. In addition, overfitting scenarios are prone to occur in high-dimensional, small-sample industrial product quality prediction. This paper proposes an ensemble learning and measurement model based on stacking and selects eight algorithms as the base learning model. The maximal information coefficient (MIC) is used to obtain the correlation between the base learning models. Models with low correlation and strong predictive power were chosen to build stacking ensemble models, which effectively avoids overfitting and obtains better predictive performance. To improve the prediction performance as the optimization goal, in the data preprocessing stage, boxplots, ordinary least squares (OLS), and multivariate imputation by chained equations (MICE) are used to detect and replace outliers. The CatBoost algorithm is used to construct combined features. Strong combination features were selected to construct a new feature set. Concrete slump data from the University of California Irvine (UCI) machine learning library were used to conduct comprehensive verification experiments. The experimental results show that, compared with the optimal single model, the minimum correlation stacking ensemble learning model has higher precision and stronger robustness, and a new method is provided to guarantee the accuracy of final product quality prediction.

Keywords: high-dimensional small sample; machine learning; MIC; OLS; MICE; combination characteristic; regression prediction; stacking ensemble model

Citation: Yu, J.; Pan, R.; Zhao, Y. High-Dimensional, Small-Sample Product Quality Prediction Method Based on MIC-Stacking Ensemble Learning. *Appl. Sci.* **2022**, *12*, 23. https://doi.org/10.3390/app12010023

Academic Editors: Kuen-Suan Chen, Kai-chao Yao, Mei-Ling Huang, Ching-Hsin Wang and Chun-Min Yu

Received: 30 November 2021
Accepted: 19 December 2021
Published: 21 December 2021

1. Introduction

In the eighteenth century, the first industrial revolution used steam as a source of power and brought major changes to industry. The second industrial revolution used electricity and assembly lines for mass production. The third industrial revolution witnessed the integration of information technology and computers in manufacturing. Now, with the proposal of "Industry 4.0" in Germany, the industry is currently undergoing the "fourth industrial revolution", marking the integration of processing equipment systems and data in the production process, which will take us to a new level [1]. Modern manufacturing enterprises focus their attention on the intelligent management of production, including the material supply chain, manufacturing process technology, intelligent warehousing, and quality control. Among these, quality control has always been a core concern. The rate of product renewal in modern manufacturing is accelerating, and large-scale and unitary product production cannot meet the differentiated needs of the market. Enterprises adapt to market changes while reducing operating costs to gain advantages in competition, and an increasing number of enterprises have established flexible production lines, seeking to build a small-batch production mode. According to recent statistics from the United States, Japan, and other countries, small and medium-sized enterprises represent 75% of all enterprises.

Multivarieties and small-batch production is rapidly becoming the main production mode, representing more than 90% of China's manufacturing industry. Multivarieties and small-batch production models require parallel production and production flexibility [2]. This production mode brings high-dimensional, small-sample quality information data. In the context of "Industry 4.0", it is necessary to further pursue high-quality output. However, in this production mode, with its continuous decline in single batch production, it is difficult to extract a large amount of quality information, quality data are relatively scarce, and the composition of the generated data information is also more complex. This production mode brings high-dimensional, small-sample quality information data.

In reference to the quality control of manufacturing processes in the literature, some researchers used data processing technology to expand the dataset, thereby expanding the sample size. Other researchers identified the quality of products by methods such as the multivariate control chart of T2 statistics [3], multivariate statistical process control (MSPC) [4], fuzzy autoregressive moving average (FARMA) [5], and multivariate exponentially weighted moving average (MEWMA) [6]. If there are systematic errors in the production process, the production process will be out of control, and the control chart will no longer be stable. The above methods have good results for specific products, but for high-dimensional, small-sample manufacturing processes, there is insufficient product data with similar characteristics. Different products have different influencing factors. The factors that determine quality are high dimensionality, multivariability, small sample size, time varying, and other characteristics that have greater uncertainty and limitations.

Artificial intelligence technology has brought a new development direction to the manufacturing industry and has attracted the attention of an increasing number of industrial enterprises. Intelligent manufacturing has gradually entered the stage of modern advanced manufacturing. Traditional statistical process quality-control technology has gradually added intelligent quality-control technology, especially prediction technology based on machine learning, which has become one of the key directions of current quality-control technology. Enterprises can collect multisource data from the production process, including data anomalies, processing parameters, and processing data. After the data is processed and analyzed, valuable information can be obtained from the manufacturing process, production system, and equipment. These technologies achieve advanced product manufacturing quality prediction, which replaces the previous postinspection method of production quality. It can discover and eliminate hidden quality hazards in advance and effectively reduces the cost of enterprise quality control. Manufacturing quality prediction also provides data support for reliability evaluation and parameter optimization, thereby improving the enterprise's intelligent management level. However, the actual use of these solutions requires a high degree of professional knowledge and computer-programming capabilities. The current small-batch production types and rapid changes have brought new challenges to the realization of accurate quality prediction.

With the rapid development and gradual maturity of artificial intelligence and machine learning, a new effective method, which uses a learning algorithm and theoretical development, is provided for research in high-dimensional, small-sample quality prediction. Supervised learning is most widely used method in machine learning, and it forms predictions through learning mapping [7]. Research on machine learning algorithms for quality regression prediction problems has become a great concern. The main task of the regression problem is to train the learner according to the existing data and map the input to the corresponding output result to achieve the purpose of prediction. GE and others analyzed the application of 10 kinds of supervised machine learning methods in the manufacturing process, including partial least square regression (PLS), random forest, artificial neural network (ANN), support vector machine (SVM), back propagation (BP) neural network algorithm, and decision tree [8,9]. Some researchers also use heuristic algorithms and machine learning model combination construction methods. Heuristic algorithms are used to find the best parameters of prediction models, mainly metaheuristic optimization algorithms, such as the particle swarm optimization algorithm (PSO) [10],

firefly algorithm (FA) [11], and genetic algorithm (GA) [12], to improve the predictive power of the model.

The abovementioned literature uses a single model for prediction and has achieved good prediction results. Considering that the specifications and types of high-dimensional, small-sample production mode and the production equipment used is diverse, diverse types of information are collected. In this case, quality regression prediction is more difficult, and single machine learning is prone to overfitting. Therefore, the problems of low estimation accuracy and weak model robustness cannot be avoided. Regression problems usually have very complicated internal and external factors, which will affect the prediction performance of different machine-learning algorithms to varying degrees.

In recent years, machine learning research based on ensemble learning has attracted the attention of the manufacturing industry and academia, and has become increasingly widely used in various fields, such as scientific research and data mining. The main task of ensemble learning is to train multiple learners on a dataset and then combine their respective predictions into the final result. Manuel et al. proposed a gaNet-C model based on the idea of integration model, which combined the excellent characteristics of a residual network model and a gradient enhancement model and had good convergence ability and regularization effect. It showed good predictive effect on both data [13]. The integrated method can take advantage of different models to obtain better prediction results. Ensemble learning has better generalization ability and stability, and different models with unique characteristics are combined to solve the limitations of a single model, thereby improving prediction performance [14]. However, the integration method of obtaining the prediction averages of multiple algorithm models or different parameter models of the same type of algorithm cannot reflect the differences in data observations of different prediction algorithms. Each algorithm cannot exert its own advantages to compensate for its own errors. This combination method does not have sufficient theoretical support, and the principle is relatively thin [15]. Common ensemble learning methods include parallelized ensemble bagging, serialized ensemble boosting, and multilayer classifier blending and stacking [16–18]. Blending and stacking use a high-level metamodel to synthesize the output features of a low-level base model to enhance generalization ability and obtain higher prediction accuracy. Xu et al. used random forest, adaptive booster, gradient booster decision tree, and other five models as the base model, and back-propagation neural network as the metamodel to build a stacking ensemble learning model for secondhand housing price prediction [19]. Yin et al. constructed four ensemble models using the stacking technique of ensemble learning, providing a high-performance prediction method for unbalanced rockburst prediction [20]. The above literature reflects the advantages of the stacking ensemble learning model in prediction. The prediction performance of the stacking model is better than that of a single machine learning model, and the model reduces the risk of falling into a local minimum. Some scholars choose base models in other fields by calculating error correlation. The stacking ensemble learning model constructed by this method has better prediction performance than random selection, which provides a new method for the selection of basic models. Dong et al. used the Spearman correlation coefficient to select the base model and built a stacking ensemble model, which can provide higher prediction accuracy in the case of intermittency and volatility of wind power [21]. Shi et al. used the Pearson correlation coefficient of the two-dimensional vector as the correlation index to select the base learner, which performed better than the stacking model of randomly selecting the base learner [15]. The abovementioned literature constructs a stacking ensemble learning model through the correlation selection base model, which effectively improves the prediction performance of the model. Therefore, in order to optimize the performance of the stacked ensemble model, the learning ability and correlation degree of each basic learner must be analyzed. In this study, MIC was used to improve the selection method of the basic model, which could effectively identify linear and nonlinear relationships and make the comprehensive results of the prediction model more robust and accurate.

Sample size and feature dimension have certain influence on prediction. It must be considered that we are dealing with high-dimensional, small samples of data, of which only a small number of samples are available. Seven features and 103 samples produce a scene that is prone to overfitting. Some scholars have conducted research on this issue. Robert et al. used the averaging-ensemble of randomly projected Fisher linear discriminant classifiers method to adopt a complex covariance regularization scheme, avoiding the partial variance decomposition method, and improving the prediction performance of high-dimensional, small samples [22]. Manuel et al. proposed a random 2D-CNN model, which is based on a collection of random learning blocks. The learning block composed of 2D convolutional layers, batch-normalization layers, and max-pooling layers reduces the possibility of overfitting. In terms of predicting Alzheimer's disease (high-dimensional, small sample), it demonstrates the advantages of the model and its practical application value. Avoiding overfitting, this model can effectively obtain the best training effect in a limited sample [23]. The above documents effectively solve the problem of high-dimensional, small samples prone to overfitting. The stacking ensemble learning model established in this article trains primary learners on the first-layer basic model through k-fold cross-validation and selects a model with strong generalization ability in the second layer to correct the learning of multiple basic learners in the first layer deviation. Moreover, none of the data blocks predicted by the primary learner participate in the training of the learner, so that all data is used only once during model training, thereby reducing the occurrence of overfitting and avoiding model deviation caused by a single algorithm.

Hawkins defines an outlier as "an observation that deviates so much from other observations that it arouses suspicion that it is caused by a different mechanism [24]." Outliers will adversely affect the fitting of the regression model, thereby making relevant statistical inferences invalid. Therefore, it is very important to perform outlier detection in the dataset [25].

Abnormalities and process fluctuations also occur in normal processes during operation. In actual industrial production, with continuous production of data in the manufacturing process, abnormal data values and data fluctuations are inevitable, and noise is common. In addition, the processing conditions of high-dimensional, small-sample processing methods are more complex and changeable, and abnormal data outliers and equipment offset errors often occur. These factors cause the machine learning process to be affected by abnormal values and affect the final quality prediction effect. In this work, we propose an outlier replacement method based on box graph technology and MICE [26,27]. This method is not affected by data types and has little influence on the overall distribution of data. Then, ordinary least squares (OLS) is used to identify the abnormal points (strong influence points) of the original data correction, integrating the four methods of studentized residual [28], leverage [29], Cook's distance [30], and DFFITS [31] to determine abnormal samples. In the feature-engineering stage, the CatBoost algorithm is used to construct combination features, and strong combination features are selected as new features to construct new feature sets. In order to improve the prediction ability of machine learning, Box–Cox [32] was used to normalize features. To ensure the generalization of the established model, the data features are not reduced in dimension.

