Research Article

Xiuyun Zhai* and Mingtong Chen

Comparison of data-driven prediction methods for comprehensive coke ratio of blast furnace

https://doi.org/10.1515/htmp-2022-0261
received August 27, 2022; accepted December 02, 2022

Abstract: The emission of blast furnace (BF) exhaust gas has been criticized by society. It is momentous to quickly predict the comprehensive coke ratio (CCR) of BF, because CCR is one of the important indicators for evaluating gas emissions, energy consumption, and production stability, and also affects composite economic benefits. In this article, 13 data-driven prediction techniques, including six conventional and seven ensemble methods, are applied to predict CCR. The result of ten-fold cross-validation indicates that multiple linear regression (MLR) and support vector regression (SVR) based on radial basis function are superior to the other methods. The mean absolute error, the root mean square error, and the coefficient of determination (R^2) of the MLR model are 1.079 $kg{\cdot}t^{-1}$, 1.668, and 0.973, respectively. The three indicators of the SVR model are 1.158 $kg{\cdot}t^{-1}$, 1.878, and 0.975, respectively. Furthermore, AdaBoost based on linear regression has also strong prediction ability and generalization performance. The three methods have important significances both in theory and in practice for predicting CCR. Moreover, the models constructed here can provide valuable hints into realizing data-driven control of the BF process.

Keywords: blast furnace, comprehensive coke ratio, multiple linear regression, support vector regression, AdaBoost, data-driven.

1 Introduction

Despite high energy consumption and a large environmental load during the ironmaking process, a blast furnace (BF) is still a crucial component of the whole system of steel production [1–4]. Fundamentally, BF is an input and output system that generates molten iron through a series of extraordinarily complex physical and chemical processes with the cooperation of main and auxiliary materials [5,6]. A schematic diagram of a representative BF is shown in Figure 1. The main materials refer to iron resources, such as iron ore, sinter, and pellets. Auxiliary materials concern energy sources or other necessary materials for transforming main materials, e.g., coke, coal, limestone, and oxygen-enriched water and refractory materials. The production stability of BF is directly influenced by the operation characteristics of main and auxiliary materials [4,7]. The principle of BF ironmaking becomes entangled because of the interaction of various factors [8,9].

Energy consumption, resource shortages, and environmental pollution caused by the development of the iron and steel industry are becoming more and more serious. A lower comprehensive coke ratio (CCR) can decrease coke, heavy oil, and other fuels consumed in producing course of BF and is an effective means to solve the above problems [10]. With the benefit of mathematics and computer technologies to simulate complex BF ironmaking processes, the methods of predicting and controlling various variables by optimizing the BF parameters have recently gained increasing attention [3,11–15]. They contain two approaches: the traditional math approach and machine learning technology. It is a notoriously difficult task to construct a mathematical model on the basis of the mechanism because BF is a complex industrial reactor, including the interacting effects of multiphases accompanied simultaneously by multiphase coupling and multiphysics field coexistence, and so on [14]. In contrast, the machine learning technique based on data-driven is a fast and efficient means that has been widely and successfully employed in many industrial processes since many enterprises have accumulated a large amount of historical data [13,16–19]. Zhang et al. employed a variety of techniques to predict the current time and multistep-ahead hot metal temperature (HMT) of BF, such as random forest (RF), boosting regression tree, neural network–based methods, and Gaussian process regression [13]. Zhai et al. constructed

* **Corresponding author: Xiuyun Zhai,** College of Physics and Electronic Engineering, Center for Magnetic Materials and Devices, Qujing Normal University, Qujing, 655011, Yunnan China; College of Mechanical and Electrical Engineering, Mianyang Normal University, Mianyang, 621000, Sichuan, China, e-mail: cqfbb2008@shu.edu.cn
Mingtong Chen: Public Experimental Teaching Center, Panzhihua University, Panzhihua, 617000, Sichuan China, e-mail: cmt2711@126.com

