Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.020220

ARTICLE

Metal Corrosion Rate Prediction of Small Samples Using an Ensemble Technique

Yang Yang[1,2,*], Pengfei Zheng[3,4], Fanru Zeng[5], Peng Xin[6], Guoxi He[1] and Kexi Liao[1]

[1]State Key Laboratory of Oil Gas Reservoir Geology and Exploitation, Southwest Petroleum University, Chengdu, 610500, China

[2]School of Earth Sciences and Technology, Southwest Petroleum University, Chengdu, 610500, China

[3]Spatial Information Technology and Big Data Mining Research Center, School of Earth Sciences and Technology, Southwest Petroleum University, Chengdu, 610500, China

[4]Sichuan Xinyang Anchuang Technology Co., Ltd., Chengdu, 610500, China

[5]Sichuan Water Conservancy College, Chengdu, 610500, China

[6]CCDC Safety, Environment, Quality Supervision & Testing Research Institute, Guanghan, 618300, China

*Corresponding Author: Yang Yang. Email: swpu_yangy@126.com

Received: 11 November 2021 Accepted: 24 February 2022

ABSTRACT

Accurate prediction of the internal corrosion rates of oil and gas pipelines could be an effective way to prevent pipeline leaks. In this study, a proposed framework for predicting corrosion rates under a small sample of metal corrosion data in the laboratory was developed to provide a new perspective on how to solve the problem of pipeline corrosion under the condition of insufficient real samples. This approach employed the bagging algorithm to construct a strong learner by integrating several KNN learners. A total of 99 data were collected and split into training and test set with a 9:1 ratio. The training set was used to obtain the best hyperparameters by 10-fold cross-validation and grid search, and the test set was used to determine the performance of the model. The results showed that the Mean Absolute Error (MAE) of this framework is 28.06% of the traditional model and outperforms other ensemble methods. Therefore, the proposed framework is suitable for metal corrosion prediction under small sample conditions.

KEYWORDS

Oil pipeline; bagging; KNN; ensemble learning; small sample size

1 Introduction

Owing to its economy and safety, pipeline transportation is one of the most important modes of oil and gas transportation [1]. However, the longer the pipeline is running, the greater the risk of corrosion [2]. Corrosion is the main incentive to increase the risk of oil and gas pipelines [3]. Due to the high cost of measuring equipment and the requirements of specific and regular calibration operations, it is not easy to obtain the status of corrosion defect measurement [4]. Thus, it is challenging to establish a high-precision corrosion prediction model. Despite this, many scholars have studied this problem and

established different corrosion rate prediction models. Traditional (semi-empirical) models include the de Waard [5,6], Cassandra (BP) [7,8], and Norsok [9] models. Of these, the de Waard model has been the most widely used since its establishment, although it rarely considers the influence of protective corrosion, especially at high temperature and pH values. The Cassandra model, on the other hand, takes into account the influence of corrosion inhibitors and can achieve better prediction performance at high temperatures, but it does not consider the influence of medium flow rates and Cl^-. The Norsok model is a purely empirical model established using a large amount of data, and takes into account the influence of a greater number of factors compared to the other models; however, as it does not consider the underlying mechanisms, it lacks universality, and it is relatively conservative in its predictions.

With the development of artificial intelligence (AI) technology, the use of deep learning methods to obtain better corrosion predictions for oil and gas pipelines has also become a focus of current research [10–12]. For example, Jain et al. [13] proposed a quantitative evaluation model for the external corrosion rate of oil and gas pipelines based on Bayesian networks. Abbas et al. [14] developed a neural network (NN) model to predict CO_2 corrosion in pipelines at high partial pressures. Ossai [15] developed a feedforward NN based on the particle swarm algorithm (PSO). Chen et al. [16] proposed a fuzzy NN model of Principal Component Analysis Based Dynamic Fuzzy Neural Network (PCA-D-FNN). Seghier et al. [4] outlined a new framework of the SVR-FFA model for more accurately predicting the maximum depth of pitting corrosion in oil and gas pipelines. Although these models have achieved high accuracy to a certain extent, such methods are also more demanding with regard to the quality and quantity of pipeline inspection data, which are not always consistent due to the diversity of field conditions and the limitations of inspection technology. As a result, this complicates the process of building models based on deep learning techniques and makes the application and extension of these methods more difficult [17,18]. Due to the limitations of data acquisition from real-world pipeline systems, there is a need for high-quality lab-scale experimental data.

Experiments in the dynamic reactor [19] allow corrosion data to be obtained under laboratory conditions; however, such methods tend to be both expensive and time-consuming, and as such can only provide limited data. To overcome the problem of small datasets, some researchers have turned to deep learning methods to obtain reliable prediction models. For example, Zhu et al. and Angshuman Paul et al. [20,21] proposed models applicable to the prediction of image data. Chen et al. [22] proposed the use of an ensemble long short-term memory (EnLSTM) model for time-series data. Although these models do lead to greater prediction accuracy for small sample sizes, they cannot fully overcome the inherent disadvantages of NN overfitting and weak generalizability [23], which means that such models do not train reliably with small datasets [20]. Some studies have found that the use of data mining techniques that build and combine individual learners to form a strong learner capable of better predictions (also known as ensemble learning methods), play an important role in overcoming the overfitting problem [24], and are effectively able to handle data sets with high dimensionality, complex structures, and small sample sizes [25]. Such methods are even helpful in solving the problem of unbalanced data distribution [26]; as a result, many researchers have directed their efforts toward ensemble learning. For instance, Dvornik et al. [27] demonstrated a significant reduction in the variance by integrating a distance-based classifier in a small-sample setting. Mahdavi-Shahri et al. [24] proposed an ensemble learning method to achieve the best classification results for a multi-label problem with small samples. Guan et al. [28] used ensemble techniques to improve the accuracy of face recognition under small-sample conditions. As the main ensemble learning methods, the bagging (bootstrap aggregation) [29] and boosting [30] algorithms, have been applied in many fields such as predicting cloud meteorological data [31], concrete bearing pressures [32], and the mapping of ecological zones in aerial hyperspectral images [33]. Bagging and KNN both perform well in small

sample data sets, and scholars [34–36] have also studied the models integrating them. However, it is still worth exploring whether the integrated model can predict small sample data sets of oil industry. Against this backdrop, a proposed framework for predicting corrosion rates under a small sample of metal corrosion data in the laboratory is proposed. The innovations and contributions are as follows:

- Different from other ensemble models, a KNN-based learner's bagging model is proposed. This model has better performance on the small sample data set of this experiment, obviously outperforming the traditional model.
- The ensemble models of bagging, boosting, and stacking are all used for comparison. In this experiment, bagging is slightly superior to other ensemble models.
- Various factors affecting the experimental results were studied.

Section 2 describes the features of the data involved in the experiment and how to obtain the data. Section 3 introduces the process and method of the experiment. In Section 4, the influencing factors such as data dimension, segmentation rate, and noise reduction processing are discussed, the prediction errors of the integrated model under this data set are compared, and the advantages of the integrated model have been experimented with. Section 5 discusses the experimental results of Section 4. Some primary conclusions are summarized in Section 5.

2 Materials and Experimental Database

For the laboratory experiments in this study, a dynamic reactor apparatus was used (Fig. 1), with a solution consisting of 3 L of water obtained from a shale gas gathering pipeline. The operating parameters inside the reactor, including pressure, temperature, and partial pressure of CO_2 were controlled. In each group of experiments, four samples of the L360 N pipeline ($50 \times 10 \times 3$ mm^3) were used to measure uniform corrosion. The experimental protocol was as follows: firstly, the reactor body and lid were sealed, then the inlet and release valves on the lid were opened and nitrogen gas was passed through for two hours. Next, the release valve was closed and CO_2 and O_2 were injected. Finally, once the reactor pressure had increased to 5 MPa through further injecting N_2, the inlet valve was also closed. The experimental period was 7 days, and the mass difference of the metal samples before and after the experiment was divided by the reaction time to obtain the average corrosion rate (weight loss tests [37]). Then, the experiments were repeated under different conditions [38,39]. Then, the multiphase flow simulation software package OLGA [40] was used to expand the experimental parameters, including liquid flow rate, temperature, inclination angle, CO_2 partial pressure, and H_2S partial pressure, in order to explore the influence of 18 other parameters including flow pattern, flow rate and shear stress on the corrosion rate. The name, abbreviation, unit, minimum/maximum value, average value, and standard derivation (SD) of the experimental data sets obtained by this method are shown in Table 1.