This paper studies the machine learning regression prediction model of various types of cutting-edge algorithms as the base learning model and establishes a high-dimensional, small-sample stacking ensemble learning [33] regression prediction model for high-dimensional, small-sample industrial production. The MIC [34] was used to calculate the error correlation degree and screen the base model with large difference degree, and it makes the best use of the different observations of data in different models, and also avoids the repeated learning of similar models, leading to the occurrence of overfitting. According to the stacking model framework-based learning mode, the parameters are selected by the combination of random search and grid search algorithm [35]. The minimum correlation stacking ensemble learning regression quality prediction model is constructed. The goals and contributions of this article are (i) in the Industry 4.0 era, it has become

easy to obtain high-dimensional multivariate data, and the number of samples may be limited, and a theoretical framework is proposed for industrial product quality prediction in high-dimensional and small-sample scenarios; (ii) there are outliers in the data collected in the industrial process, and an outlier processing method combining multiple algorithms is constructed. This method can effectively reduce the destructive influence of abnormal values on the regression quality prediction results in the actual process; (iii) CatBoost's greedy strategy constructs combined features, and creates a new feature set with strong combined features and original features to improve the accuracy of quality prediction; (iv) in the stage of model building, a variety of methods are used to combat overfitting, and applying MIC to the selection of the basic model of the stacking ensemble learning model can effectively identify linear and nonlinear relationships, more comprehensively select models with lower correlation, avoiding repeated learning of similar models, and provide a new method for basic model screening; (v) to verify that the minimum correlation stacking ensemble model can have better prediction performance in high-dimensional, small-sample regression quality prediction scenarios. Providing accurate quality prediction is very important for industrial processes. It can avoid additional economic loss factors for enterprises.

2. Materials and Methods

2.1. UCI Concrete Slump Data

This article uses the concrete slump data in the UCI machine learning library as the experimental dataset. Because concrete is a very complex material, the product easily loses control, resulting in poor final quality, and only at the end of the 28th day can we understand its quality. The final quality of cement is affected by seven characteristics: slag, fly ash(ash), water, superplastic, coarse aggregate(coarseagg), and fine aggregate(fineagg). The dataset includes 103 data points with 7 input variable features and 3 output variables. Some data and characteristic information of the concrete slump dataset are shown in Tables 1 and 2.

Table 1. Information of Concrete Slump dataset.

The Dataset	The Input Variable	The Output Variable	Number of Samples
Concrete Slump	Cement Slag Fly ash Water Superplastic Coarse Aggr Fine Aggr	Slump (cm) Flow (cm) 28-day Compressive Strength (MPa)	103

Table 2. Part of the data: here are the first three samples and the last three samples.

Sample	Cement	Slag	Fly Ash	Water	Superplastic	Coarse Aggr.	Fine Aggr.	Strength
1	273.00	82.00	105.00	210.00	9.00	904.00	680.00	34.99
2	163.00	149.00	191.00	180.00	12.00	843.00	746.00	41.14
3	162.00	148.00	191.00	179.00	16.00	840.00	743.00	41.81
...	...	...	...	...	...	...	...	...
101	258.80	88.00	239.60	175.30	7.60	938.90	646.00	50.50
102	297.10	40.90	239.90	194.00	7.50	908.90	651.80	49.17
103	348.70	0.10	223.10	208.50	9.60	786.20	758.10	48.77

2.2. Data Analysis

With the progress of manufacturing and control technology, the processing process has become increasingly stable and controllable. The normal distribution process has become

the main distribution form of processing data in the production and assembly processes of the manufacturing industry. The way in which the processing data obey the normal distribution is called the normal process. Actual industrial data generally approximate a normal distribution. The three output variables were analyzed through density figures and quantitative quantile plots (see Figure 1). If the data does follow a normal distribution, the points on the quantitative quantile plots will fall roughly on a straight line. Compared with the other two output parameters, slump (cm) and flow (cm), 28-day compressive strength (strength) is more in line with the normal distribution and fits the product quality in high-dimensional, small-sample actual production. Finally, 28-day compressive strength is selected as the output variable of this article. The concrete slump data is complete.

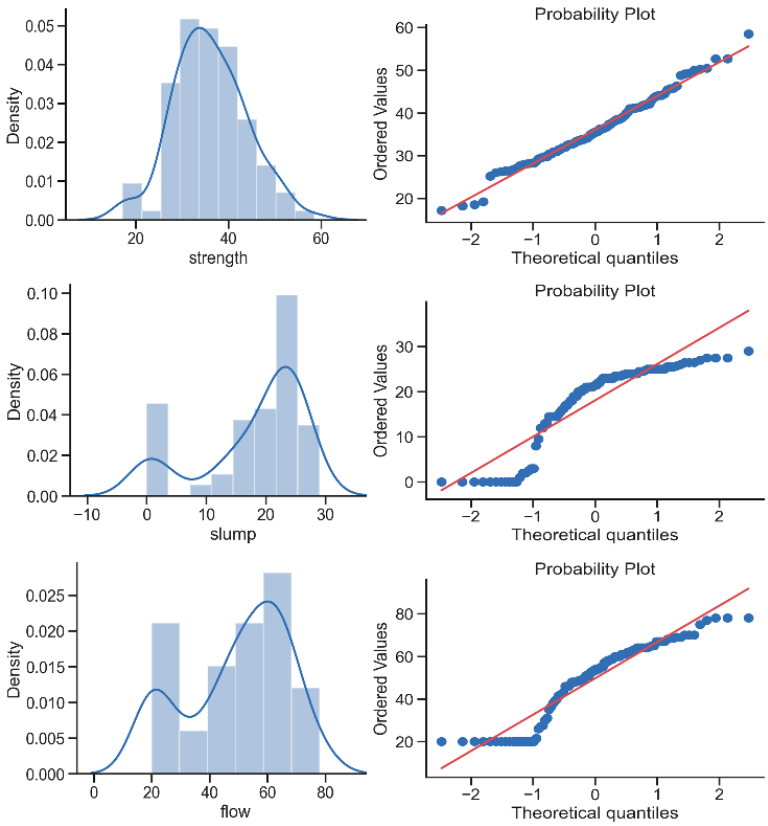

Figure 1. Output variable distribution status.

2.3. Outlier Detection and Replacement

These anomalies/outliers are often ignored or discarded as noise. Some existing machine learning and data mining algorithms consider outliers but only consider the calculation of outliers when any algorithm should do so [36]. In addition, there are fewer samples of high-dimensional, small-sample data. To fully tap the value of the data, this article replaces outliers instead of removing outliers from the dataset. It is necessary to make the filling value close to the true value of the data as much as possible; otherwise, it may cause deviation in the analysis of the filled dataset.

2.3.1. Box Plot Theory

This article first uses box plots to identify outliers. The 3σ criterion assumes that the data obeys the normal distribution, but the actual data often does not strictly obey the normal distribution. The standard for judging outliers is based on calculating the mean and standard deviation of the data. The resistance of the mean and standard deviation is extremely small, and the outliers themselves will have a greater impact on them, so the number of outliers judged is very small. Box plot theory does not use the mean and variance as the basis to determine the abnormality of the data and does not need to assume that

the data obeys a specific distribution. It can intuitively describe the discrete distribution of the data. The box plot uses quartiles and interquartile ranges. It has strong resistance and is not easily disturbed by abnormal data, which provides a preliminary standard for identifying abnormal values (see Figure 2).

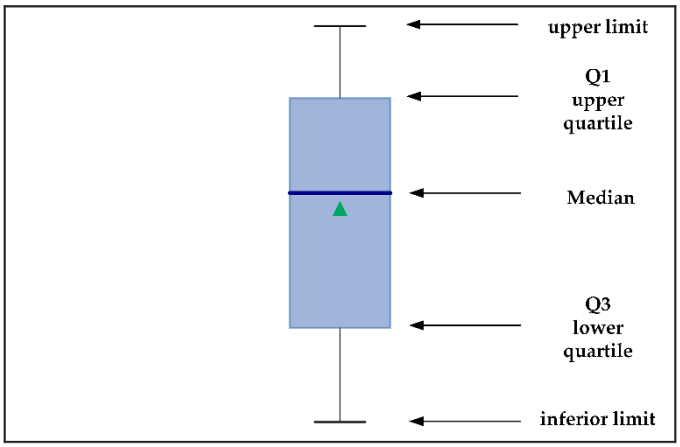

Figure 2. Box plot: the green triangle is the sample median.

In the boxplot, Q_1 is the lower quartile (25% quartile), Q_3 is the upper quartile (75% quartile), and the interquartile range (IQR) refers to the difference between the 25% and 75% quartiles.

The formula for calculating IQR is as follows: $IQR = Q_1 - Q_3$, where minimum: $Q_3 - 1.5 \times IQR$, maximum $Q_1 + 1.5IQR$.

The criteria for determining outliers are:

$$\text{Outlier} = \text{value} < (Q_3 - 1.5IQR) \text{ or value} > (Q_1 + 1.5IQR)$$

2.3.2. MICE Theory

This paper addresses outliers of small-sample data of many varieties and small batches. First, a box was used to eliminate outliers, and then the MICE method was used to fill in the outliers. The replacement of the abnormal outlier values should make the filling value close to the true value to avoid deviation of the analysis result of the original dataset when analyzing the filled dataset. At present, most of the filling methods use single-value interpolation, average, median, etc., but it is very likely that the original distribution of the data will be changed, which will have a greater impact on the mean and variance of the data and increase machine learning. The error cannot adequately reflect the uncertainty of the value that needs to be replaced and this affects the later prediction. MICE is proposed by Buuren to define a predictive model for each variable with missing data [37,38]. The basic idea is as follows:

- Use initial estimates to fill in missing values by randomly sampling from existing observations.
- For each missing value, use the observed parts of other variables as predictors to estimate the regression model.
- Replace the missing values with randomly drawn values from the posterior prediction distribution of the results. Using the observed and most recent estimates as the predicted value, this process is repeated for each variable with missing data.

Given the following dataset with p incomplete variables $X = X_1, X_2, \ldots, X_p$, the observed part of X is $X^{obs} = X_1^{obs}, X_2^{obs}, \ldots, X_3^{obs}$, and the missing part of X is $X^{miss} = X_1^{miss}, X_2^{miss}, \ldots, X_p^{miss}$, where $\theta_1, \theta_2, \ldots, \theta_p$ describes the conditional density parameter of $X_1, X_2, \ldots X_p$. Interpolation for each incomplete variable in $(i = 1, 2, \ldots, p)$ is iteratively drawn. First, the random extraction of parameters is simulated, and then the random extraction of the missing values in the variables is simulated. The posterior

distribution of the missing variable θ is established by Gibbs iterative sampling of the observation data. The variable X_{obs} is used as a covariate to perform regression modeling on other variables with missing values X_{miss}. For all variables with missing values, this process is repeated, that is, X_1 to X_p need to be interpolated for each iteration. At the end of each iteration, missing values are replaced by predicted values from the regression model. The description is shown in the equation:

$$\begin{aligned} &\theta_1^{*(t)} \sim P(\theta_1 | X_1^{obs}, X_2^{t-1}, \ldots, X_p^{t-1}) \\ &X_1^{*(t)} \sim P(X_1 | X_1^{obs}, X_2^{t-1}, \ldots, X_p^{t-1}, \theta_1^{*(t)}) \\ &\vdots \\ &\theta_p^{*(t)} \sim P(\theta_1 | X_p^{obs}, X_1^{t-1}, \ldots, X_{p-1}^{t-1}) \\ &X_p^{*(t)} \sim P(X_p | X_p^{obs}, X_2^{t-1}, \ldots, X_{p-1}^{t-1}, \theta_p^{*(t)}) \end{aligned}, \quad (1)$$

where t is the number of iteration cycles $t \in (1, 2, \ldots, n)$, $\theta_1^{*(t)}, \theta_2^{*(t)}, \ldots, \theta_p^{*(t)}$ is the random drawing parameter of $\theta_1, \theta_2, \ldots, \theta_p$ during t iterations, and $X_1^{*(t)}, X_2^{*(t)}, \ldots, X_p^{*(t)}$ is the estimated value of $\theta_1^{miss}, \theta_2^{miss}, \ldots, \theta_p^{miss}$ in iteration t. The observed value X_{obs} will not change in the iterative update process, but the missing data X_{miss} will be updated in each iteration.

2.3.3. OLS Outlier Identification Method

When there are outliers in the regression analysis, these outliers will directly affect the fit and the stability of the model. When the data is small and there are outliers in the data, the outliers will produce heteroscedasticity. The outliers will pull the regression line closer to them, resulting in greater "distortion" of the regression line, and the resulting regression model has lower prediction accuracy. After using the box plot to eliminate outliers, the next step is to use OLS to build statistics to establish a data deletion model for a single outlier and determine the four outliers by using studentized residuals, leverage values, DFFITS, and Cook's distance. Each method was used to identify and eliminate outliers and strong influence points. Using a single judgment to investigate is sometimes inaccurate, and often overlooks the comprehensive application of several aspects. Here, studentized residuals are selected, and the heteroscedasticity is considered compared with residuals. It is more effective to detect outliers and improve the accuracy of regression prediction.