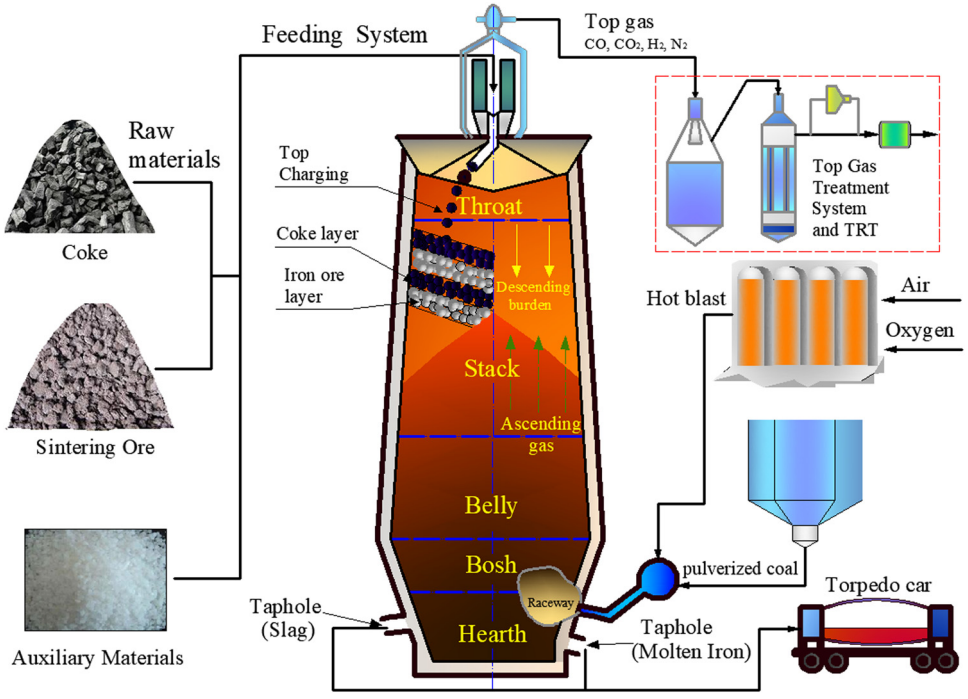

Figure 1: Schematic diagram of BF.

a support vector regression (SVR) model based on the radial basis function (RBF) to predict the BF fuel ratio with merely six parameters [17]. Zhang et al. have demonstrated that the ensemble pattern trees method is superior to several conventional methods for predicting the HMT of BF [14].

Although predictive techniques are emerging in an endless stream, research has not made an appearance on comparing data-driven methods in detail for predicting the CCR of BF. The present work provides a study on predicting CCR with the 13 prediction technologies. They include six conventional methods, namely, multiple linear regression (MLR), decision tree regression (CART), Lasso, elastic net (EN), *k*-nearest neighbor (KNN), and SVR, as well as seven ensemble methods, namely, AdaBoost, RF, AdaBoost based on KNN (KNN-AdaBoost), gradient boosting regression (GBR), AdaBoost based on linear regression (LR-AdaBoost), extremely randomized trees (ERTs), and AdaBoost based on RF (RF-AdaBoost). The results of ten-fold cross-validation and external validation demonstrated that MLR, SVR, and LR-AdaBoost were superior to other methods. The evaluation indicators of the models indicated that they were highly valuable in theory and practice to realize data-driven control of the BF process.

The remaining structure and organization of this article are as follows: The 13 predictive methods are succinctly illustrated in Section 2. In Section 3, the evaluation indicators of the models are explicated. The comparative analysis and application research of the predictive techniques are presented in Section 4. The conclusions of the work are discussed in Section 5.

2 Predictive techniques

In the work, a number of predictive techniques are employed for predicting the CCR of BF, namely, MLR, CART, Lasso, EN, KNN, SVR, AdaBoost, RF, KNN-AdaBoost, GBR, LR-AdaBoost, ERT, and RF-AdaBoost. These methods are briefly depicted as follows.

2.1 MLR

MLR attempts to establish the relationship between a response variable and two or more explanatory variables by fitting a linear equation to the observed data [20]. The value of the response variable y has a relationship with every value of the independent variable x. The population regression line for p explanatory variables x_1, x_2, …, x_p is defined to be $\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$. This line describes how the mean response μ_y changes with the

explanatory variables. The observed values for y vary about their means μ_y and are assumed to have the same standard deviation σ. Formally, the MLR model for n observations can be represented as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i \quad \text{for} \quad i = 1,\ 2,\ldots,\ n, \tag{1}$$

where ε is the notation for the model deviations.

2.2 CART

CART [21,22] is a supervised learning technique that is suitable for both classification and regression problems. It provides a flexible tree-like structure, where internal nodes, branches, and leaf nodes represent the features of a dataset, the decision rules, and the outcomes, respectively. Two kinds of nodes are in a decision tree, i.e., a decision node, including multiple branches for making decisions, and a leaf node for outputting a decision without further branches.

The principle of CART is described as follows. Without losing generality, X and Y are given as the input vector and the response vector, respectively. The training set is defined as $D = \{(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_N, y_N)\}$, where $x_i = (x_i^{(1)},\ x_i^{(2)},\ \ldots,\ x_i^{(n)})$ is known as feature vector, n represents the number of features, $i = 1,\ 2,\ldots,\ N$, N is sample size.