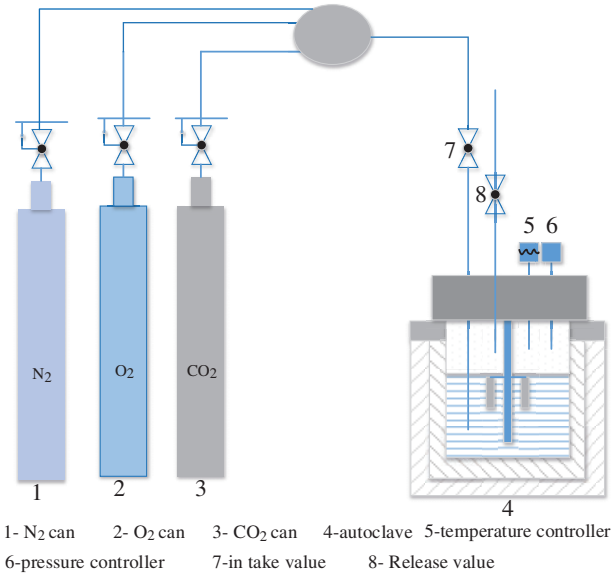


Figure 1: Schematic diagram of the dynamic reactor apparatus [38]

Table 1: Parameters in total in the experimental data sets

Parameter	Code	Unit	Minimum	Maximum	SD
Flow type*	a	——	1	3	——
Flow	b	m^3/d	2.48	15.6	3.18
Gas shear stress	c	bar	1.94E − 05	4.11E − 05	9.59E − 06
Liquid shear stress	d	bar	9.61E − 10	0.00012	2.92E − 05
Gas flow rate	e	m/s	1.21	6.39	1.3
Liquid flow rate	f	m/s	0.0025	1.41	0.36
Superficial gas flow rate **	g	m/s	1.2	6.14	1.17
Superficial flow rate of liquid **	h	m/s	0.0009	0.0036	0.0007
Solid phase deposition rate	i	kg/m^3-s	0	0.0064	0.0014
Gas flow	j	m^3/d	3312.52	26257.18	6087.2
Inclination	k	°	1.15	2.1	0.17
Liquid holding rate ***	l	——	0.005	0.38	0.09
CO_2 partial pressure	m	bar	5.21	6.59	0.32
pH	n	——	4.64	4.72	0.019
Temperature	o	°C	62.67	78.88	3.66
Gas density	p	kg/m^3	55.96	71.14	3.66
Liquid density	q	kg/m^3	1001.28	1002.35	0.26
Liquid surface tension ****	r	N/m	0.0014	0.006	0.001
Gas-phase thermal conductivity	s	W/(m·°C)	0.038	0.04	0.00051
Liquid-phase temperature	t	°C	24.5	29.9	1.39

(Continued)

Table 1 (continued)

Parameter	Code	Unit	Minimum	Maximum	SD
Gas viscosity	u	cP	0.014	0.014	0.00019
Liquid viscosity	v	cP	0.8	0.9	0.023
H_2S partial pressure	w	bar	3.78	4.76	0.235
Corrosion rate	x	mm/a	0.089	0.59	0.12

Note: ∗ Flow type: the OLGA classifies flow into four types: stratified flow, annular flow, segmental plug flow, and bubble flow, which were denoted as 1, 2, 3, and 4, respectively, for this dataset. ∗∗ Apparent flow rate refers to a virtual (artificial) flow or a single fluid flow velocity (known as the apparent gas or liquid velocity depending on the type of fluid). ∗∗∗ Liquid holding rate is also known as the true liquid content rate or cross-sectional liquid content rate, refers to the proportion of the cross-sectional area of the liquid phase to the total cross-flow area in the process of water and gas flow. ∗∗∗∗ Liquid surface tension: the force that acts on the surface of a liquid to reduce its surface area.

In order to better visualize the correlation between the parameters and the corrosion rate, a Pearson correlation coefficient matrix was drawn (Fig. 2). The graph is used to show the linear correlations between parameters, where positive and negative values represent positive and negative correlation, respectively. As shown in the figure, the first five correlations are the solid phase deposition rate and liquid performance flow rate (0.8), the gas flow rate and liquid surface tension (0.78), the liquid density and liquid viscosity (0.77), the partial pressure of carbon dioxide and gas viscosity (0.77), the liquid-phase temperature and liquid viscosity (0.77).

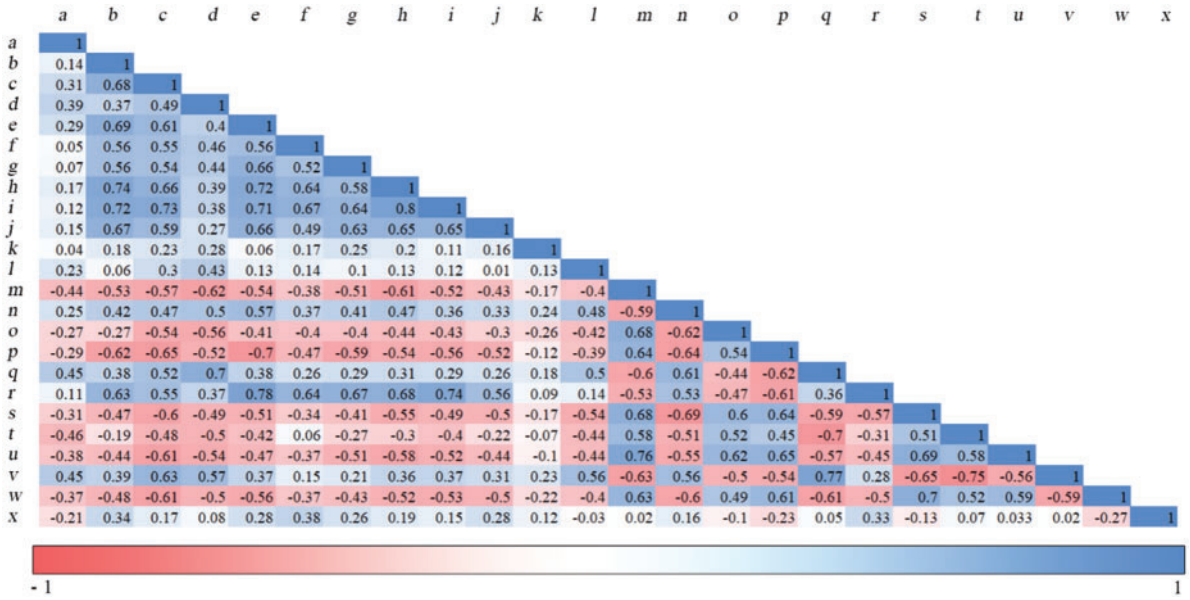


Figure 2: The interaction indices were calculated by interaction detector (big value means strong interaction)

3 Establishment and Methods of the Framework

3.1 Bagging Algorithm

Ensemble learning is a very important area of artificial intelligence and data mining, which aims to build an integrated model by combining individual learners to improve the overall performance [41]. In terms of integration approaches bagging and boosting (the adaptive boosting (AdaBoost) algorithm [42] is the most commonly-implemented type of boosting algorithm) are two representative models of ensemble learning. Bagging is one of the first ensemble learning algorithms and uses a parallel integration strategy to randomly select different subsets of the training data. Each subset is then trained based on the same individual learners, and the final prediction results are obtained using a minority-majority approach for the classification problem and a simple average approach for the regression problem [43,44]. The bagging algorithm can improve generalization by reducing the variance error [45]. The integration steps are as follows: Suppose we have a training set

$$D = \{(x1,y1), (x2,y2), \ldots, (x\mathrm{m},y\mathrm{m})\} \tag{1}$$

The bagging algorithm first resamples the data set, creates a new training subset, and then puts the sampled data into the original data set again. Although this approach may result in some training samples being selected multiple times, while others may not be selected at all, this does not affect the prediction performance of the model. After selecting the individual learning algorithm, put each training subset into the algorithm for calculation.

$$ht = L(Dt) \tag{2}$$

Combining multiple individual learners to build a learner with stronger predictive ability in the regression problem, the combination strategy is to simply average the prediction results of each weak learner.

$$H(x) = \frac{1}{T}\sum_{i=1}^{T} hi(x) \tag{3}$$

In Eqs. (1)–(3), D is the training dataset, $(xi, yi)(i = 1, \ldots, m)$ is the sample in the training dataset, m is the total number of samples, xi is the features of the input data, yi is the label value of the sample, ht is an individual learner,Dt is the subset constructed after resampling, t is the number of samples, L is the individual learning algorithm, $H(x)$ is a learner with stronger predictive ability after integration.

3.2 KNN Algorithm

Individual learners refer to algorithmic models with simple structures. Generally speaking, in the bagging integration strategy, the individual learners are called base learners, and in the boosting integration strategy, they are called component learners [44]. For convenience, we collectively call them individual learners.

The main idea behind this algorithm is that the more similar things are, the more likely they are to be adjacent to each other, and it obtains the maximum possibility of the data type of the current point by looking for the data category with the most adjacent data points. This algorithm was chosen because of its simplicity and the ability to distinguish it from other algorithms through the idea of distance sampling [46]. The advantage of the algorithm is that it is simple and insensitive to outliers. The disadvantage of this algorithm is that it has high time complexity and spatial complexity, and its interpretation ability is not strong. However, under the condition of small samples, the computational

pressure of the algorithm is greatly reduced. Fig. 3 shows the forecasting principle of the KNN algorithm.

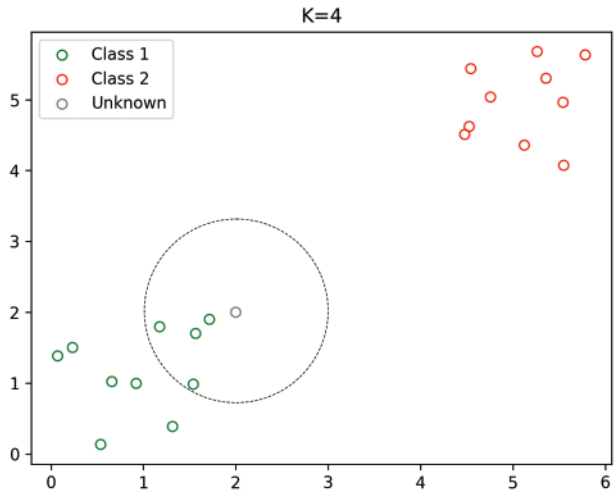


Figure 3: Schematic diagram of KNN (K indicates the number of selected neighbors) [47]

3.3 Framework Building and Experimental Process

Because the model used for comparison has the same construction steps as this framework, we describe its construction process together in this section, and Fig. 4 illustrates the overall design flow of this study.