- Leverage value:

  Outliers in the x-space are called high leverage points. The linear model is expressed as:

$$Y = Xb + e, \quad (2)$$

where X is the $n \times (p+1)$ observation matrix, Y is the observation vector of $n \times 1$, b is the unknown parameter vector, and e is the random error vector.

When OLS is used for regression estimation, hat matrix H can be obtained, which can be expressed as

$$H = X(X'X)^{-1}X', \quad (3)$$

Many researchers diagnose lever points through the diagonal element h_{ii} of hat matrix H to determine the possible influence points, especially the point $h_{ii} > 2(p+1)/n$. Therefore, $h_{ii} > 2(p+1)/n$ is used as the diagnostic benchmark of the diagnostic lever point, where p is the number of features and n is the number of samples. The variable h_{ii} can be expressed as

$$h_{ii} = (HH)_{ii} = \sum_{j=1}^{n} h_{ij}h_{ji} = \sum_{j=1}^{n} h_{ji}h_{ij}, \quad (4)$$

- Studentized residual

The internal studentized residual t_i is the mean square error of the regression model obtained by fitting all data:

$$\begin{aligned} t_i &= \frac{r_i}{s\sqrt{1-h_{ii}}} \\ s &= \frac{Y-X\hat{b}}{\sqrt{n-p-1}} \end{aligned}, \tag{5}$$

where r_i is the residual. When the measurement error of y_i is independent and obeys a normal distribution $N(0, \sigma^2)$, the internal studentized residual can be obtained.

The external studentized residual, the mean square error of the regression model, is obtained by fitting all other data with the i-th observation value deleted:

$$t(i) = \frac{r_i}{s(i)\sqrt{1-h_{ii}}}, \tag{6}$$

where $s(i)$ is the estimate of σ for the entire regression run without the ith observation. $t(i)$ are also known as external studentized residuals.

The internal studentized residual t_i obeys the n distribution of degrees of freedom $(n-p-2)$, and the external studentized residual $t(i)$ obeys the t distribution of degrees of freedom $(n-p-3)$. Approximately 95.44% of the studentized residual falls in the interval $[-2, 2]$ and does not show any trend. If the studentized residuals exceed this range, these points may be outliers.

- DFFITS

Welsch–Kuh distances (DFFITS) are proposed from the viewpoint of data fitting. Considering the influence of before and after deletion of the ith point on the fitting value at x_i, DFFITS is defined as

$$DFFITS(i) = t(i)\sqrt{\frac{h_{ii}}{1-h_{ii}}}, \tag{7}$$

Belsley uses the point of $DFFITS \geq 2\sqrt{\frac{p+1}{n}}$ as the threshold for determining outliers.

- Cook's distance

Cook's distance is a statistic that measures the influence of the ith observation point on regression. For each observation point, the Cook's distance is defined as

$$D_i(M, C_o) = \frac{(\hat{\beta}_{(i)} - \hat{\beta})^T M(\hat{\beta}_{(i)} - \hat{\beta})}{C_o}, i = 1, 2, \cdots, n, \tag{8}$$

where $M = X^T X$ is the deviation matrix of the observed data, $C_o = p\hat{\sigma}^2$ is the mean square error, and n is the sample size. When $|D_i| > 4/n$, it is judged as the strong influence point.

2.4. Combination Feature Construction and Assessing

Prokhorenkova et al. refer to these interactions as "feature combinations". New feature combinations can be constructed, including digital features and classification features [39]. Another useful feature that CatBoost adds to GBDT is its support for combined interactive features. CatBoost will adopt a greedy strategy to consider combinations [40]. The first split in the tree does not consider any combination, but for the second, and all subsequent splits, CatBoost will combine all preset combinations with all features in the dataset [41].

Aiming at the high-dimensionality and small-sample data of multiple varieties and small batches, this paper explores the effect of combined features on improving the quality of prediction accuracy and uses the function of CatBoost feature combination to identify important combined features to obtain new powerful features and achieve feature expansion. However, it is impossible to construct all combinations as new features, which will cause problems with feature dimensions. Finally, the median and mean values of the

combined features that have the highest impact on algorithm prediction are selected as the new features.

2.5. Box–Cox Transform

The Box–Cox transformation is usually used to make the data characteristics conform to the normal distribution so that the high-dimensional, small-sample data meet the homoscedasticity as the basic assumption of the regression model, which can reduce unobservable error and the correlation of the predictive variables to a certain extent. It is more in line with the actual production process data form. The formula is as follows:

$$\widetilde{y}(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & \lambda \neq 0, \\ \ln(y), & \lambda = 0 \end{cases}, \tag{9}$$

The skewness value of each feature was calculated, and the Box–Cox transformation was used on the features of $Skewness > 0.5$ and $Skewness < -0.5$. This paper uses the square root transformation ($\lambda = 0.5$).

2.6. Machine Learning Algorithms

Aiming to address the multivariety, small-batch, high-dimensionality, small-sample regression prediction problem, a model fusion method based on the stacking algorithm is proposed. The basic learners used are linear regression [42], SVR [43], GBDT [44], XGBoost [45], ExtraTrees [46], random forest [47], KNN [48], and CatBoost [49].

2.6.1. Linear Regression

The purpose of linear regression is to find a hyperplane with the smallest variance. Each sample data has n features, where each feature corresponds to a weight value and the product of the weight and the weight plus a bias value:

$$y = \sum_{i=0}^{n} \omega_i x_i + b, \tag{10}$$

where ω is the weight value of each feature and b is the offset value.

2.6.2. SVR

SVR is a branch of SVM's regression application. The basic principle is to map the input space to a high-dimensional space through nonlinear mapping. The support vector machine structure regression function is obtained by minimizing the following objective function:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*), \tag{11}$$

where ξ_i and ξ_i^* are two relaxation variables, and C is the penalty factor. Constraints are added:

$$\begin{aligned} & y_i - w(x) - b \leq \varepsilon + \xi_i^* \\ & (w, \theta(x)) + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned}, \tag{12}$$

where ε is the loss function parameter.

2.6.3. GBDT

GBDT is an integrated learning model in which is a strong learner is obtained by combining multiple weak learners according to a certain combination strategy. Its weak learner model is a regression tree model, which connects multiple regression trees in series and uses the cumulative result of all trees as the final result. The regression tree

is initialized, and then it continues to iterate into more regression trees. The final strong learner is a combination of all trees. The negative gradient of the loss function is used to fit the approximate value of the current round of loss. The negative gradient of the loss function is shown in the formula:

$$r_{ti} = -\frac{\partial L(y_i, h_{t-1}(x_i))}{\partial h_{t-1}(x_i)}, \tag{13}$$

where $i = 1, 2, \cdots, m$, and t is the number of iterations. The base learning is initialized:

$$h_o(x) = \text{argmin}_c \sum_{i=1}^{m} L(y_i, c), \tag{14}$$

where the regression tree is fitted, its corresponding leaf node region is $R_{tj}, j = 1, 2, \cdots, J$, and J is the number of leaf nodes. The best fitting value is calculated:

$$c_{tj} = \text{argmin}_c \sum_{x_i \in Rtj} L(y_i, h_{t-1}(x_i) + c), \tag{15}$$

The updated stronger learner is

$$h_t(x) = h_{t-1}(x) + \sum_{j=1}^{J} c_{ij} I(x \in R_{tj}), \tag{16}$$

Finally, the strong learner is obtained:

$$H(x) = h_o(x) + \sum_{t=1}^{T} \sum_{j=1}^{J} c_{ij} I(x \in R_{tj}), \tag{17}$$

2.6.4. XGBoost

XGBoost is one of the representative algorithms of boosting ensemble learning, which is derived from GBDT [50]. The tree integration model constructed by K additive equations is defined as follows:

Given the dataset

$$D = \{(x_i, y_i)\} (|D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}), \tag{18}$$

$$\hat{y}_i = \phi(x_i) = \sum_{K=1}^{K} f_k(x_i),\ f_k \in F, \tag{19}$$

where $F = \left\{ f(x_i) = \omega_{q(x)} \right\} (q : \mathbb{R}^m \to T, W = (\omega_1, \ldots, \omega_T) \in \mathbb{R}^T)$ is the regression tree space. Each additional tree is equivalent to adding a new function to the model to fit the residual of the previous prediction. The regularized objective is minimized. The loss function is

$$\begin{array}{l} L(\phi) = \sum\limits_i l(\hat{y}_i, y_i) + \sum\limits_k \Omega(f_k) \\ \Omega(f) = \gamma T + \frac{1}{2}\lambda \|W\|^2 \end{array}, \tag{20}$$

where l represents the differentiable convex loss function, which is used to measure the difference between the predicted value $\hat{y}_i$ and the straight real value y_i, and Ω represents the penalty term of model complexity.

2.6.5. ExtraTrees

ExtraTrees is an ensemble learning algorithm that contains many decision trees. The biggest feature of this algorithm is that a random method is adopted in the selection of split features, and a certain value of a feature is randomly selected as the split point of

the feature [51]. Each tree uses the entire training sample to reduce bias. ExtraTrees has a strong randomness for the acquisition of split features and split values. When processing categorical feature attributes, samples with certain categories are randomly selected as the left branch, and samples with other categories are selected as the right branch. When processing numerical feature attributes, one characteristic attribute value that is between the maximum and minimum values of the feature attribute is randomly selected. When the characteristic attribute value of the sample is greater than this value, it is regarded as the left branch, and when it is less than this value, it is regarded as the right branch.

From $h\{(x,\theta_t), t = 1,2,3,\cdots,n\}$ set of regression decision subtrees, the mean value of each decision tree is used as the final regression prediction value:

$$\overline{h}(x) = \frac{1}{N}\sum_{t=1}^{N}\{h(x,\theta_t)\}, \tag{21}$$

2.6.6. Random Forest

Random forest(RF) is one of the representative algorithms of bagging ensemble learning, which generates random forest from multiple decision trees [52]. The core idea of random forest is to bootstrap the training set to form multiple training subsets, and the combination model is composed of random vectors (regression tree) $\{h(X,\theta_i), k = 1,\ldots,p\}$. The model uses numerical variables as predictors to generate multiple nonlinear regression radio frequency models. The decision trees $h(X,\theta_i)$ form the predictions of the model relative to k averages. In the regression function $E_\theta h(X,\theta)$, x represents the input, y represents the output, and θ is the random vector representing the node characteristics of the decision tree.

2.6.7. KNN

The main purpose of KNN is to use known labels to calculate the distance between an object and its neighbors [53]:

$$d(x,y) = \left(\sum_{i=1}^{q}|x_i - y_i|^p\right)^{1/p}, \tag{22}$$

where x is an object, and y is a neighboring object with a known label. KNN continues to assign the label with the highest frequency among the selected k nearest neighbors to the object. The number of nearest neighbors (k) has a significant impact on the classification results of the model.

2.6.8. CatBoost

CatBoost is an improved GBDT toolkit similar to XGBoost. In the process of using the CatBoost algorithm to train the data, the latter tree adjusts its weights according to the previous tree so that the tree with the larger residual is assigned a larger weight. Finally, multiple weak regressors are integrated to form a strong regressor. CatBoost solves the problem of gradient deviation and prediction bias. CatBoost has the following advantages [40,54]: CatBoost has two implementations: CPU and GPU. The GPU implementation allows faster training and is faster than GBDT, XGBoost, and LightGBM on similarly sized sets.

Target statistics (TS) is a very effective method. In the decision tree, the average value of the labels will be used as the criterion for node splitting. This method is called greedy target-based statistics, or greedy TS for short. CatBoost uses a standard method of improving greedy TS, adding a prior distribution item to reduce the influence of noise data on the data distribution:

$$\widehat{x}_k^i = \frac{\sum_{j=1}^{p-1}[x_{\sigma_{j,k}} - x_{\sigma_{p,k}}]Y_{\sigma_j} + ap}{\sum_{j=1}^{p-1}[x_{\sigma_{j,k}} = x_{\sigma_{j,k}}] + a}, \tag{23}$$

where p is the added prior term and a is usually a weight coefficient greater than 0. For regression problems, CatBoost performs a random arrangement of the dataset, with the prior term taking the mean of the dataset labels.