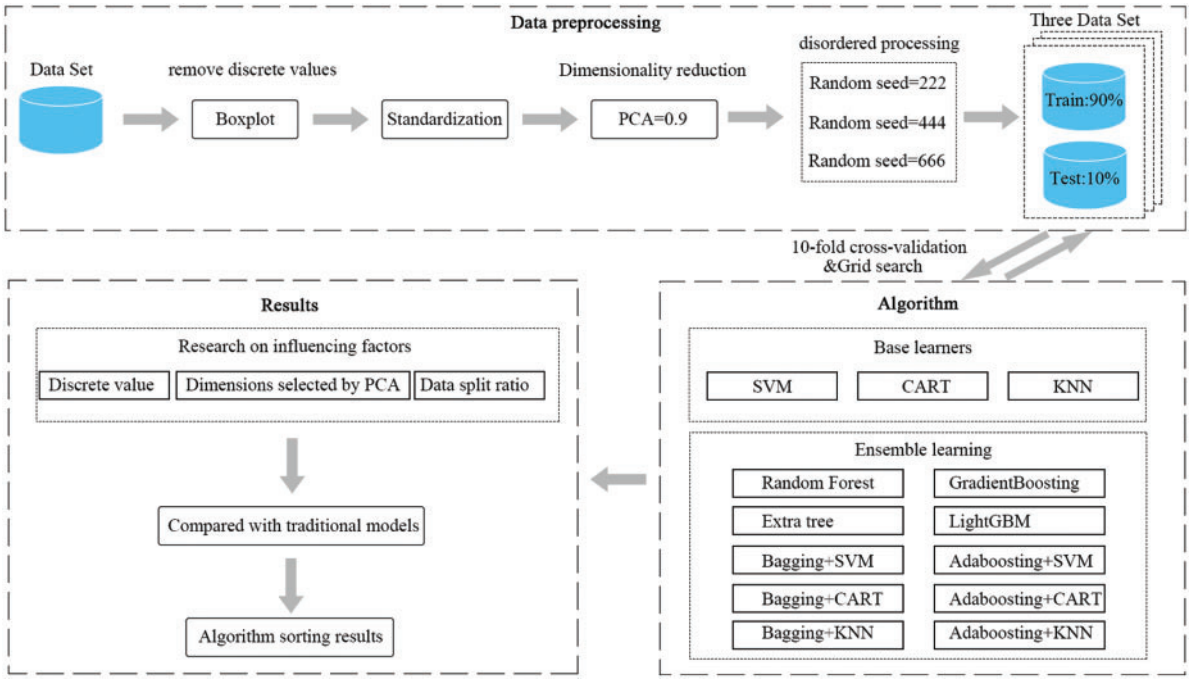


Figure 4: Flow chart of the framework and experimental process

In order to make the model comparison more convincing, we use K-nearest neighbor (KNN) [48], Support vector machine (SVM) [49], and Classification and regression trees (CART) [50] as individual learners for bagging and boosting integration, and in addition, we choose four more mature integrated models of Random forest (RF) [51], Extra tree [52], Gradient boosting [53] and Light gradient boosting machine (LightGBM) [54] for comparison. Based on these four models, our experiments first explored the influence of factors such as eliminating discrete values, reducing dimensions, and changing the proportion of data segmentation on the experiment, which provided the basis for data preprocessing in the framework construction process.

According to the exploration results of the influencing factors, in the aspect of data preprocessing, firstly, according to the corrosion rate, the experimental data set was eliminated by box-plot [55]. Considering the different divisions of the training set and the test set under the condition of small samples, the results may be quite different. Before data splitting, three random seeds (222, 444 and 666, respectively) are used for out-of-order processing, to ensure that there are multiple groups of experiments to compare under the same data set. The data was then split into training and test sets at a ratio of 9:1 [56], and after the data was uniformly normalized, Principal Component Analysis (PCA) [57] dimensionality reduction was performed to retain more than 90% of the information.

During the process of model building, a combination of 10-fold cross validation [58] and grid search was used to determine the optimal hyperparameter of each model. It was also debugged as a hyperparameter, and the grid search was carried out within a certain range. It was worth noting that we separately search the basic model and the integration method in the grid, and then integrated them according to the selected optimal hyperparameter. Firstly, the training data was divided into 10 subsets on average, of which 9 subsets were used for model training and the rest were used for model verification. This operation was repeated 10 times in a row, the average error was obtained, and then all the parameters in a certain range were traversed by the grid search method. Finally, the parameter combination with the smallest error in 10-fold cross-validation was obtained. The purpose of this process is to achieve the best predictability for each model and to minimize the influence of random seeds on the accuracy of the model. The generalization ability of each model was then tested on the test set and its error was calculated on the test set using mean square error (MSE), mean absolute percentage error (MAPE), and mean absolute error (MAE) metrics, where MSE is the main evaluation index. Finally, the Friedman ranking [59] was used to test the advantages and disadvantages of the model.

To verify the superiority of the model, we also compared the traditional empirical model with the more complex stack integration model and discussed whether the integrated basic learner can improve the performance.

4 Results

This section shows a comparison between the results for the different ensemble learning methods under the conditions of a small sample dataset, followed by the analysis of the effects of discrete values, PCA dimensionality reduction, and data splitting ratio on the prediction. To verify the superiority of the ensemble algorithm approach, the prediction results for the ensemble learning models are compared with those of the weak classifier and traditional models. It is noted that, because of the smaller number of data sets in this experiment, the efficiency of calculation models (e.g., run-time) and not included in the comparison index.

4.1 Ensemble Learning Model Prediction Performance Comparison Results

MSE, MAE, and MAPE were used to determine the error value of the integrated algorithm on the test set. The differences between the predicted and true values of the bagging and boosting algorithms based on MSE are demonstrated in Fig. 5. The closer the result is to the diagonal, the smaller the error between the two groups of data. The results show that when the random seeds were 222 and 444, the extra trees, bagging + KNN, and AdaBoosting + SVR models showed the minimum MSE values. However, when the random seed was 666, this changed to bagging + CART, AdaBoosting + CART, and AdaBoosting + SVR. This indicates that the prediction performance of the model changes significantly depending on the training and test sets used.

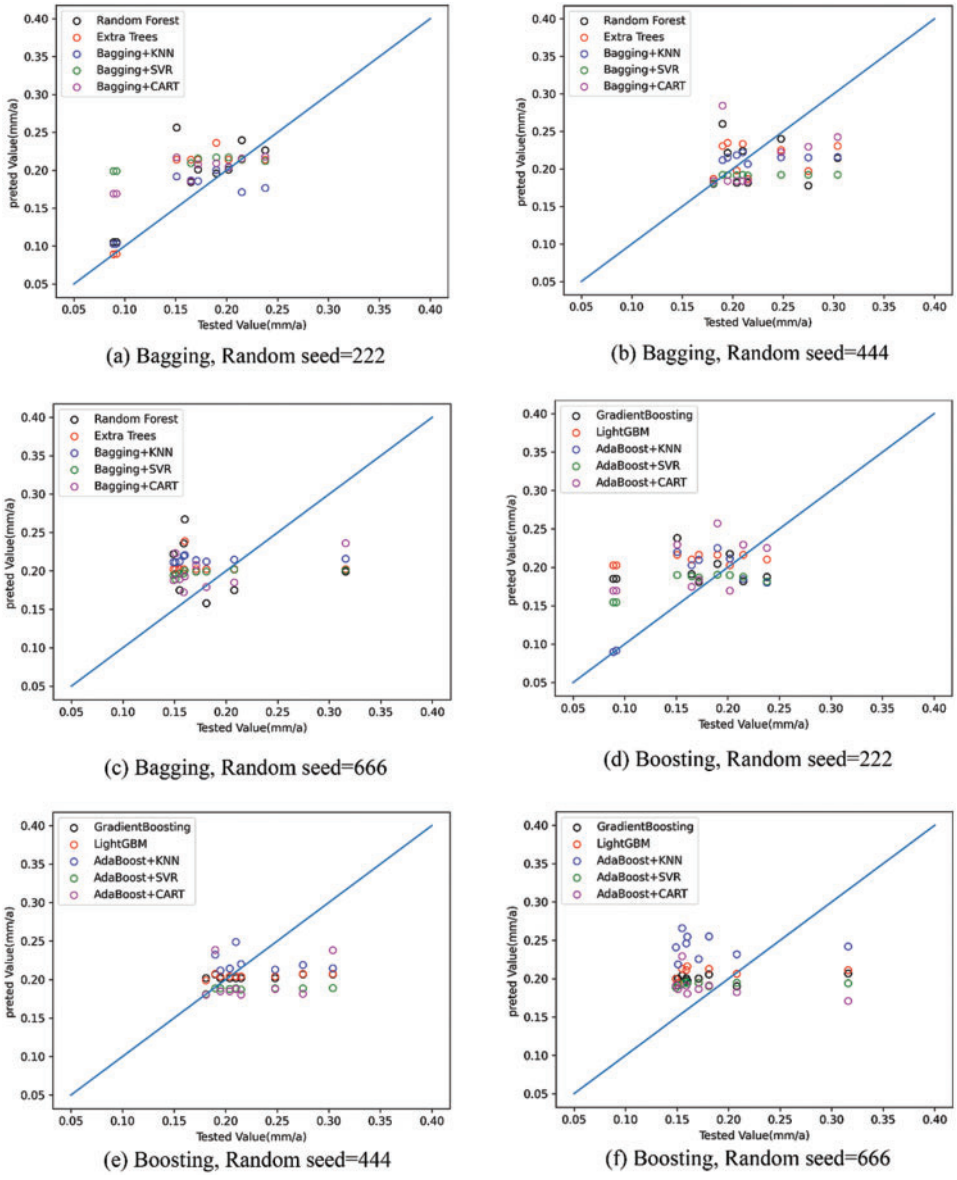


Figure 5: Comparison between model predictions and experimental values

To evaluate the overall performance of each model on multiple datasets, a Friedman ranking plot was drawn (Fig. 6). The blue dots in the graph indicate the average ranking of the algorithm, and the farther to the left the numerical value is, the better the model is. The results show that the prediction performance of the bagging + KNN algorithm is the best, followed by extra-trees and bagging + CART, and the prediction performance of random forest was the lowest. The horizontal lines in the figure indicate the allowable fluctuation range of each algorithm ranking. If the horizontal lines of a given pair of algorithms do not overlap each other, it indicates that there are significant differences between these algorithms. However, here, the algorithms in the graph all overlap with each other, indicating that there is no significant difference between them. The order of all algorithms from good to bad is bagging + KNN, extra trees, bagging + CART, adaboosting + KNN, adaboosting + CART, adaboosting + SVR, bagging + SVR, Gradient boosting, LightGBM, Random Forest.

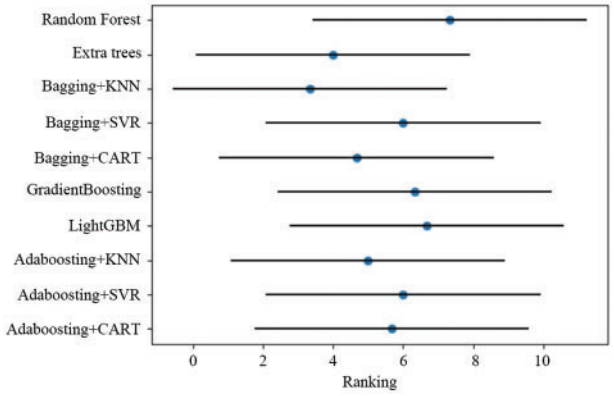


Figure 6: Friedman diagram for the different ensemble learning algorithms

4.2 Experimental Results for Exploring the Influencing Factors

4.2.1 Error Comparison Results after Eliminating Discrete Values

The box-plot uses the quartile as a boundary for analysis of the distribution characteristics of the data, and the data for the corrosion rate term was used as the basis for drawing this plot (Fig. 7). The red dashed line in the scatter plot indicates the split line, and the red points represent the discrete points (defined as >0.42) that need to be eliminated by the box-plot. The total number of discrete points determined using this method is 11.

Among the models used for comparison, we uniformly set their hyperparameters to n_estimators = 100 and random_state = 222 and set the learning rates of the gradient boosting and LightGBM models to 0.1. Then, the data before and after excluding discrete values were substituted into these models, the results of the 10-fold cross-validation were recorded as shown in Fig. 8. The results show that after removing outliers, the average prediction errors of random forest, extra-tree, gradient boosting, and LightGBM algorithms are reduced by 64.16%, 68.50%, 62.88%, and 63.81%, respectively. This shows that the use of box-plot to eliminate discrete values can help improve the prediction performance of the model.

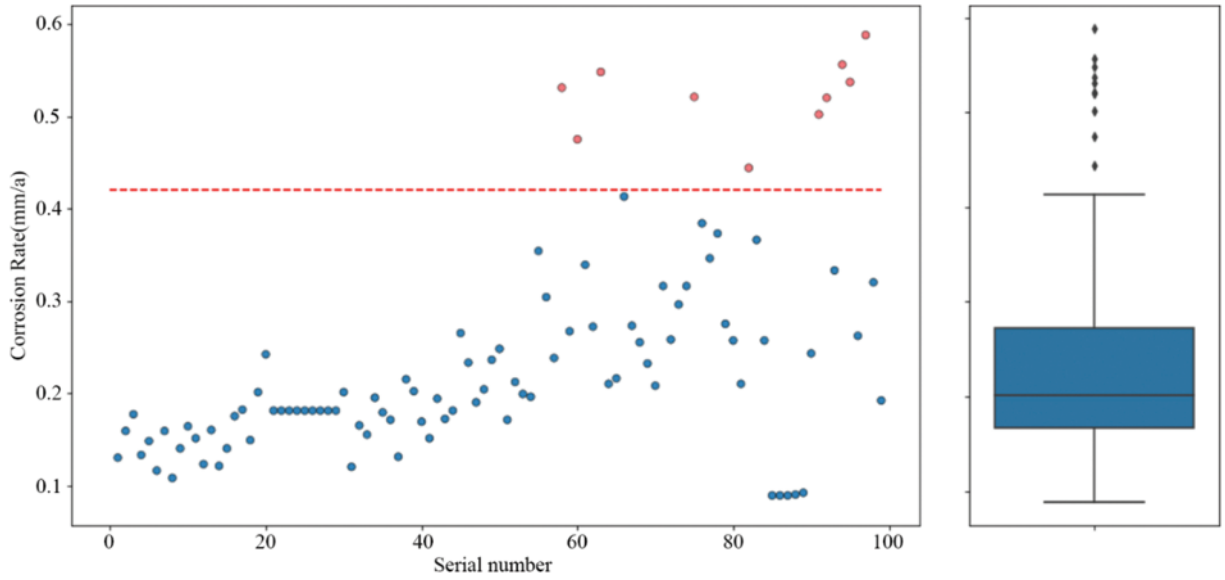


Figure 7: Scatter plot and edge box-plot of the corrosion rate

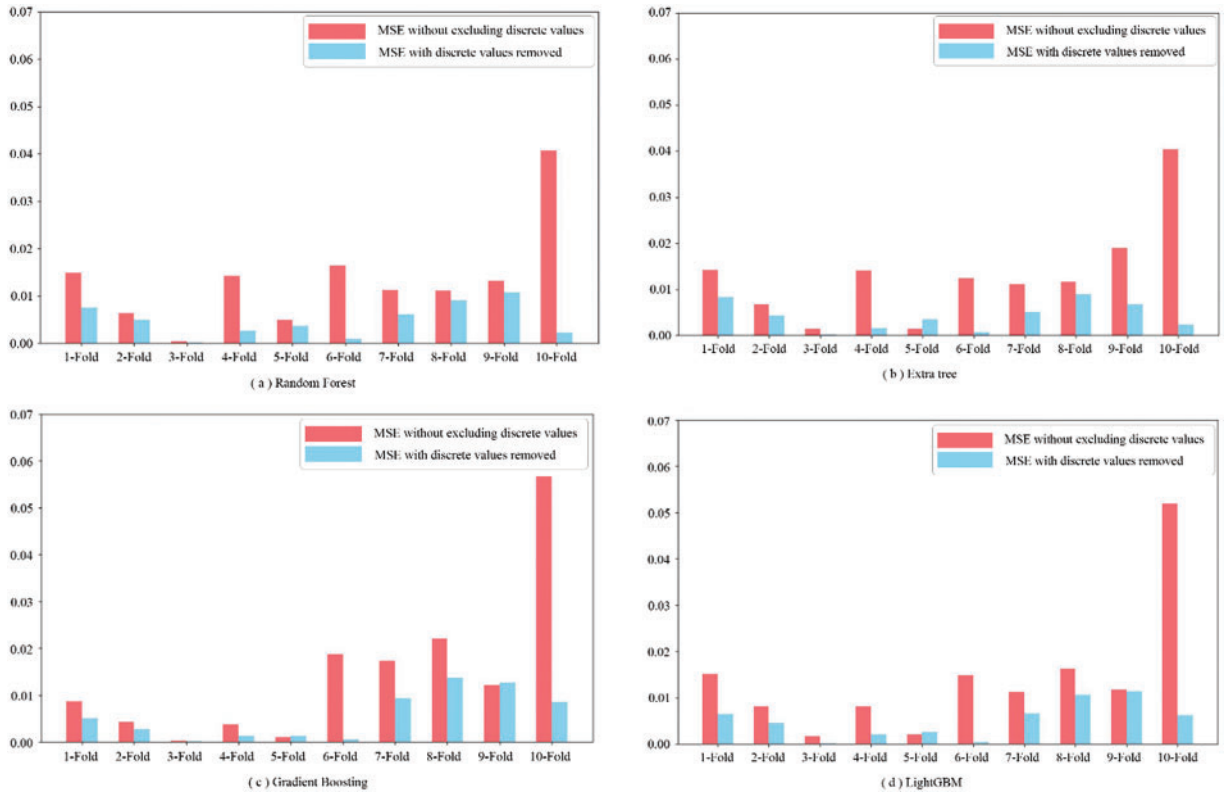


Figure 8: 10-fold cross-validation histograms before and after removing the discrete values

4.2.2 Error Comparison Results after Using PCA to Reduce the Dimensionality

In general, although the dataset needs multiple dimensions of features to be characterized as exhaustively as possible, it is not always advantageous to have more features, as the presence of some features may cause the prediction result to drop. In other words, some poorly correlated parameters can be removed [60]. In this data set, the percentage of each principal component, wherein the information interpretation rate of the first principal component is 55.68%, and the corresponding cumulative sums of principal components are the sums of the first 14, 11, and 8 principal components when 95%, 90%, and 85% of the information is retained, respectively.

In this part, only the dataset with an out-of-order seed of 222 was selected when the data was split, and PCA was used to divide this dataset into three-dimension data with dimensions retaining 95%, 90%, and 85% of the information. The hyperparameter involved in the model was set to random_state = 222, and the learning rate of the gradient boosting and LightGBM was 0.1. 19 numbers from 10 to 200 were moderately spaced, and these numbers were set to n_estimator. Finally, the evaluation value of the predicted results of these models was used as the basis for model comparison. Fig. 9 shows the results of the calculations when no principal component scaling is used. Table 2 shows more details. This shows that the error value is significantly higher when the proportion of principal components was selected as 95% as opposed to 90% and 85%; further, the error values for the latter two cases are similar, except that the MSE value of the extra tree model with 90% principal components (0.001545) was smaller than that with 85% principal components (0.002592). Therefore, although PCA is a commonly-used method for achieving data dimensionality reduction under small sample conditions, there is no uniform standard for how much information should be retained in different scenarios, and 90% was found to be the optimum value in this study.