CatBoost replaces the traditional gradient boosting algorithm with the ordered boosting algorithm. This algorithm can effectively handle the noise points in the training set to avoid the gradient estimation deviation, solve the inevitable gradient deviation in the iterative process, and improve the model generalization ability. Oblivious trees were used as base predictors. In such a tree, the segmentation used at each level of the tree must be consistent. Therefore, the structure of the tree is balanced, the unbiased estimation of the gradient is obtained, and then the gradient descent is carried out, thus alleviating the overfitting phenomenon. The pseudocode is shown in Algorithm 1.

Algorithm 1: Ordered Boosting

Input : $\{(X_k, Y_k)\}_{k=1}^{n}$ ordered according to, the number of trees I
$\sigma \leftarrow$ random permutation of $[1, n]$
$M_i \leftarrow 0$ for $i = 1 \ldots n$
for $t \leftarrow 1$ to I do
  for $i \leftarrow 1$ to n do
    $r_i \leftarrow y_i - M_{\sigma(i)-1}(x_i)$;
  for $i \leftarrow 1$ to n do
    $\Delta M \leftarrow$
     $LeaenModel((x_j, r_j) :$
     $\sigma(j) \leq i)$;
    $M_i \leftarrow M_i + \Delta M$;
return M_n

To obtain an unbiased gradient estimate, CatBoost will train a separate model M_i for each sample x_i, which is obtained by using a training set that does not contain sample x_i. The model M_i is used to obtain the gradient estimate of the sample, and the gradient is used to train the base learner and obtain the final model. In CatBoost, a random arrangement of training data is generated. By sampling random permutations and obtaining gradients based on them, multiple permutations will be used to enhance the robustness of the algorithm. These arrangements are the same as those used to calculate classification feature statistics. To train different models, different permutations will be used, so using several permutations will not lead to overfitting.

2.7. Blending Ensemble Learning Model Algorithm Principles

The blending ensemble learning method considers heterogeneous weak learners [55]. Disjoined datasets were used for the training process of different model layers. Either multiple homogeneous or heterogeneous models are chosen as the basic model. Then, the training set is used to train these models, verify the trained model on the verification set, and obtain the predicted features as the second-level training set. The training set of the second layer of the blending model comprises the predicted features obtained from the first layer. The testing process of the blending model is divided into two layers. In the first layer, the trained model is used to predict the test data to obtain the prediction features of the test set; in the second layer, the prediction features are predicted to obtain the final prediction result.

The general steps for blending models are as follows:

- The original dataset is divided into training and testing sets.
- For each base model at the first level, the k-fold cross validation method is implemented to train and output t, the prediction of validation set and test set.
- The prediction set of the validation set of each basic model are combined together as the new meta-training set and the prediction set of the test set are combined together as the new meta-testing set.

- The new meta-training set is adopted to train the meta-learner at second floor and to output the final prediction results.

2.8. Stacking Ensemble Learning Model Algorithm Principles

This paper proposes a prediction method based on multiple models under the stacking ensemble learning architecture. The predictive power of a single primary learner is the basis of the predictive power of the ensemble model [15]. At the same time, there should be a certain degree of difference between the primary machine learning models, allowing the models to make up for each other. The stacking algorithm can integrate the advantages of different learning models [56]. Single machine learning algorithms cannot effectively map the relationship between the quality of high-dimensional, small-sample products and influencing factors in multiple dimensions and are easily affected by the amount of dataset and model parameters. Therefore, the stacking model ensemble learning algorithm is used to map this association. Whether regarding the improvement of model prediction accuracy, generalization ability, or robustness, the stacking model is more effective than a single model.

The stacking algorithm was first proposed by Wolpert in 1992 [57]. In 1996, Breiman introduced stacked effects, a scheme for estimating (and then correcting) generalization errors [58].

The structure of the stacking model is generally two layers. The first layer is generally called the primary learner, and the second layer is generally called the meta-learner. First, the original dataset is divided into several subdatasets: the subdatasets are input into the first layer to train multiple different primary learners, and each primary learner in the first layer outputs its own prediction result. Then, we construct the prediction result of the primary learner as the new input feature and combine the original feature as the input of the second layer, train the meta-learner of the second-layer prediction model, and then output the final prediction result from the second-layer model. In this process, overfitting may occur. To reduce the risk of model overfitting, K-fold cross-validation [59] is used to train the primary learner. Taking the five-fold cross-validation stacking model as an example, the algorithm principle is shown in Figure 3.

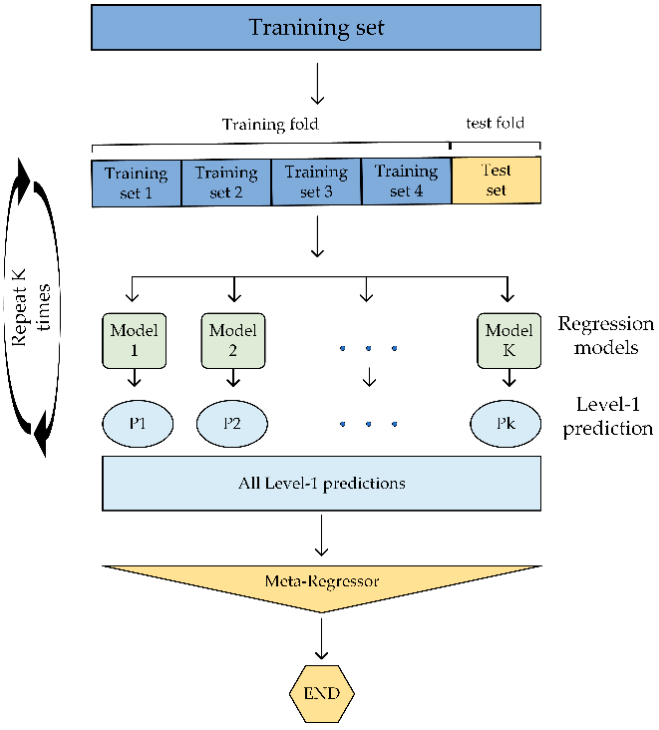

Figure 3. Stacking model structure.

This article analyzes the individual predictive ability of each elementary learner and comprehensively considers the way that each elementary learner observes the data space. In the optimization process of single-model training, the model often has the risk of falling into a local minimum. Combining multiple strong models can reduce the risk of falling into a local area.

Considering the single predictive ability of the primary learner, in the first layer of the stacking model, this paper uses eight machine learning models optimized through hyperparameter grid search as the selection range of the base learner. The stronger the learning prediction ability of the base model, the more effectively the overall prediction effect of the model can be improved. The second layer selects a model with strong generalization ability, corrects the bias of the multiple primary learner learning algorithms of the first layer to the training set, and prevents the overfitting effect from appearing through the aggregation method.

The specific training methods of high-dimensional, small-sample stacking regression prediction ensemble learning model are as follows. The dataset is $S = \{(x_i, y_i), i = 1, 2, \cdots N\}$, where x_i is the eigenvector of the ith sample and y_i is the predicted value of the ith sample. First, for the first layer prediction algorithm containing K primary learners, the k-fold cross-validation algorithm is used to divide the training dataset into K equal subsets. These subsets, which do not overlap with each other, are $S_1, S_2, \ldots, S_K$. A subset is selected as the test set and another K-1 subset as the test training set. The variables S_K and S_{-K} are defined as the k-th folding test set and training set, respectively, in the k-folding cross validation. The base model $L_1, L_2, \ldots, L_K$ is obtained after the K primary learners in the first layer train and learn the training set S_{-K}. According to the feature vector of each sample (x_i, y_i) in the test set, the K primary learners obtain the predicted value $p_{1,n}, p_{2,n}, \ldots, p_{K,n}$, where $n = 1, 2, \ldots, N$. The predicted value obtained is combined with y_i to obtain the dataset $S_{combine} = \{y_i, p_{1,n}, p_{2,n}, \ldots, p_{K,n}\}$, which is used as the input data of the second-level meta-learner. Then, combined with the k prediction results obtained from the test set training, the average value of the k prediction results of the test set is taken as the test set data of the second-level meta-learner. In this way, the transformation process of all data from input features to output features is realized, and the data blocks predicted by the primary learner are not involved in the training of the learner. Such a configuration means that all data is used only once in model training, effectively preventing the occurrence of overfitting.

2.9. Evaluation Criteria

After using the machine learning model for prediction, it is necessary to propose criteria for evaluating the model. Different models must adopt different evaluation criteria. To evaluate the prediction results, this study uses three performance indicators: root-mean-square error (RMSE), mean absolute error (MAE), and goodness of fit statistic (R^2). Among them, the root-mean-square error (RMSE) is the degree of change in the evaluation data, the expected value between the predicted value and the target value, and the most representative evaluation index in the regression model. The smaller the mean square error value is, the higher the accuracy of the prediction result and the better the model performance. The average absolute value error, the average of the absolute value of the deviation of all individual observations from the arithmetic mean, better reflects the actual situation of the predicted value error. The square correlation coefficient R^2 determines the percentage of the dependent variable change in the regression model and describes how well the regression curve fits the real data points. Between the values [0, 1], the closer R^2 is to 1, the higher the prediction effect, as shown in the following formula:

$$\begin{array}{l} RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \\ MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \\ R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \end{array}, \quad (24)$$

where y_i is the true value of each substance and $\hat{y}_i$ is the predicted value of the true value.

2.10. Data Standardization

To solve the different dimensions of variable features in different dimensions, one should improve the training speed to a certain extent, and to ensure that the experimental results are more intuitive and accurate, it is necessary to standardize the data. Then, the data can be standardized to a specific interval using standard deviation standardization, and the formula is

$$x^* = \frac{x - \mu}{\sigma}, \tag{25}$$

where μ is the mean of all sample data and σ is the standard deviation of all sample data.

The logarithm of strength is used to compress the scale of variables without changing the nature and correlation of the data. To facilitate calculation, the absolute value of the data is reduced.

3. Results

3.1. Box Plot Results and MICE Replace Outliers

In this paper, outliers are found after analyzing the box plot. Outliers, by definition, occupy a smaller proportion of the dataset compared with normal points. Two outliers were detected in 103 sets of sample data. In superplastic, there are two sets (i.e., approximately 2%), the sample data of the third and sixth groups, whose values are larger than the judgement range of outliers, which is obviously different from other data. The results are shown in Figure 4 and Table 3.

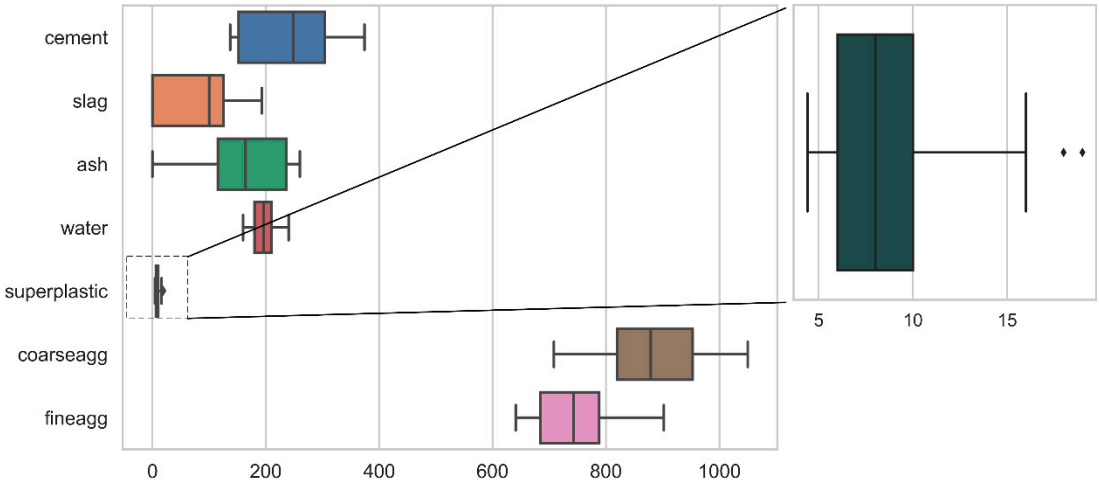

Figure 4. Box plot outlier recognition. There are two values in the feature superplastic that exceed the upper boundary and are identified as outliers.