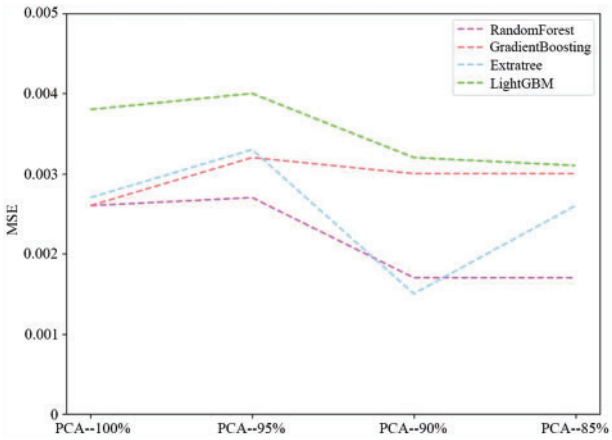


Figure 9: Prediction error of each model after PCA dimensionality reduction

4.2.3 Error Comparison Results for Different Data Segmentation Ratios

Different models were used to compare the results of the prediction errors for different split ratios. As shown in Fig. 10 and Table 3, the results show that when the split ratio is 9:1, these models predict

the minimum average error, where the MSE average of RF is 0.0017, Gradient Boosting is 0.0026, Extra tree is 0.0024, and LightGBM is 0.0025. When the split ratio is 8:2 and 7:3, the performance on different models is also different.

Table 2: MSE results for the different principal components

Algorithm	Principal components	The MSE result of 10 random numbers		
		Min	Max	Mean
Random forest	100%	0.0024	0.0028	0.0026
Random forest	95%	0.0024	0.003	0.0027
Random forest	90%	0.0015	0.0023	0.0017
Random forest	85%	0.0015	0.003	0.0017
Gradient boosting	100%	0.0023	0.003	0.0026
Gradient boosting	95%	0.0029	0.0035	0.0033
Gradient boosting	90%	0.0029	0.003	0.003
Gradient boosting	85%	0.0029	0.003	0.003
Extra tree	100%	0.0024	0.003	0.0027
Extra tree	95%	0.003	0.0041	0.003
Extra tree	90%	0.0012	0.002	0.0015
Extra tree	85%	0.0023	0.0034	0.0026
LightGBM	100%	0.0035	0.0039	0.0038
LightGBM	95%	0.0036	0.0042	0.004
LightGBM	90%	0.0025	0.004	0.0032
LightGBM	85%	0.0025	0.0039	0.0031

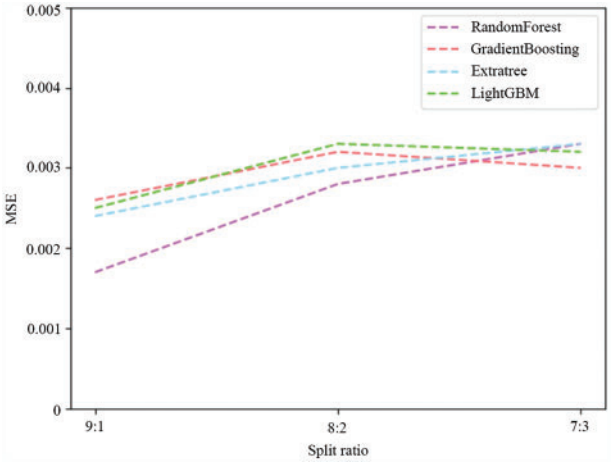


Figure 10: Prediction error of each model under different split ratios

Table 3: Detailed parameter values of MSE results for different split ratios

Algorithm	Split ratio	MSE results			
		Random_state = 222	Random_state = 444	Random_state = 666	Mean
Random forest	9:1	0.0016	0.0016	0.0018	0.0017
Random forest	8:2	0.0021	0.0037	0.0027	0.0028
Random forest	7:3	0.0037	0.0033	0.0031	0.0033
Gradient boosting	9:1	0.0029	0.0023	0.0025	0.0026
Gradient boosting	8:2	0.0028	0.0046	0.0023	0.0032
Gradient boosting	7:3	0.0032	0.0033	0.0025	0.0030
Extra tree	9:1	0.0022	0.002	0.0029	0.0024
Extra tree	8:2	0.0024	0.0034	0.0031	0.003
Extra tree	7:3	0.0031	0.0036	0.0033	0.0033
LightGBM	9:1	0.0031	0.0013	0.0031	0.0025
LightGBM	8:2	0.0029	0.0037	0.0032	0.0033
LightGBM	7:3	0.0038	0.003	0.0029	0.0032

4.3 Comparison Results of Prediction Errors between Individual Learners and Ensemble Models

To investigate whether the ensemble learning models performed better than the individual learners, the results of integrating the individual learners KNN, SVR, and CART with the bagging and AdaBoosting algorithms were compared in three randomized segmentation experiments (Fig. 11). These results show that the bagging algorithm reduces the MSE error values when the individual learners are KNN and the random seed is 222 or 444, while AdaBoosting increases the error value under all three random seeds; further, when the individual learners are SVR, except for the case with a random seed of 666, AdaBoosting significantly reduced the error, and in other cases, bagging and AdaBoosting had similar error values as the individual learners. However, when the individual learners are CART and the random seed is 444 or 666, both bagging and AdaBoosting reduce the prediction error values, while the opposite is observed for a random seed of 222.

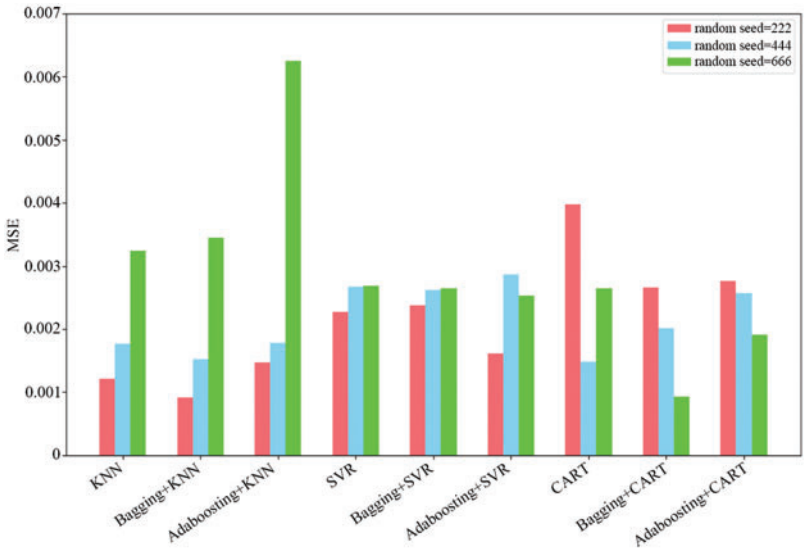


Figure 11: Comparison of the MSE values of different algorithms

4.4 Comparison Results of Prediction Errors between Ensemble Methods and Traditional Models

A traditional model for oil and gas pipeline corrosion rate prediction, the de Waard model [5,6], is given by the following equations:

$$\frac{1}{Vcorr} = \frac{1}{Vr} + \frac{1}{Vm} \tag{4}$$

$$\log_{10} Vr = 4.93 - \frac{1119}{t+273} + 0.58\log_{10} Pco2 - 0.34(pHact - pHco2) \tag{5}$$

$$pHco2 = 3.82 + 0.00384t - 0.5\log_{10} Pco2 \tag{6}$$

$$Vm = 2.45\frac{Vl^{0.8}}{d^{0.2}} Pco2 \tag{7}$$

In contrast to the de Waard model, the Norsok M506 model [9] uses different equations over different temperature intervals. In the temperature range from 20°C–150°C, the model is given by:

$$Vcorr = Ktf_{co2}^{0.62}\left(\frac{\tau w}{19}\right)^{0.146+0.0324\log_{10} fco2} f(pH)t \tag{8}$$

Between 15°C–20°C, the model is given by:

$$Vcorr = Ktf_{co2}^{0.36}\left(\frac{\tau w}{19}\right)^{0.146+0.0324\log_{10} fco2} f(pH)t \tag{9}$$

Between 5°C–15°C, the model is given by:

$$Vr = Ktf_{co2}^{0.36} f(pH)t \tag{10}$$

In Eqs. (4)–(10), $Vcorr$ is the corrosion rate in mm/a, Vr is the reaction rate in mm/a, Vm is the mass transfer rate in mm/a, t is the media temperature in °C, $Pco2$ is the CO_2 partial pressure in MPa, $pHact$ is the actual pH, $pHco2$ is the pH of CO_2-saturated solvent, Vl is the liquid phase flow rate of the medium in m/s,d is the pipe diameter in m, $fco2$ is the fugacity of CO_2, Kt are temperature-dependent constants, τw is the wall shear force in Pa, and $f(pH)t$ is the pH-dependence term.

As the parameters of the de Waard model include the pipe diameter, the effect of which cannot be measured under laboratory conditions, the Norsok M506 model was selected as the control model for this study, as it is a purely empirical model based on a large amount of experimental data. The remaining data after the exclusion of the discrete values were selected for the comparison of prediction results, and the MAE value was used as the performance metric. For the ensemble learning algorithms, the average values of MAE for each model under three random seeds were compared. Fig. 12 shows that the MSE error value of the bagging + KNN framework recommended in this study is only 0.0346, while the error value of the traditional Norsork M506 model is 0.1322. This framework is also slightly better than other ensemble models.

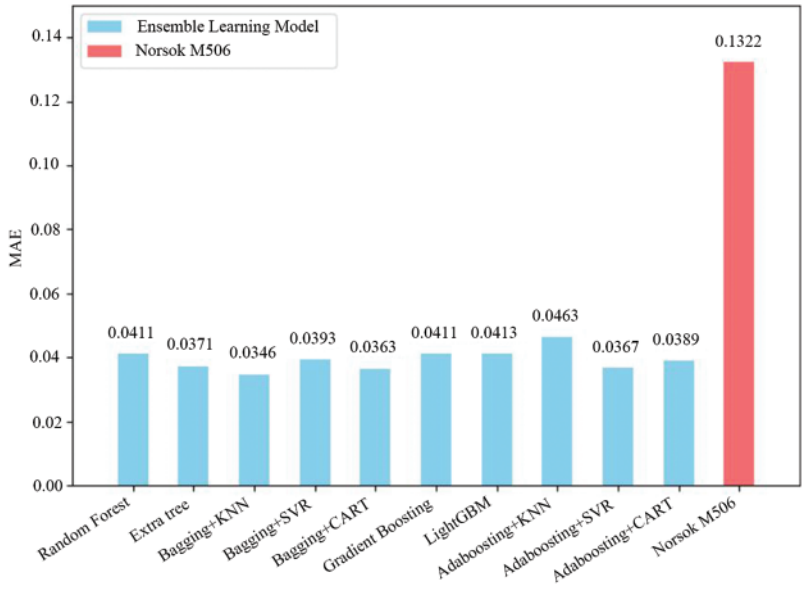


Figure 12: Error comparison results between the traditional and ensemble models

4.5 Comparison Results of Prediction Errors between Ensemble Methods and Stacking Models

The Stacking [61] model allows any combination of different types of models. The integration strategy is to first use the K-fold cross-validation method on the original data to predict the model that needs to be integrated. After obtaining the predicted value of each integrated unit, the average value of the predicted results is used as input into a simpler prediction model (the purpose of this is to prevent overfitting), and the predicted value is the final result. The ensemble units of this ensemble model are usually selected as bagging model and boosting model. In this section, a total of 11 ensemble models (with the addition of the XGBoost [62] model) were used as base models to be integrated into the Stacking mode, in this experiment, to control the number of combinations, each model was only used once. Fig. 13 shows the prediction errors under different combinations. From this figure, it can be seen that with the increase of the number of combinations, the prediction error values become more

concentrated, and almost every segment has similar error variation rules. Table 4 shows the minimum values of MSE for different combinations and the percentage of their predictions that are improved (relative to the models involved in the combination). When the random splitting seed is 666, under the premise that the prediction error of the basic model is larger, the integrated prediction error is further enlarged, which leads to the fact that the prediction performance of almost all models hardly improves under this data set partition condition. If only the random splitting seeds 222 and 444 are analyzed, as the increase of integration times, the probability that the prediction effect after integration is better than that of a single sub-model will gradually decrease. These results show that even if there is a combination of better prediction effects in stacking, regardless of the prediction performance of the model or the proportion of prediction performance improvement, the stacking integration mode is not very suitable for small sample prediction under this data.

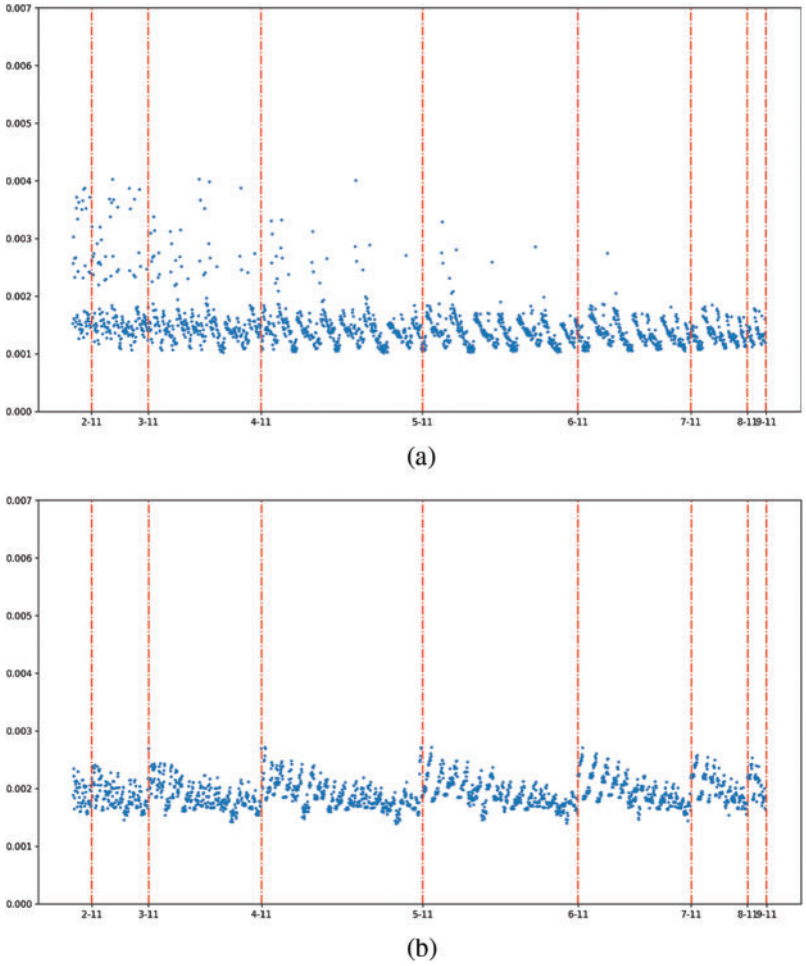
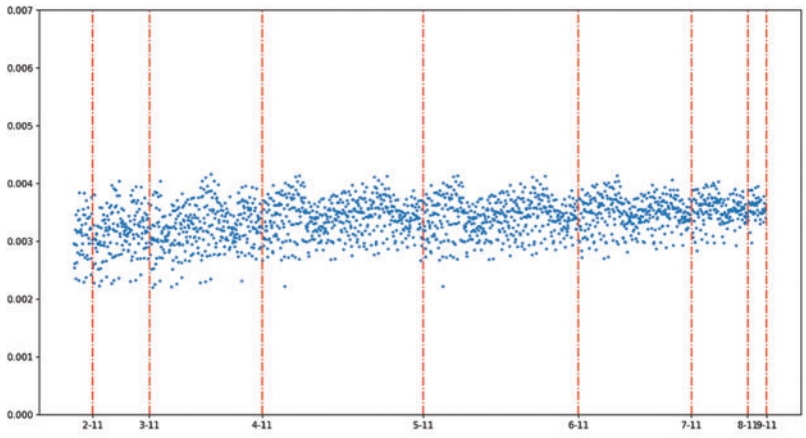


Figure 13: (Continued)



(c)

Figure 13: Combined prediction error plot in stacking model. a, b, c indicate the prediction results when the random splitting seed is 222, 444, 666, respectively; the horizontal coordinates of the graph indicate the number of models involved in the combination

Table 4: Comparison table of prediction performance of each combination in Stacking model

Number of combinations	Random_state = 222		Random_state = 444		Random_state = 666		Average
	Minimum of MSE	Percentage of result improvement	Minimum of MSE	Percentage of result improvement	Minimum of MSE	Percentage of result improvement	
2	0.001262	11/55	0.001559	23/55	0.002294	9/55	0.31
3	0.001067	30/165	0.00146	49/165	0.002227	7/165	0.24
4	0.001022	57/330	0.001419	76/330	0.0022	2/330	0.20
5	0.001016	77/462	0.001391	71/462	0.002218	0/462	0.16
6	0.001023	71/462	0.0014	48/462	0.002218	0/462	0.13
7	0.001045	45/330	0.001439	18/330	0.0027	0/330	0.10
8	0.001072	18/165	0.001538	3/165	0.002833	0/165	0.06
9	0.00112	3/55	0.001635	0/55	0.002973	0/55	0.03

5 Discussion

5.1 Predictive Performance of the Framework

In this study, the ensemble learning algorithms with the smallest MSE were found to be bagging + KNN, extra trees, and bagging + CART, which were all based on the bagging framework. This shows that the bagging algorithm is superior to boosting algorithm in small data set prediction. The results obtained here are consistent with Wang et al. [44], those who found that the bagging method was superior to boosting method in classification, and the former performed relatively better in the presence of noise. However, when using ensemble learning to analyze remote sensing images,

Both Chan et al. and DeFries et al. [63,64] found that the AdaBoost algorithm had higher accuracy and that the bagging algorithm is more stable. The reason for the differences between the aforementioned studies and the present study is that in the latter, there are some instances where the discrete values (i.e., noise) are not completely removed, which is more detrimental to boosting methods [65]. The differences between the optimization strategies of the two algorithms also contributed to their different prediction performance. The bagging algorithm uses a parallel strategy to average the errors of some discrete values while boosting uses a serial strategy to increase the weight of the discrete value errors present in the training set during the optimization process. It is worth noting that among different models, the random forest algorithm has the lowest prediction rank, which is mainly due to the large prediction error of the last random seed. Bagging + KNN framework has achieved the best prediction results, but under the condition of small samples, data plays a vital role in the results, so the performance of other data sets is still worth exploring.

5.2 Exploratory Analysis of Influencing Factors

In this experiment, the prediction error of each model is reduced after the discrete values are eliminated, because when the amount of data with some special features is small, the distribution of these data relative to other data forms a small disjunctive area which may not be visible [66,67]; Therefore, it is not easy to determine whether the values in these areas are true values or noise [68], and the existence of such values leads to insufficient support of the prediction bounds, making it more difficult for the learning algorithm to achieve good generalization [26]. Therefore, excluding these values not only reduces the number of small intermittent areas but also has a positive impact on the model's generalization (the model's generalization is inherently weak under the condition of small samples). Admittedly, the model fitting in this study has some defects in the elimination of discrete value. Discrete rejection based on corrosion rate alone would have resulted in the incomplete rejection of discrete values as well as possible rejection of non-discrete values. The reason why this culling is not based on the feature values is that the data set has multiple dimensions, and discrete culling of each feature may lead to a small sample size, thus leading to a larger prediction errors. In the model applicability evaluation, when the random seed of the control partition is 666, the prediction error increases significantly, which is likely due to the discrete values in the test set being mixed with those that have not been removed. Therefore, when the study sample is small, more rigorous rules for determining the discrete values should be established. In addition, when building a prediction model based on small data set, when summarizing the model, the verification data should be kept as small as possible with respect to the training set.