Table 3. The box plot identifies each eigenvalue of two abnormal samples.

Sample	Cement	Slag	Ash	Water	Superplastic	Coarseagg	Fineagg	Strength
3	162.00	148.00	190.00	179.00	19.00	838.00	741.00	42.08
6	152.00	139.00	178.00	168.00	18.00	944.00	695.00	38.86

This article uses MICE to perform five interpolations and uses the optimal value to fill in the eliminated outliers. The characteristic superplastic outlier sample values 19 and 18 are replaced by 12.9 and 12.65, respectively. We compared the original data with the data distribution after using the MICE and traditional single-value filling methods to replace the median outliers, as shown in Table 4. MICE is used to replace outliers, the mean values of the dataset from 8.54 to 8.43, and the standard deviations from 2.81 to 2.50. When the

median is used to replace the outliers, the mean values of the dataset move from 8.54 to 8.35, and the standard deviations from 2.81 to 2.43. By comparison, MICE have less influence on data structure. The results show that the method of replacing outliers with MICE is effective.

Table 4. Mean replacement, MICE replacement, effect comparison.

Superplastic	Count	Mean	Std	Min	25%	50%	75%	Max
Original data	103.00	8.54	2.81	4.40	6.00	8.00	10.00	19.00
MICE	103.00	8.43	2.50	4.40	6.00	8.00	10.00	16.00
Mean	103.00	8.35	2.43	4.40	6.00	8.00	10.00	16.00

3.2. OLS Outlier Recognition Results

All sample data exceeding the judgment benchmark are summarized, as shown in Table 5. The threshold values of the four methods used to judge abnormal data points for the data in this paper are as follows: the lever value is [+0.1553, −0.1553], the student residual is [+2, −2], Cook's value is [+0.039, −0.039], and DFITS value is 0.2606. First, considering leverage and studentized residuals, it can be observed from Figure 5 that the sample 82 data has the highest leverage value of 0.19 and a studentized residual value of 1.96. With further reference to the Cook's distance and DFFITS distance, the Cook's distance and DFFITS distance of the 82nd sample were 0.11 and 0.96, respectively, both exceeding the threshold for identifying outliers. Due to the small data sample size, the 82nd sample that has a strong influence on the regression is finally eliminated.

Table 5. Samples that exceed the thresholds of Leverage, Studentized Residual, DFFITS, and Cook's: the value in bold indicates that the threshold is exceeded.

Sample Number	Leverage	Studentized Residual	DFFITS	Cook's
7	0.14	**2.81**	**1.13**	**0.15**
13	0.14	**2.80**	**1.14**	**0.15**
16	0.10	−1.97	−0.66	**0.05**
48	0.11	**3.31**	**1.18**	**0.16**
59	0.12	**2.04**	**0.76**	**0.07**
82	**0.19**	1.96	**0.96**	**0.11**
86	0.06	**−2.35**	−0.57	**0.04**
87	**0.16**	1.55	**0.67**	**0.06**

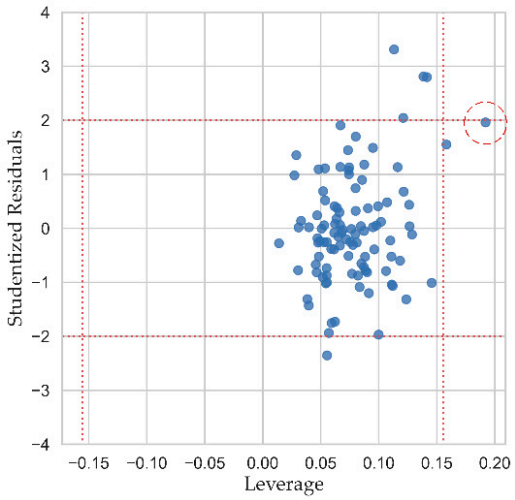

Figure 5. Student residuals and leverage values.

The validity of the identified outliers is verified by the change in the parameter estimate after deleting a certain data point. If the parameter estimation changes greatly, then the data point is considered to have a greater impact on the parameter estimation, and the data point is judged to be an abnormal value or a strong influence point; it is easy to hide between the abnormal values, and a certain statistic may be unable to accurately identify outliers, so using the OLS again to verify is recommended. As shown in Figure 6, R − squared, Akaike information criterion (AIC) [60], and Bayesian information criterion (BIC) [61] are used as reference indicators. AIC is a statistic describing the degree of fit of the regression curve to the real data points. The closer the value is to 1, the better the degree of fit. AIC is a standard to weigh the complexity of the estimation model and the excellence of fitting data. AIC is defined as $\mathrm{AIC} = 2k - 2\ln(L)$, where k is the number of parameters and L is the likelihood function. BIC means that under incomplete information, subjective probability is used to estimate some unknown states, and then a Bayesian formula is used to modify the occurrence probability. Finally, the expected value and modified probability are used to make optimal decisions, which is defined as: $\mathrm{BIC} = \ln(n)k - 2\ln(L)$, where k is the number of model parameters, n is the number of samples, and L is the likelihood function. The penalty of BIC is larger than that of AIC, and the "penalty" is imposed according to the number of independent variables. The smaller the values of AIC and BIC are, the better the model. The before and after indicators, which prove the effectiveness of outlier elimination, are shown in Table 6. R−square improved from 0.860 to 0.900, AIC decreased from 498.1 to 490.3, and BIC decreased from 519.2 to 511.3, demonstrating the effectiveness of eliminating abnormal samples. This ensures the stability and scientificity of the data for regression modeling, and improves the accuracy and stability of the regression equation.

OLS Regression Result

Dep.Variablr:	Compressive Strength	R-squared:	0.896
Model:	OLS	Adj. R-squared:	0.889
Method:	Least Squares	F-statistic:	119.2
No.Observations:	103	Prob(F-statistic):	7.16×10^{-44}
Df Residuals:	95	Log-likehood:	-241.05
Df Model:	7	AIC:	498.1
Covariance Type:	nonrpbust	BIC:	519.2

(**a**)

OLS Regression Result

Dep.Variablr:	Compressive Strength	R-squared:	0.900
Model:	OLS	Adj. R-squared:	0.893
Method:	Least Squares	F-statistic:	121.3
No.Observations:	102	Prob(F-statistic):	3.33×10^{-44}
Df Residuals:	94	Log-likehood:	-237.16
Df Model:	7	AIC:	490.3
Covariance Type:	nonrpbust	BIC:	511.3

(**b**)

Figure 6. OLS was used for regression analysis: (**a**) OLS was used for regression analysis of 103 original data. (**b**) After the 82nd abnormal sample was removed, 102 data were regression-analyzed by OLS.

Table 6. The changes of R−Square, AIC and BIC before and after the removal of outliers were compared.

	R−Square	AIC	BIC
Primary Data	0.860	498.1	519.2
Revised data	0.900	490.3	511.3

3.3. CatBoost Combination Feature Construction Results

The RMSE value is used as an evaluation indicator to optimize tree depth and iteration. The result of the operation is shown in Figure 7. Figure 7 shows that the importance of cement and fly ash is significantly higher than that of other assemblage features. The median and mean values of cement and age were taken as the new characteristics, respectively. The CatBoost algorithm was used to calculate the importance of each feature again, as shown in Figure 8. The importance of the median value of the combined feature was higher than the mean value. The weight of all the features further checked by ELI5 is shown in Table 7. For quantitative comparison of feature importance, the median weight of water and fly ash combination features is 0.0217 ± 0.0093, while the mean weight of water and fly ash combination features is 0.0153 ± 0.0109. It is observed that the median has a bigger impact on the forecast. ELI5 is a Python package for examining machine learning machines and interpreting their predictions.

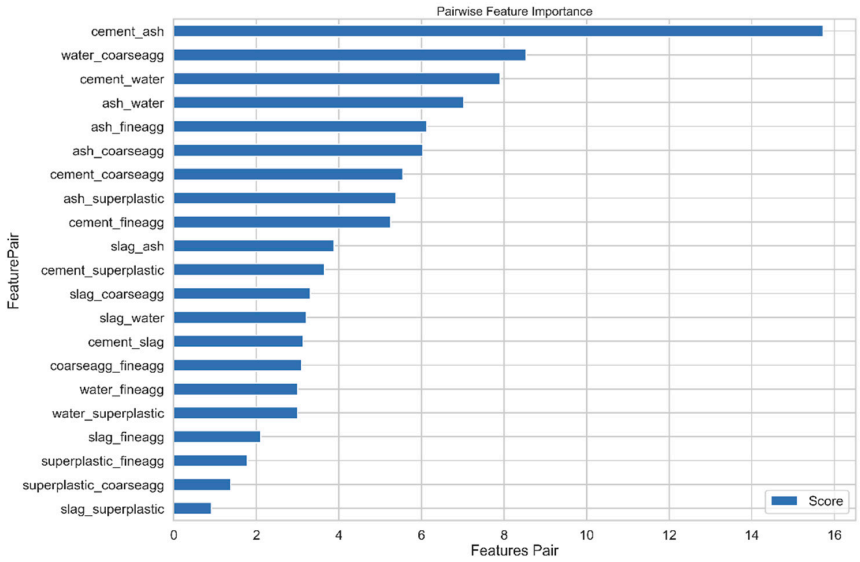

Figure 7. Importance of full combination characteristics: The importance of cement and fly ash combination is much higher than that of other characteristics, and is about twice of that of water and coarse aggregate.

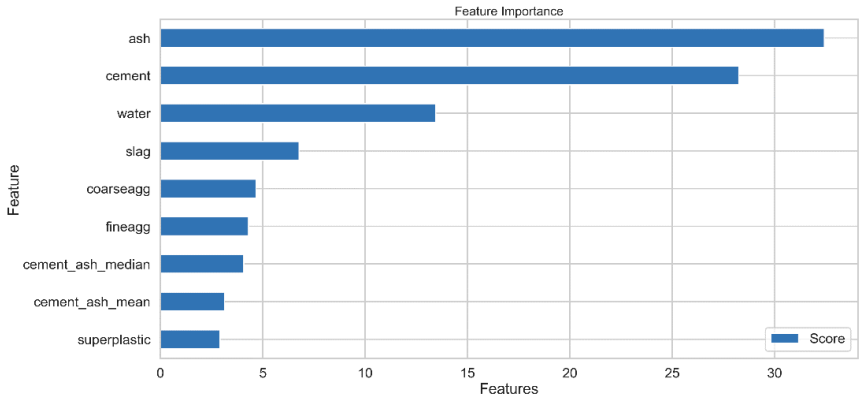

Figure 8. Feature importance of new feature set: The median of the combined characteristics of cement and fly ash was more important than its mean.

Table 7. ELI5 feature weight. The median importance of cement and fly ash combination characteristics was 0.0217 ± 0.0093, higher than the mean of 0.0153 ± 0.0109. Cement_ash_median indicates the median value of combination features, and cement_ash_mean indicates the mean value of combination features.

Weight	Feature
0.5153 ± 0.4424	cement
0.4739 ± 0.1984	ash
0.1865 ± 0.0590	water
0.0495 ± 0.0405	slag
0.0421 ± 0.0184	coarseagg
0.0228 ± 0.0180	fineagg
0.0217 ± 0.0093	cement_ash_median
0.0153 ± 0.0109	cement_ash_mean
0.0069 ± 0.0095	superplastic

Because of the high correlation between the median and mean of water and fly ash combination features, repeated feature learning is selected. We chose to delete the mean value of combination features and retain the median value of combination features. In this paper, by assessing strong combination features, we can effectively use the connection between features, which greatly enriches the feature dimension.

Finally, the CatBoost machine learning regression prediction model is selected to verify the improvement effect of the new combined feature set on the prediction accuracy, and the mean value of the verification set, and the test set is used as the evaluation criterion. By dividing the dataset into three parts, 71 samples (60%) were used for the training set, 10 samples (10%) were used for the validation set, 21 samples (20%) were used for the test set, and 8-fold cross-validation was adopted. Test results shows that the prediction effect of the model is better improved. The average R^2 for validation sets and test sets improved by about 3.33%, which indicates that the new strong features are effective in improving the accuracy of quality prediction.