The research shows that dimension reduction is helpful to improve the prediction accuracy of the model. In our experiment, when 90% of the information dimension are retained, the prediction effect is the best, while when other information dimensions are retained, the result fluctuates greatly. This shows that although PCA is a common method for dimensionality reduction of small sample data [69], there is no uniform standard for how much information should be retained in different scenarios, which should be selected according to the actual situation.

In the experiment of the influence of segmentation ratio on the prediction error, we found that with the decrease of the training set ratio, the prediction error becomes larger. This is because when the number of training sets decreases, over-fitting is more likely to occur, which will lead to poor generalization ability of the model. This is consistent with the current mainstream research [67], so whether it is a small sample condition or not, increasing the amount of training data will always help to predict the results.

5.3 Comparison with Other Models

Although many researchers believe that ensemble learning is superior to individual learners and helps solve small-sample problems [27], the results from Fig. 11 shows this is not necessarily the case. Windeatt [70] put forward that, to obtain a good integration, the conditions of accuracy and diversity must necessarily be satisfied. Therefore, when the dataset is small and contains some discrete values, ensemble learning methods should not be used blindly. In such cases, if the bagging algorithm is not able to achieve good prediction results, it is worth trying individual learners.

Fig. 12 shows that the predicted MAE value of the Norsok M506 model was 0.1322, which is significantly larger than that of the ensemble learning algorithm. This indicates that ensemble learning still performs better than traditional empirical models even under small sample conditions. In this case, the results from the Norsok M506 model are relatively conservative compared to other traditional models, which results in larger predicted corrosion rates and thus higher prediction errors [9].

The results of Fig. 13 show that the floating range of the prediction error becomes smaller as the number of combinations increases in the Stacking combination model. This means that if one wants to integrate with the Stacking model, it is not better to have more model combinations, rather a smaller number of combinations is more likely to make the combined prediction error decrease. In most cases of this part of the experiment, the prediction performance after using the stacking combination did not outperform the prediction performance of the underlying ensemble model, indicating that using the Stacking algorithm does not further improve the predictive power of the model. As stated by other researchers [71], improving prediction performance using Stacking models requires that the models involved in the integration are differentiated and have good performance, and these conditions cannot be met in the small sample condition. The results in this section show that although the prediction error values of the bagging + KNN framework are still largely under small sample conditions, it is more suitable for prediction under small sample conditions than the traditional model and the more complex Stacking model.

6 Conclusions

In this paper, a bagging ensemble framework based on knn-based learners is proposed to predict metal corrosion rates under small laboratory sample conditions in the absence of real oil and gas pipeline data. The two most important steps in the experimental preprocessing step are PCA dimensionality reduction and boxplot-based removal of discrete values. These two steps are aimed at optimizing the data set and reducing the influence of extraneous factors and noise on the experiment.

99 data sets are used for training and testing at the ratio of 9: 1. Based on this data set, other models are compared and analyzed. The results show that when the MAE value of the Norsok model is 0.1322, the error of the bagging + KNN framework is only 0.0346, and the error value of this framework is slightly smaller than that of other integrated models, indicating that this framework has certain advantages in the scene. The stacking mode is more complicated but the prediction effect is not ideal, so it is not recommended to use stacking under the condition of small samples. In addition, the effect of only using the individual learner as the prediction model in this experiment is not as bad as imagined, so when the performance of the computing equipment is poor, and the prediction results of the integrated model are not ideal, you may try the individual learner.

The effects of various factors on the experimental results are discussed. The results show that using box-plot to remove discrete values and reduce the dimension moderately (the best result in this experiment is to keep the dimension of 90% information), and increasing the number of training samples can improve prediction performance to some extent. In addition, our research found that the

error value of bagging mode is often smaller than that of Boosting mode in the small sample scenario of this experiment, which indicates that bagging has greater advantages in this scenario.

Although the performance of the proposed framework on this data set is better than that of other models, there is still a large error value on the test set, so the generalization ability of this model is still worth discussing. Under the condition of small samples, slight changes of data may cause a great interference to the results, so this framework only provides an idea to study the corrosion of oil and gas pipelines under the condition of small samples. It is worth noting that the framework removes discrete values and reduces the fluctuation range of the data set, so its generalization ability will be limited when it is used to verify new data sets. Therefore, for future researchers, it is very important to explore the effect of removing outliers in small samples in more detail and how to improve the generalization ability of the model.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 52174062).

Conflicts of Interest: It should be understood that none of the authors have any financial or scientific conflicts of interest concerning the research described in this manuscript.

References

1. Totlani, M. K., Athavale, S. N. (2000). Electroless nickel for corrosion control in chemical, oil and gas industries. *Corrosion Reviews, 18(2–3),* 155–179. DOI 10.1515/CORRREV.2000.18.2-3.155.
2. Xu, Z. D., Zhu, C., Shao, L. W. (2021). Damage identification of pipeline based on ultrasonic guided wave and wavelet denoising. *Journal of Pipeline Systems Engineering & Practice, 12(4),* 1–14. DOI 10.1061/(ASCE)PS.1949-1204.0000600.
3. Guillal, A., Seghier, M. E. A. B., Nourddine, A., Correia, J. A. F. O., Mustaffa, Z. B. et al. (2020). Probabilistic investigation on the reliability assessment of mid-and high-strength pipelines under corrosion and fracture conditions. *Engineering Failure Analysis, 118,* 104891. DOI 10.1016/j.engfailanal.2020.104891.
4. Seghier, M. E. A. B., Keshtegar, B., Taleb-Berrouane, M., Abbassi, R., Trung, N. T. (2021). Advanced intelligence frameworks for predicting maximum pitting corrosion depth in oil and gas pipelines. *Process Safety and Environmental Protection, 147,* 818–833. DOI 10.1016/j.psep.2021.01.008.
5. Pots, B. F. M., John, R. C., Rippon, I. J., Thomas, M. J. J. S., Kapusta, S. D. et al. (2002). Improvements on de waard-milliams corrosion prediction and applications to corrosion management. *NACE-International Corrosion Conference Series*, Denver, Colorado.
6. de Waard, C., Lotz, U., Milliams, D. E. (1991). Predictive model for CO_2 corrosion engineering in wet natural gas pipelines. *CORROSION, 47(12),* 976–985. DOI 10.5006/1.3585212.
7. Hedges, B. (2000). The corrosion inhibitor availability model. *NACE International Annual Conference & Exposition*, Orlando, Florida, USA.
8. Moghissi, O., Burwell, D., Eckert, R., Vera, J., Sridhar, N. et al. (2004). Internal corrosion direct assessment for pipelines carrying wet gas-methodology. *2004 International Pipeline Conference*. DOI 10.1115/IPC2004-0552.
9. Olsen, S. (2003). CO_2 corrosion prediction by use of the norsok M-506 model guidelines and limitations. *CORROSION 2003*, San Diego, California.

10. Lu, H., Iseley, T., Matthews, J., Liao, W. (2021). Hybrid machine learning for pullback force forecasting during horizontal directional drilling. *Automation in Construction, 129,* 103810. DOI 10.1016/j.autcon.2021.103810.
11. Duong, H. T., Phan, H. C., Tran, T. M., Dhar, A. S. (2021). Assessment of critical buckling load of functionally graded plates using artificial neural network modeling. *Neural Computing & Applications, 33(23),* 1–13. DOI 10.1007/s00521-021-06238-61-13.
12. Seghier, M., Corriea, J., Jafari-Asl J., Malekjafarian, A., Trung, N. T. (2021). On the modeling of the annual corrosion rate in main cables of suspension bridges using combined soft computing model and a novel nature-inspired algorithm. *Neural Computing & Applications, 33(23),* 15969–15985. DOI 10.1007/s00521-021-06199-w1-17.
13. Jain, S., Sánchez, A. N., Guan, S., Wu, S., Ayello, F. et al. (2015). Probabilistic assessment of external corrosion rates in buried oil and gas pipelines. *NACE-International Corrosion Conference Series*, Dallas, Texas.
14. Abbas, M. H., Norman, R., Charles, A. (2018). Neural network modelling of high pressure CO_2 corrosion in pipeline steels. *Process Safety & Environmental Protection: Transactions of the Institution of Chemical Engineers Part B, 119,* 36–45. DOI 10.1016/j.psep.2018.07.006.
15. Ossai, C. I. (2020). Corrosion defect modelling of aged pipelines with a feed-forward multi-layer neural network for leak and burst failure estimation. *Engineering Failure Analysis, 110,* 104397. DOI 10.1016/j.engfailanal.2020.104397.
16. Chen, X., Wang, L., Huang, Z. (2020). Principal component analysis based dynamic fuzzy neural network for internal corrosion rate prediction of gas pipelines. *Mathematical Problems in Engineering, 2020(1),* 1–9. DOI 10.1155/2020/3681032.
17. Kishawy, H. A., Gabbar, H. A. (2010). Review of pipeline integrity management practices. *International Journal of Pressure Vessels and Piping, 87(7),* 373–380. DOI 10.1016/j.ijpvp.2010.04.003.
18. Vanaei, H. R., Eslami, A., Egbewande, A. (2017). A review on pipeline corrosion, in-line inspection (ILI), and corrosion growth rate models. *International Journal of Pressure Vessels and Piping, 149(16),* 43–54. DOI 10.1016/j.ijpvp.2016.11.007.
19. Zhang, G., Zeng, Y., Guo, X., Jiang, F., Shi, D. et al. (2012). Electrochemical corrosion behavior of carbon steel under dynamic high pressure H_2S/CO_2 environment. *Corrosion Science, 65,* 37–47. DOI 10.1016/j.corsci.2012.08.007.
20. Zhu, F., Ma, Z., Li, X., Chen, G., Chien, J. et al. (2019). Image-text dual neural network with decision strategy for small-sample image classification. *Neurocomputing, 328,* 182–188. DOI 10.1016/j.neucom.2018.02.099.
21. Angshuman Paul, Y. X. T., Thomas, C. S., Ronald, M. S. (2021). Discriminative ensemble learning for few-shot chest x-ray diagnosis. *Medical Image Analysis, 68,* 101911. DOI 10.1016/j.media.2020.101911.
22. Chen, Y., Zhang, D. (2020). Well log generation via ensemble long short-term memory (EnLSTM) network. *Geophysical Research Letters, 47(23),* 1–9. DOI 10.1029/2020GL087685.
23. Gu, K., Zhang, Y., Qiao, J. (2021). Ensemble meta-learning for few-shot soot density recognition. *IEEE Transactions on Industrial Informatics, 17(3),* 2261–2270. DOI 10.1109/TII.9424.
24. Mahdavi-Shahri, A., Karimian, J., Javadi, A., Houshmand, M. (2018). Multi-Label Classification of Small Samples Using an Ensemble Technique, *Iranian Conference on Electrical Engineering (ICEE),* Mashhad, Iran.
25. Elmousalami, H. H. (2021). Comparison of artificial intelligence techniques for project conceptual cost prediction: A case study and comparative analysis. *IEEE Transactions on Engineering Management, 68(1),* 183–196. DOI 10.1109/TEM.17.
26. Lopez, V., Fernandez, A., Garcia, S. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences, 250,* 113–141. DOI 10.1016/j.ins.2013.07.007.

27. Dvornik, N., Schmid, C., Mairal, J. (2019). Diversity with cooperation: Ensemble methods for few-shot classification. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* pp. 3723–3731.
28. Feng, Y. P., Pang, T. F., Li, M. Q., Guan, Y. Y. (2020). Small sample face recognition based on ensemble deep learning. *2020 Chinese Control And Decision Conference (CCDC)*, IEEE.
29. Breiman, L. (1996). Bagging predictors. *Machine Learning, 24(2),* 123–140. DOI 10.1007/BF00058655.
30. Freund, Y., Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55(1),* 119–139. DOI 10.1006/jcss.1997.1504.
31. Zhang, J., Liu, P., Zhang, F., Iwabuchi, H., de H. e Ayres de Moura, A. A. et al. (2020). Ensemble meteorological cloud classification meets internet of dependable and controllable things. *IEEE Internet of Things Journal, 8(5),* 3323–3330. DOI 10.1109/jiot.2020.30432891.
32. Feng, D. C., Liu, Z. T., Wang, X. D., Chen, Y., Chang, J. Q. et al. (2020). Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Construction & Building Materials, 230,* 117000. DOI 10.1016/j.conbuildmat.2019.117000.
33. Chan, J. C. W. (2008). Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sensing of Environment, 112(6),* 2999–3011. DOI 10.1016/j.rse.2008.02.011.
34. Azhari, M., Abarda, A., Alaoui, A., Ettaki, B., Zerouaoui, J. (2020). Detection of pulsar candidates using bagging method. *Procedia Computer Science, 170,* 1096–1101. DOI 10.1016/j.procs.2020.03.062.
35. Abbes, W., Sellami, D., Marc-Zwecker, S., Zanni-Merk, C. (2021). Fuzzy decision ontology for melanoma diagnosis using KNN classifier. *Multimedia Tools and Applications, 80(17),* 25517–25538. DOI 10.1007/s11042-021-10858-4.
36. Devi, L., Thirumurugan, N. P. (2021). Cervical cancer classification from pap smear images using modified fuzzy C means, PCA, and KNN. *IETE Journal of Research, 67,* 1–8. DOI 10.1080/03772063.2021.1997353.
37. Liao, K. (2020). The effect of acetic acid on the localized corrosion of 3Cr steel in the CO_2-saturated oilfield formation water. *International Journal of Electrochemical Science, 15,* 8622–8637. DOI 10.20964/2020.09.24.
38. Liao, K., Qin, M., He, G., Yang, N., Zhang, S. (2021). Study on corrosion mechanism and the risk of the shale gas gathering pipelines. *Engineering Failure Analysis, 128(5),* 105622. DOI 10.1016/j.engfailanal.2021.105622.
39. Peng, S., Zeng, Z. (2015). An experimental study on the internal corrosion of a subsea multiphase pipeline. *Petroleum, 1(1),* 75–81. DOI 10.1016/j.petlm.2015.04.003.
40. Bendiksen, K., Maines, D., Moe, R. (1991). The dynamic two-fluid model OLGA: Theory and application. *SPE Production Engineering, 6(2),* 171–180. DOI 10.2118/19451-PA.
41. Hansen, L. K., Salamon, P. (1990). Neural network ensembles. pattern analysis and machine intelligence. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(10),* 993–1001. DOI 10.1109/34.58871.
42. Nayak, D. R., Dash, R., Majhi, B. (2016). Brain MR image classification using two-dimensional discrete wavelet transform and adaboost with random forests. *Neurocomputing, 177,* 188–197. DOI 10.1016/j.neucom.2015.11.034.
43. Aflah, W. M., Zubir, M., Aziz, A., Jaafar, J. (2019). Evaluation of machine learning algorithms in predicting CO_2 internal corrosion in oil and gas pipelines. In: *Computational and statistical methods in intelligent systems*, pp. 236–254. Berlin, German: Springer.
44. Wang, G., Hao, J., Ma, J., Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications, 38(1),* 223–230. DOI 10.1016/j.eswa.2010.06.048.
45. Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). *The Annals of Statistics, 26(3),* 801–849. DOI 10.1214/aos/1024691079.

46. Hechenbichler, K., Schliep, K., Chang, C., Lin, C. (2004). Weighted k-nearest-neighbor techniques and ordinal classification. *ACM Transactions on Intelligent Systems and Technology, 2(3),* 1–39. DOI 10.5282/ubm/epub.1769.
47. Peterson, L. (2009). K-nearest neighbor. *Scholarpedia, 4,* 1883. DOI 10.4249/scholarpedia.1883.
48. Aha, D. W., Kibler, D., Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning, 6(1),* 37–66. DOI 10.1007/BF00153759.
49. Chang, C., Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2(3),* 1–39. DOI 10.1145/1961189.1961199.
50. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984). Classification and regression trees. *Journal of the American Statistical Association, 81,* 393. DOI 10.2307/2530946.
51. Deng, H., Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern Recognition, 46(12),* 3483–3489. DOI 10.1016/j.patcog.2013.05.018.
52. Geurts, P., Science, C., Ernst, D., Science, C., Wehenkel, L. et al. (2006). Extremely randomized trees. *Machine Learning, 36(1),* 3–42. DOI 10.1007/s10994-006-6226-1.
53. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29(5),* 1189–1232. DOI 10.1214/aos/1013203451.
54. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W. et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. In: *Advances in neural information processing systems 30 (NIPS 2017)*.
55. Kleiner, B., Graedel, T. E. (1980). Exploratory data analysis in the geophysical sciences. *Reviews of Geophysics, 18(3),* 699–717. DOI 10.1029/RG018i003p00699.
56. Seyedzadeh, S., Rahimian, F. P., Oliver, S., Glesk, I., Kumar, B. (2020). Data driven model improved by multi-objective optimisation for prediction of building energy loads. *Automation in Construction, 116* 103188. DOI 10.1016/j.autcon.2020.103188.
57. Phan, H. C., Duong, H. T. (2021). Predicting burst pressure of defected pipeline with principal component analysis and adaptive neuro fuzzy inference system. *International Journal of Pressure Vessels and Piping, 189*, 104274. DOI 10.1016/j.ijpvp.2020.104274.
58. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, *International Joint Conference on Artificial Intelligence*.
59. Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association, 32(200),* 675–701. DOI 10.1080/01621459.1937.10503522.
60. Lu, H., Xu, Z. D., Iseley, T., Matthews, J. C. (2021). Novel data-driven framework for predicting residual strength of corroded pipelines. *Journal of Pipeline Systems Engineering & Practice, 12(4),* 1–10. DOI 10.1061/(ASCE)PS.1949-1204.0000587.
61. Menahem, E., Rokach, L., Elovici, Y. (2009). Troika–An improved stacking schema for classification tasks. *Information Sciences, 179(24),* 4097–4122. DOI 10.1016/j.ins.2009.08.025.
62. Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system, *22nd ACM SIGKDD International Conference*.
63. Chan, J. C. W., Huang, C., DeFries, R. (2001). Enhanced algorithm performance for land cover classification from remotely sensed data using bagging and boosting. *IEEE Transactions on Geoscience and Remote Sensing, 39(3),* 693–693. DOI 10.1109/36.911126.
64. DeFries, R. S., Chan, J. C. W. (2000). Multiple criteria for evaluating machine learning algorithms for land cover classification from satellite data. *Remote Sensing of Environment, 74(3),* 503–515. DOI 10.1016/S0034-4257(00)00142-5.
65. Briem, G. J., Benediktsson, J. A., Sveinsson, J. R. (2002). Multiple classifiers applied to multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing, 40(10),* 2291–2299. DOI 10.1109/TGRS.2002.802476.

66. Orriols-Puig, A., Casillas, J., Bernadó-Mansilla, E. (2009). Fuzzy-UCS: A michigan-style learning fuzzy-classifier system for supervised learning. *IEEE Transactions on Evolutionary Computation, 13(2),* 260–283. DOI 10.1109/TEVC.2008.925144.
67. Weiss, G. M., Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research, 19,* 315–354. DOI 10.1613/jair.1199.
68. Jo, T., Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter, 6(1),* 40–49. DOI 10.1145/1007730.1007737.
69. Lameiro, C., Schreier, P. J. (2017). A sparse CCA algorithm with application to model-order selection for small sample support. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4721–4725. New Orleans, LA, USA.
70. Windeat, T., Ardeshir, G. (2004). Decision tree simplification for classifier ensembles. *International Journal of Pattern Recognition and Artificial Intelligence, 18(5),* 749–776. DOI 10.1142/S021800140400340X.
71. Džeroski, S., Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning, 54(3),* 255–273. DOI 10.1023/B:MACH.0000015881.36452.6e.