3.4. Prediction Results of Single Model

This article chooses to include the following models: linear regression (LR), which is very important in solving regression problems; random forest (RF) and extremely randomized trees (ExtraTrees), which use the bagging ensemble learning method; gradient-boosted decision tree (GBDT); extreme gradient boosting (eXtreme gradient boosting; XGBoost); and categorical boosting (CatBoost), which use gradients with the boosting ensemble learning method to improve decision trees. The ensemble learning of bagging and boosting has achieved good results in other fields. SVM (support vector machine) can obtain much better results than other algorithms can on small-sample training sets and has unique advantages in the performance of high-dimensional regression problems. KNN also has good practical application effects because of its mature theory and efficient training. All selected models are supported by rigorous mathematical theory. All models were implemented and optimized in Python 3.7. The operating system is a 64-bit Windows 7 with a quad-core Intel Core I7 CPU @ 2.6 GHz (8 cpus) and 8.00 GB of RAM.

First, it is necessary to optimize the parameters of each monomer base model to achieve the optimal prediction effect. In this paper, random and grid searches are used to optimize the parameters to improve the training speed and accuracy. The essence of the grid search method is to divide all the parameters to be searched into a grid with the same length according to the established space search scope and the proposed coordinate system. Next, a random search is used to select the best point for rough search with stride length in a large range. Then, a smaller step size is used to search the grid near the best advantage to select the optimal solution and repeat the above steps. The model's generalization ability is further improved through eight-fold cross validation to avoid overfitting.

The optimal parameters of each model are shown in Table 8. Statistical indicators are calculated on the test set to evaluate the predictive performance of these eight models. The prediction and evaluation results are shown in Table 9. For high-dimensionality and small-sample regression quality prediction, SVR and linear regression perform the best among the eight models. According to each statistical indicator, the RMSE of the SVR prediction result is 0.0788, which is the smallest among all models. The MAE of the linear regression prediction result is 0.0619, which is the smallest among all models. In the test set, the SVR and linear regression model R^2 is the highest, reaching 0.8883. KNN performed the worst, with R^2 of 0.6593. Except for RandomForest and KNN, all other models have R^2 above 0.8.

The error of each model is shown in Figure 9. The error between the predicted value of each single model and the true value is different in the prediction effect of each sample point in the test set. It can be seen that different machine learning models have different observation angles on the data space. The machine learning model with high prediction performance cannot achieve better prediction results at every data point, and the model with weak performance can also have small errors at some data points. Therefore, to

achieve the optimal performance of the overlay ensemble learning model, it is necessary to analyze the learning ability of each basic learner and the relevance of each learner, build a stacking ensemble learning model, and use the advantages of each single model.

Table 8. Optimal parameters for each model.

Model	Parameters	Value
Linear Regression		
KNN	The number of neighboring points	3
RandomForest	Number of decision trees	100
	Maximum characteristic number	"sqrt"
	Maximum depth of decision tree	3
GradientBoost	Learning rate	0.1
	Number of decision trees	100
	Maximum characteristic number	"sqrt"
	Maximum depth of decision tree	3
ExtraTrees	Number of decision trees	800
	Maximum characteristic number	"auto"
	Maximum depth of decision tree	9
XGboost	Number of decision trees	600
	Learning rate	0.1
	Maximum depth of decision tree	4
SVR	Kernel	RBF
	C	2
	gamma	0.01
CatBoost	Learning rate	0.01
	L2 regular parameter	2
	Tree deep	3
	Loss function	RMSE

Table 9. Forecast evaluation of statistical indicators.

Model	R^2	RMSE	MAE
Linear Regression	0.8883	0.0799	0.0619
KNN	0.6593	0.1392	0.1162
RandomForest	0.7996	0.1068	0.0855
GradientBoost	0.8601	0.0890	0.0695
ExtraTrees	0.8616	0.0886	0.0730
XGboost	0.8091	0.1043	0.0755
SVR	0.8883	0.0788	0.0639
CatBoost	0.8083	0.1025	0.0819

3.5. Calculation of Correlation Degree of Each Model

In the first level of stacking models, a more diverse model is selected as the primary learner. This is because different algorithms observe data in different data spaces and data structures and build corresponding models according to their respective algorithm rules. It effectively combines the advantages of various model algorithms, reduces the risk of low-accuracy performance of a single model, and is suitable for high-dimensionality and small-sample data.

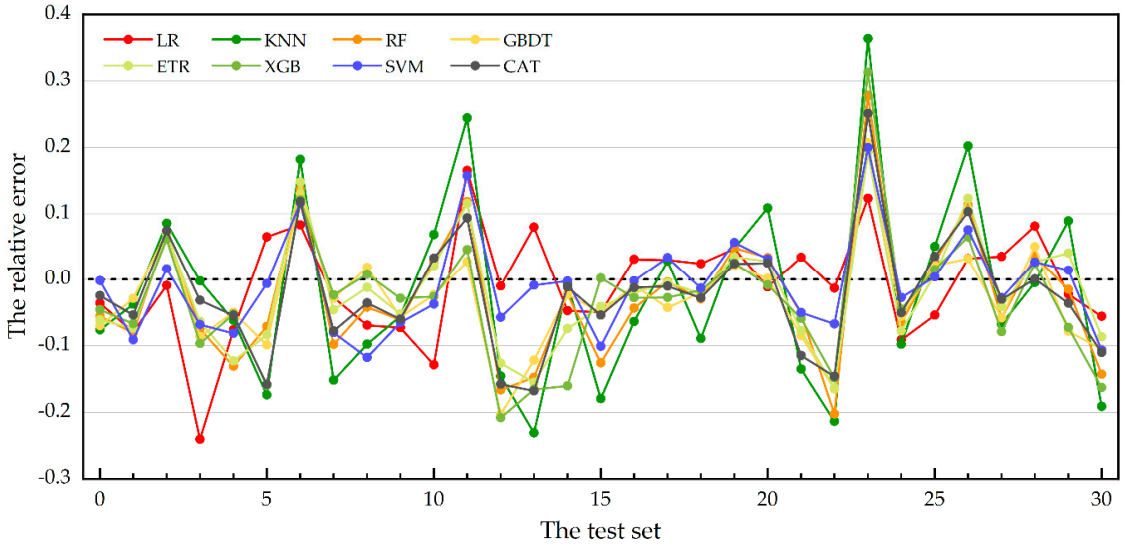

Figure 9. Relative error of prediction of each single model: The error between the predicted value and the actual value of each model.

Therefore, based on ensuring the accuracy of prediction, the selection of algorithms with large differences can maximize the advantages of different algorithms so that each differentiated model can learn from each other. Diversity and difference make the ensemble learning results more robust and accurate and improve the prediction effect. Considering the complexity of models, the number of basic learning models should not be too large.

Metrics such as Pearson's correlation coefficient [62], Spearman's correlation coefficient [63], and Kendall's correlation coefficient [63] in the existing methods are widely used to measure the linear relationship between features or simple monotonous nonlinear data, but they struggle to well represent the existing nonlinear relationship; the computational complexity of the kernel density estimation (KDE) [64] algorithm and KNN metrics is too high. In this paper, the maximal information coefficient (MIC) [34] is selected to measure the linear or nonlinear correlation strength between the selected models, and the prediction errors of each model are used for calculation. Compared with the correlation coefficient, MIC can not only measure the linear and nonlinear relationship between variables in a large amount of data, but also can widely dig out the nonfunctional dependence relationship between the response variables. In addition, its computational complexity is low, and its robustness is strong. The larger the MIC value, the higher the importance of the feature to the response variable. The definition of mutual information is

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \tag{26}$$

where $X = \{x_i, i = 1, 2, \ldots, n\}$ and $Y = \{x_i, i = 1, 2, \ldots, n\}$ are two random variables, and n is the sample size. $p(x)$ and $p(y)$ are the marginal probability distribution densities of X and Y.

Maximum mutual information is defined as

$$mic(x,y) = \max_{|X||Y|<B} \frac{\max(I(X,Y))}{\log_2(\min(|X|,|Y|))}, \tag{27}$$

where $\max(I(X,Y))$ represents the maximum mutual information value; B is the upper limit of grid partitioning and is a growth function related to the number of data samples n. $B = n^{0.6}$ has the best effect, and this paper takes the optimal value.

To optimize the performance of the stacking model in this paper, after completing the hyperparameter optimization of each primary learner to improve learning ability, it is

necessary to further consider the correlation between the models and select the first-layer base learner with large differences. On the premise of ensuring the accuracy of prediction, it is necessary to add different kinds of prediction algorithms as much as possible. To select the best base model for combination, the prediction error distribution of each optimal primary learner is separately predicted, and the MIC is used as the correlation index. The error correlation analysis of each base model is shown in Figure 10.

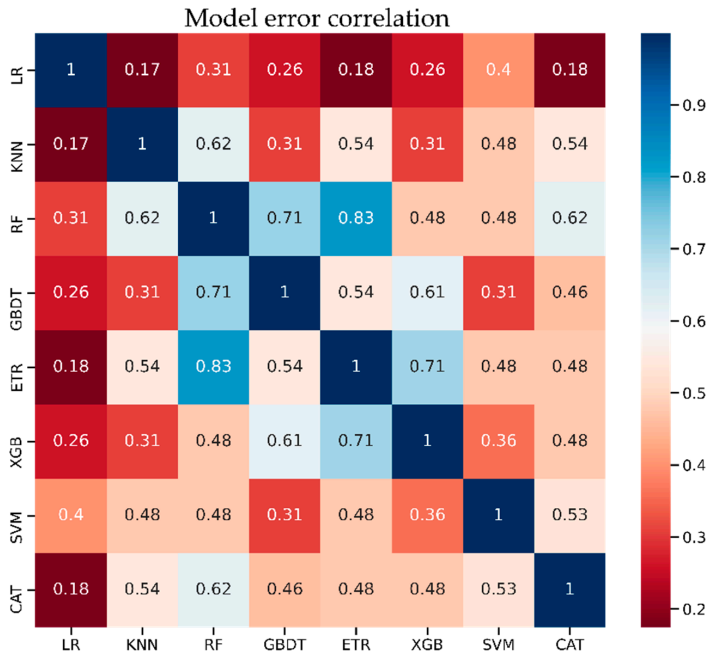

Figure 10. MIC model correlation analysis.

Among them, random forest and ExtraTrees have the highest correlation, with a correlation coefficient of 0.83, because both are ensemble methods of bagging, but the tree splitting method has changed; the correlation between XGBoost and GBDT is also as high as 0.61, because both are the integration methods of boosting. Random forest, ExtraTrees, XGBoost, GBDT, and CatBoost were highly correlated; this is due to the fact that both bagging and boosting are based on decision trees, although their principles are slightly different. The differences in the way the algorithm observes the data are not very large. Because of its low prediction accuracy and large inherent error, KNN has a high error correlation with other models. The linear regression has a large gap with other models in training mechanism, so the error correlation is low, and the correlation with SVR is relatively high, due to their high prediction accuracy and the inherent errors that cause certain correlation.

3.6. Stacking Model Combination Performance Analysis

To verify the rationality of the abovementioned single-model combination method, and due to the small sample, the input of the metamodel is obtained through eight cross-validations to prevent overfitting. There is a significant gap in the prediction effect of the ensemble model obtained by combining different single models. To further verify the influence of base learner selection in the stacking ensemble learning model on the prediction results, Table 10 and Figure 11 show the prediction results of different combination methods on the test set.

The combination method Stacking_1 selects the model combination method with the greatest correlation, and the result of the ensemble model is worse than the prediction effect of the single model linear regression and SVM. The stacking model has an R^2 of 0.8796, and MAE and RMSE of 0.0607 and 0.0784, respectively. This is because the first-layer model contains the most relevant random forest, ExtraTrees, GBDT, and XGBoost, which causes the data to be repeatedly trained, resulting in overfitting, which reduces the accuracy of the model.

Table 10. Prediction results of different stacking models.

Stacking Model	Base Model Combination	The Evaluation Index		
		R2	MAE	RMSE
Stacking 1	GBDT, Random Forest, XGBoost, ExtraTrees, GBDT	0.8796	0.0607	0.0784
Stacking 2	SVR, XGBoost, Random Forest, GBDT, CatBoost	0.9415	0.0441	0.0546
Stacking 3	SVR, XGBoost, ExtraTrees, Random Forest CatBoost	0.9445	0.0435	0.0532
Stacking 4	Linear Regression, KNN, SVR, CatBoost	0.9702	0.0304	0.0389

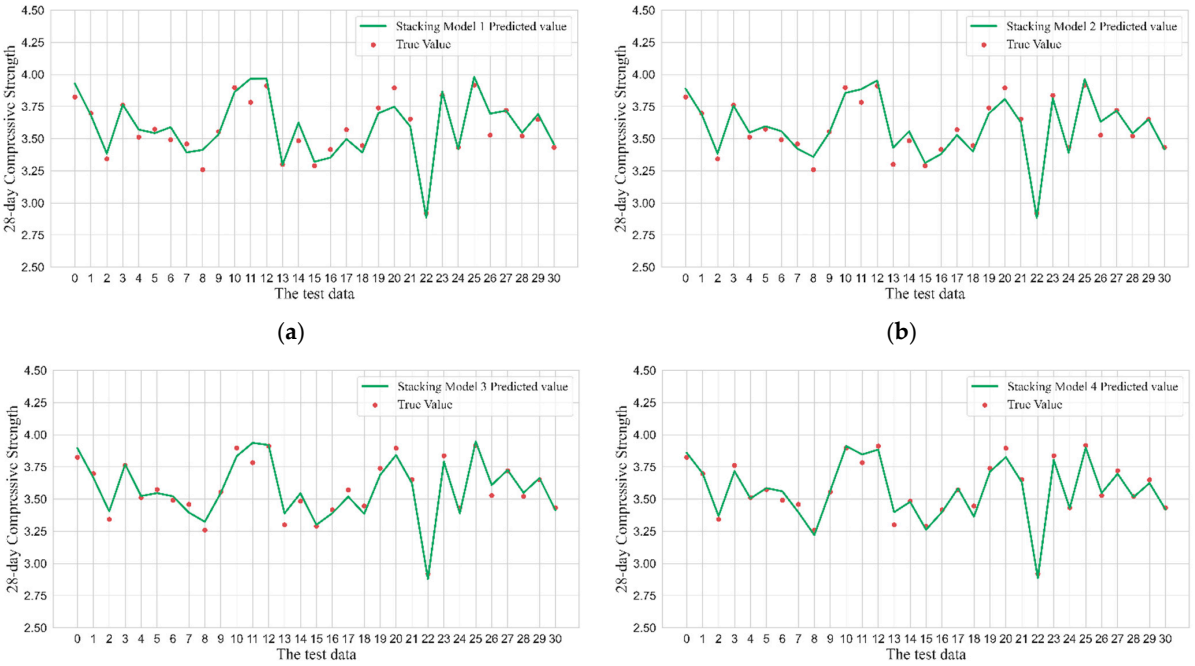

Figure 11. Comparison of prediction errors of different stacking models: (**a**) Stacking model 1 builds the most relevant ensemble learning model; (**b**) Stacking model 2 builds the ensemble learning model randomly; (**c**) Stacking model 3 builds the ensemble learning model randomly; (**d**) Stacking model 4 builds a minimum correlation ensemble learning model. The ordinate of 28-day compressive strength is logarithmically processed.

The combination methods Stacking_2 and Stacking_3 choose random combination models, which are both higher than the optimal single model prediction accuracy. The R^2 of Stacking_2 model is 0.9415, and MAE and RMSE are 0.0441 and 0.0546, respectively. The R^2 of Stacking_3 model is 0.9445, and MAE and RMSE are 0.0435 and 0.0532, respectively.

The combination method Stacking_4 chooses the combination method that meets the minimum correlation on the basis of having stronger models (linear regression, SVR, KNN, CatBoost) to make the best use of the advantages of different algorithms to observe data from different data spaces to ensure prediction accuracy, effectively improving the generalization performance of the fusion ensemble model and showing the best predictive effect. The R^2 of Stacking_4 model is 0.9702, and MAE and RMSE are 0.0304 and 0.0389, respectively.

The experimental results show that it is necessary to analyze the correlation of each model before constructing the stacking ensemble learning model. It is difficult to achieve the best prediction effect by randomly selecting a model to build a stacking model. MIC can

effectively analyze the linear and nonlinear correlation of the model, which provides strong support for the model selection when constructing the stacking ensemble model. Compared with the stacking 2 ensemble model and stacking 3 ensemble model constructed by random selection, the most differentiated stacking model has an average increase of 2.72% in R^2, an average reduction of 1.34% in MAE, and an average reduction in RMSE of 1.5%.

In addition to choosing models with low relevance, the final prediction effect is higher than that of the single model. The reason is that the stacking model makes full use of the advantages of different types of algorithms, makes up for the poor prediction effect of each algorithm, and further amends it through the second layer of learner. On the other hand, a single model is more sensitive to data types. During the training process, the model often has the risk of falling into a local minimum. The corresponding model generalization performance may be poor, and the stacking model combines multiple base learners to effectively reduce the risk of falling into a local minimum. The test results prove that the stacking ensemble model has improved the accuracy of high-dimensionality and small-sample data quality prediction.

3.7. Blending Ensemble Learning Method Prediction Results and Comparison of Multiple Models

According to the evaluation index parameters, the specific results of each algorithm are shown in Figure 12. It can be determined that the R^2 of the stacking ensemble learning model is 0.9702. Compared with the blending ensemble learning model, linear regression, and SVR, the R^2 is increased by 0.0333, 0.0819, and 0.0819, respectively, and the average increase is about 0.0657; the RMSE of stacking is 0.0389, which is reduced by 0.0178, 0.0410, and 0.0399, with an average decrease of 0.0329; stacking's MAE is 0.0304, with an average decrease of 0.0137, 0.0315, and 0.0335, with an average decrease of 0.0262. That is, the fitting effect of the stacking integration algorithm is obviously better than that of other single models, and it has higher accuracy. The two ensemble learning models are higher than the optimal single model. The R^2, RMSE, and MAE of the stacking ensemble learning model are better than the hybrid algorithm, which proves the feasibility of the established model. The stacking ensemble learning model is more suitable for high-dimensional, small-sample quality regression prediction.

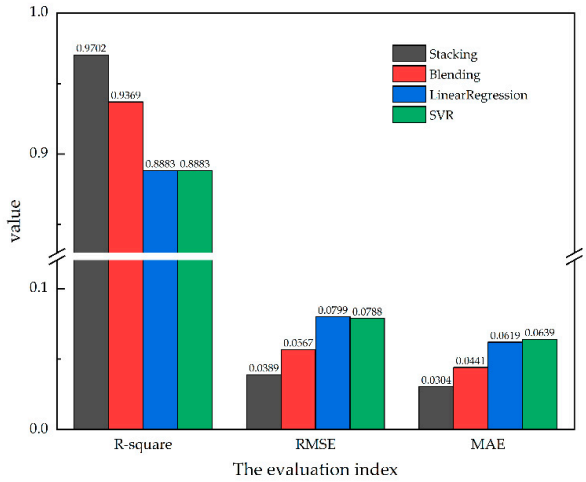

Figure 12. Comparison of prediction performance of stacking ensemble learning model, blending ensemble learning model, linear regression model, and SVR model.

4. Discussion

This article combines the characteristics of high-dimensional, small-sample regression prediction. UCI's real dataset is used to verify the feasibility and effectiveness of the algorithm. The box plot method and the MICE algorithm are used to detect and replace the outliers in the features in the database, respectively. In addition, OLS is used to further detect abnormal sample values with the help of studentized residual, leverage,

Cook's distance, and DFFITS distance to avoid the impact of strong influential points on the prediction accuracy. To further increase the accuracy of regression prediction, using the function of CatBoost feature combination, adding a strong combination feature as a new feature, and verifying that the new feature set is compared with the original feature, CatBoost shows an average R^2 increase of 3.333% on the verification set and the test set. Box–Cox transformation was performed on the features with skewness greater than 0.5 to make them conform to the approximate normal distribution and standardize the data. Then, by dividing the database into a training set (70%), validation set (10%), and test set (20%), R^2, RMSE, and MAE were used as model evaluation indicators. Through the data preprocessing, strong combination feature construction, and grid search methods in this paper, a single model has achieved good prediction results. Aiming at the problem that the single-model prediction accuracy is still insufficient in the actual production process, a regression prediction algorithm based on stacking model fusion is proposed. Different regression models were combined through the stacking method, and eight-fold cross-validation was used to prevent overfitting. Performing model correlation analysis before model establishment can effectively measure the differences between models. Maximizing model differences can effectively combine the advantages of single-model prediction algorithms. When one observes the data space and structure from different angles, different algorithms complement each other to obtain the best prediction model.

Table 11 longitudinally compares the R^2, RMSE, and MAE of the minimum correlation stacking ensemble learning model, the blending ensemble learning model, and the single model optimized by hyperparameters. Experiments show that the minimal correlation stacking model can achieve the best prediction effect, and R^2, RMSE, and MAE are 0.9702, 0.0389, and 0.0304, respectively. The basic model of the stacking algorithm is used to synthesize the prediction results of the multidimensional prediction model, which overcomes the shortcomings of single model generalization and poor applicability. In a single model, the machine learning algorithms linear regression, GradientBoost, ExtraTrees, SVR, XGboost, CatBoost, RandomForest and KNN obtained R^2 values of 0.8883, 0.8601, 0.8616, 0.8883,.0.8091, 0.8083, 0.7996, and 0.6593, respectively. The RMSE values were 0.0799, 0.0890, 0.08860.0788, 0.1043, 0.1025, 0.1068, and 0.1392, respectively. The MAE values were 0.0619, 0.0695, 0.0730, 0.0639, 0.0755, 0.0819, 0.0855, and 0.1162, respectively. The minimal correlation stacking model increased the R^2 value by 8.19%, 11.01%, 10.86%, 8.19%, 16.11%, 16.19%, 17.06%, and 31.09%, and decreased the RMSE value by 4.1%, 5.01%, 4.97%, 3.99%, 6.54%, 6.36%, 6.79%, and 10.03%, respectively. The MAE values decreased by 3.15%, 3.91%, 4.26%, 3.35%, 4.51%, 5.15%, 5.51%, and 8.58%, respectively, showing satisfactory predictive performance. Compared with the blending ensemble learning method of optimal model combination, the proposed stacking model also achieved better results. Our study demonstrates that the stacking model has excellent accuracy and high application value in high-dimensional, small-sample regression prediction and shows unique advantages.

Table 11. Comparison of single model, blending ensemble model, and stacking ensemble model.

Model	R^2	RMSE	MAE
Stacking	0.9702	0.0389	0.0304
Blending	0.9369	0.0567	0.0441
SVR	0.8883	0.0788	0.0639
Linear Regression	0.8883	0.0799	0.0619
ExtraTrees	0.8616	0.0886	0.0730
GradientBoost	0.8601	0.0890	0.0695
XGboost	0.8091	0.1043	0.0755
CatBoost	0.8083	0.1025	0.0819
RandomForest	0.7996	0.1068	0.0855
KNN	0.6593	0.1392	0.1162

5. Conclusions

In the selection of the base model of the stacking ensemble model, in order to achieve higher prediction performance, on the one hand, it is necessary to integrate low-relevance and diversified machine learning algorithms. On the other hand, the predictive ability of a single model will also have an impact. In future research, the following issues will be further discussed in depth. Considering that the stacking design is more complicated, the parameter selection of a single model requires multiple trainings, which increases the training time, and small-sample data also takes a long time. Therefore, modeling optimization is necessary in future research. The process is further divided into parallel, reducing the complexity of the algorithm, ensuring the accuracy of prediction, and reducing the calculation time.

The stacking ensemble model is used to predict the manufacturing quality. The experimental results show that the stacking ensemble model, with strong forecasting ability and low-correlation base learner combination, has good accuracy and high application value in high-dimensional, small-sample quality forecasting. Therefore, the proposed stacking ensemble model has high forecasting accuracy and stability, and several application values, as follows: (1) it provides quality regression prediction guidance for the high-dimensional, small-sample production of small samples with high-dimensionality, and (2) it is beneficial for enterprises to find quality problems of production products in time, to avoid losses and improve competitiveness in the market.

Author Contributions: Conceptualization, J.Y. and Y.Z.; methodology, J.Y.; software, J.Y.; validation, J.Y., R.P. and Y.Z.; formal analysis, J.Y.; resources, Y.Z.; data curation, J.Y.; writing—original draft preparation, J.Y.; writing—review and editing, J.Y.; visualization, J.Y.; supervision, R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Industry & Information Technology, China (TC200H01X-5), by the Science & Technology Department of Xinjiang Production and Construction Corps, China (S2020AA784).

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Carvalho, T.P.; Soares, F.A.A.M.N.; Vita, R.; Francisco, R.D.P.; Basto, J.; Alcalá, S.G.S. A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.* **2019**, *137*, 106024. [CrossRef]
2. Li, Q.; Wei, F.; Zhou, S. Early warning systems for multi-variety and small batch manufacturing based on active learning. *J. Intell. Fuzzy Syst.* **2017**, *33*, 2945–2952. [CrossRef]
3. Aparisi, F.; Luna, M.A.D. The Design and Performance of the Multivariate Synthetic-T Control Chart. *Commun. Stat.* **2009**, *38*, 173–192. [CrossRef]
4. Kourti, T.; Lee, J.; Macgregor, J.F. Experiences with industrial applications of projection methods for multivariate statistical process control. *Comput. Chem. Eng.* **1996**, *20*, S745–S750. [CrossRef]
5. Park, Y.-M.; Moon, U.-C.; Lee, K.Y. A self-organizing power system stabilizer using fuzzy auto-regressive moving average (FARMA) model. *IEEE Trans. Energy Convers.* **1996**, *11*, 442–448. [CrossRef]
6. Lowry, C.A.; Woodall, W.H.; Champ, C.W.; Rigdon, S.E. A multivariate exponentially weighted moving average. *Technometrics* **1992**, *34*, 46. [CrossRef]
7. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
8. Ge, Z.; Song, Z.; Ding, S.X.; Huang, B. Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access* **2017**, *5*, 20590–20616. [CrossRef]
9. Jiao, A.; Zhang, G.; Liu, B.; Liu, W. Prediction of Manufacturing Quality of Holes Based on a BP Neural Network. *Appl. Sci.* **2020**, *10*, 2108. [CrossRef]

10. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]
11. Yang, X.S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [CrossRef]
12. Cem, B.; Tahsin, K. Proper estimation of surface roughness using hybrid intelligence based on artificial neural network and genetic algorithm. *J. Manuf. Processes* **2021**, *70*, 560–569.
13. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A. IoT type-of-traffic forecasting method based on gradient boosting neural networks. *Future Gener. Comput. Syst.* **2020**, *105*, 331–345. [CrossRef]
14. Li, Z.; Chen, X.; Wu, L.; Ahmed, A.-S.; Wang, T.; Zhang, Y.; Li, H.; Li, Z.; Xu, Y.; Tong, Y. Error Analysis of Air-Core Coil Current Transformer Based on Stacking Model Fusion. *Energies* **2021**, *14*, 1912. [CrossRef]
15. Shi, J.; Zhang, J. Load Forecasting Based on Multi-model by Stacking Ensemble Learning. *Proc. CSEE* **2019**, *39*, 4032–4041.
16. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **2020**, *14*, 241–258. [CrossRef]
17. Aaa, B.; Hd, B. A bagging algorithm for the imputation of missing values in time series. *Expert Syst. Appl.* **2019**, *129*, 10–26.
18. Wang, B.; Pineau, J. Online Bagging and Boosting for Imbalanced Data Streams. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 3353–3366. [CrossRef]
19. Xu, L.; Li, Z. A New Appraisal Model of Second-Hand Housing Prices in China's First-Tier Cities Based on Machine Learning Algorithms. *Comput. Econ.* **2021**, *57*, 617–637. [CrossRef]
20. Yin, X.; Liu, Q.; Pan, Y.; Huang, X.; Wu, J.; Wang, X. Strength of Stacking Technique of Ensemble Learning in Rockburst Prediction with Imbalanced Data: Comparison of Eight Single and Ensemble Models. *Nat. Resour. Res.* **2021**, *30*, 1795–1815. [CrossRef]
21. Dong, Y.; Zhang, H.; Wang, C.; Zhou, X. Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm. *Neurocomputing* **2021**, *462*, 169–184. [CrossRef]
22. Durrant, R.; Kabán, A. Random projections as regularizers: Learning a linear discriminant from fewer observations than dimensions. *Mach. Learn.* **2015**, *99*, 257–286. [CrossRef]
23. Lopez-Martin, M.; Nevado, A.; Carro, B. Detection of early stages of Alzheimer's disease based on MEG activity with a ran-domized convolutional neural network. *Artif. Intell. Med.* **2020**, *107*, 101924. [CrossRef] [PubMed]
24. Hawkins, D.M. *Identification of Outliers*; Chapman and Hall: London, UK, 1980.
25. Cade, B.S.; Noon, B.R. A gentle introduction to quantile regression for ecologists. *Front. Ecol. Environ.* **2003**, *1*, 412–420. [CrossRef]
26. Hang, J.K.; Reiter, J.P.; Wang, Q.; Cox, L.H.; Karr, A.F. Multiple Imputation of Missing or Faulty Values Under Linear Constraints. *J. Bus. Econ. Stat.* **2014**, *32*, 375–386.
27. Zhao, Y.; Qi, L. Multiple imputation in the presence of high-dimensional data. *Stat. Methods Med. Res.* **2013**, *25*, 2021–2035. [CrossRef]
28. Zhang, Z. Residuals and regression diagnostics: Focusing on logistic regression. *Ann. Transl. Med.* **2016**, *4*, 195. [CrossRef]
29. Nurunnabi, A.; Nasser, M.; Imon, A. Identification and classification of multiple outliers, high leverage points and influential observations in linear regression. *J. Appl. Stat.* **2016**, *43*, 509–525. [CrossRef]
30. Cook, R.D. Detection of Influential Observation in Linear Regression. *Technometrics* **1977**, *19*, 15–18.
31. Belsley, D.A.; Kuh, E.; Welsch, R.E. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*; Belsley, D.A., Kuh, E., Welsch, R.E., Eds.; John Wiley & Sons: Hoboken, NJ, USA, 2005.
32. Zuehlke, T.W. Estimation of a type 2 Tobit model with generalized Box-Cox transformation. *Appl. Econ.* **2021**, *53*, 1952–1975. [CrossRef]
33. Yonghui, X.; Ruotong, M.; Xi, Z. Research on a Gas Concentration Prediction Algorithm Based on Stacking. *Sensors* **2021**, *21*, 1597.
34. Reshef, D.N.; Reshef, Y.A.; Finucane, H.K.; Grossman, S.R.; McVean, G.; Turnbaugh, P.J.; Lander, E.S.; Mitzenmacher, M.; Sabeti, P.C. Detecting Novel Associations in Large Data Sets. *Science* **2011**, *334*, 1518–1524. [CrossRef]
35. Sánchez-Illana, Á.; Perez-Guaita, D.; Cuesta-García, D.; Sanjuan-Herráez, J.D.; Vento, M.; Ruiz-Cerdá, J.L.; Quintás, G.; Kuligowski, J. Model selection for within-batch effect correction in UPLC-MS metabolomics using quality control—Support vector regression. *Anal. Chim. Acta* **2018**, *1026*, 62–68. [CrossRef] [PubMed]
36. Knorr, E.M.; Ng, R.T.; Tucakov, V. Distance-based outliers: Algorithms and applications. *VLDB J.* **2000**, *8*, 237–253. [CrossRef]
37. Royston, P.; White, I. Multiple Imputation by Chained Equations (MICE): Implementation in Stata. *J. Stat. Softw.* **2011**, *45*. [CrossRef]
38. Buuren, S.V.; Oudshoorn, K. *Flexible Mutlivariate Imputation by MICE*; TNO: Leiden, The Netherlands, 1999.
39. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *arXiv* **2017**, arXiv:1706.09516.
40. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient boosting with categorical features support. *arXiv* **2018**, arXiv:1810.11363.
41. Huang, G.; Wu, L.; Ma, X.; Zhang, W.; Fan, J.; Yu, X.; Zeng, W.; Zhou, H. Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions. *J. Hydrol.* **2019**, *574*, 1029–1041. [CrossRef]
42. Sales, J. The use of linear regression to predict digestible protein and available amino acid contents of feed ingredients and diets for fish. *Aquaculture* **2008**, *278*, 128–142. [CrossRef]
43. Cherkassky, V.; Ma, Y. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Netw.* **2004**, *17*, 113–126. [CrossRef]
44. Pan, Y.; Chen, S.; Qiao, F.; Ukkusuri, S.V.; Tang, K. Estimation of real-driving emissions for buses fueled with liquefied natural gas based on gradient boosted regression trees. *Sci. Total Environ.* **2019**, *660*, 741–750. [CrossRef] [PubMed]

45. Chen, T.; Tong, H.; Benesty, M. Xgboost: Extreme Gradient Boosting. 2016. Available online: https://github.com/dmlc/xgboost (accessed on 18 December 2021).
46. Boobier, S.; Hose, D.R.J.; Blacker, A.J.; Nguyen, B.N. Machine learning with physicochemical relationships: Solubility prediction in organic solvents and water. *Nat. Commun.* **2020**, *11*, 5753. [CrossRef] [PubMed]
47. Breiman, L. Random forest. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
48. Keller, J.M.; Gray, M.R.; Givens, J.A. A fuzzy K-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 580–585. [CrossRef]
49. Bentéjac, C.; Csrg, A.; Martínez-Muoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2020**, *54*, 1937–1967. [CrossRef]
50. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference, San Francisco, CA, USA, 13–17 August 2016.
51. Samat, A.; Persello, C.; Liu, S.; Li, E.; Miao, Z.; Abuduwaili, J. Classification of VHR Multispectral Images Using Extratrees and Maximally Stable Extremal Region-Guided Morphological Profile. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3179–3195. [CrossRef]
52. Guo, X.; Gao, Y.; Zheng, D.; Ning, Y.; Zhao, Q. Study on short-term photovoltaic power prediction model based on the Stacking ensemble learning. *Energy Rep.* **2020**, *6*, 1424–1431. [CrossRef]
53. Lee, T.R.; Wood, W.T.; Phrampus, B.J. A Machine Learning (kNN) Approach to Predicting Global Seafloor Total Organic Carbon. *Glob. Biogeochem. Cycles* **2019**, *33*, 37–46. [CrossRef]
54. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. *J. Big Data* **2020**, *7*, 94. [CrossRef]
55. Twab, C.; Wei, Z.A.; Xjab, C.; Wga, C.; Yah, A. Evaluation of stacking and blending ensemble learning methods for estimating daily reference evapotranspiration. *Comput. Electron. Agric.* **2021**, *184*, 106039.
56. Huaichun, F.; Shouwei, G.; Yan, P.; Nan, Z. Prediction of fishing vessel operation mode based on Stacking model fusion. *J. Phys. Conf. Ser.* **2021**, *1792*, 012030.
57. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. [CrossRef]
58. Breiman, L. Stacked regressions. *Mach. Learn.* **1996**, *24*, 49–64. [CrossRef]
59. Ling, H.; Qian, C.X.; Kang, W.C.; Liang, C.Y.; Chen, H.C. Machine and K-Fold cross validation to predict compressive strength of concrete in marine environment. *Constr. Build. Mater.* **2019**, *206*, 355–363. [CrossRef]
60. Akaike, H. *Information Theory and an Extension of the Maximum Likelihood Principle*; Springer: New York, NY, USA, 1998.
61. David, P.; Buckley, T.R. Model Selection and Model Averaging in Phylogenetics: Advantages of Akaike Information Criterion and Bayesian Approaches over Likelihood Ratio Tests. *Syst. Biol.* **2004**, *53*, 793–808.
62. Saqlain, S.M.; Sher, M.; Shah, F.A.; Khan, I.; Ashraf, M.U.; Awais, M.; Ghani, A. Fisher score and Matthews correlation coefficient-based feature subset selection for heart disease diagnosis using support vector machines. *Knowl. Inf. Syst.* **2019**, *58*, 139–167. [CrossRef]
63. Puth, M.-T.; Neuhäuser, M.; Ruxton, G.D. Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits. *Anim. Behav.* **2015**, *102*, 77–84. [CrossRef]
64. Pérez, A.; Larrañaga, P.; Inza, I. Bayesian classifiers based on kernel density estimation: Flexible classifiers. *Int. J. Approx. Reason.* **2009**, *50*, 341–362. [CrossRef]