The feature space is divided by the heuristic method. In each division, each value of the features in the current set will be investigated one by one, and the optimal one will be selected as the segmentation point according to the least-squares error criterion. The j-th feature of the training set is denoted as $x^{(j)}$, the value of which is s. $x^{(j)}$ and s are regarded as segmentation variable and segmentation point, respectively. Suppose two regions: $R_1(j, s) = \{x|x^{(j)} \le s\}$ and $R_2(j, s) = \{x|x^{(j)} > s\}$. Solve the following equation to obtain optimal j and s:

$$\min_{j,s}\left[\min_{c1}\sum_{x_i\in R_1(j,s)}(y_i - c_1)^2 + \min_{c2}\sum_{x_i\in R_2(j,s)}(y_i - c_2)^2\right], \tag{2}$$

where c_1 and c_2 denote the fixed output values of the two regions. The above formula can also be written as follows:

$$\min_{j,s}\left[\sum_{x_i\in R_1(j,s)}(y_i - \hat{c}_1)^2 + \sum_{x_i\in R_2(j,s)}(y_i - \hat{c}_2)^2\right], \tag{3}$$

where $\hat{c}_1 = \frac{1}{N_1}\sum_{x_i\in R_1(j,s)} y_i$ and $\hat{c}_1 = \frac{1}{N_2}\sum_{x_i\in R_2(j,s)} y_i$.

After finding the optimal segmentation point (j, s), the input space is divided into two regions. Repeat the above division process for each region until the termination condition is reached. Based on the above process, a least-squares decision tree is constructed.

2.3 Lasso

Lasso regression [23,24], sometimes called L1 regularization of LR, is a modified form of LR. A regularization parameter as punishment is used to control model complexity. It is a biased estimation for dealing with complex collinearity data and obtains a more accurate model by constructing a penalty function. Suppose a regression function:

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n. \tag{4}$$

Its loss function can be expressed as follows:

$$J_L(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2, \tag{5}$$

where $X \in R^{n\times m}$ is the input vector, $Y \in R^m$ is the response vector, n and m are the feature number and the sample size, respectively. The emergence of Lasso is to tackle two problems: over-fitting of LR and the occurrence of X transpose multiplied by irreversible X in course of solving θ by regular methods. To this end, a regularizer is introduced into the loss function, which is as follows:

$$J_L(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2 + \lambda\sum_{j=1}^{n}|\theta_j|, \tag{6}$$

where λ is a regularization parameter. If λ is set too big, it will cause an under-fitting result; otherwise, it will cause over-fitting.

In addition, the loss function can also be matrixed into the following form:

$$J_L(\theta) = \arg\min_{\theta\in R^n} ||y - X\theta||_2^2 + \lambda||\theta||_2^2. \tag{7}$$

2.4 EN

EN, the mixture of Ridge and Lasso regressions, is a linear regression method with L1 and L2 norms as the priori regularizers [25]. A model with only a few non-zero and sparse parameters is learned by this combination, which is just like Lasso. But it still maintains some regular properties like Ridge regression. The matrix of the loss function of EN is described as follows:

$$J_E(\theta) = \arg\min_{\theta \in R^n} ||y - X\theta||^2 + \lambda_2||\theta||_2^2 + \lambda_1||\theta||_1. \quad (8)$$

2.5 KNN

KNN is one of the most basic and simplest machine learning algorithms [26]. Its idea is very simple: Each of the n-dimensional input vectors corresponds to a point in the feature space, and the output is a category label or a prediction value. When it is used for regression prediction, the average of those target values (y_i) of k nearest samples is taken as the prediction value $\hat{y}$ of the new sample [27]. The formula is as follows:

$$\hat{y} = \frac{1}{K}\sum_{i=1}^{K} y_i. \quad (9)$$

2.6 SVR

Based on kernel functions and ε insensitive function, SVR first proposed by Cortes and Vapnik [28] has been applied in many fields successfully [29,30], such as character recognition [31], drug design [32], and combinatorial chemistry [33]. The input and output of the i-th sample ($i = 1, 2, \ldots, l$) are denoted as $x_i \in R^n$ and y_i, respectively. The solution of the nonlinear SVR could be obtained via the following equation:

$$\begin{aligned} \min_{\alpha,\alpha^*} W(\alpha, \alpha^*) = \min_{\alpha,\alpha^*} & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) \\ & + \sum_{i=1}^{l}[(\varepsilon - y_i)\alpha_i + (\varepsilon + y_i)\alpha_i^*], \end{aligned} \quad (10)$$

$$\text{s.t. } 0 \le \alpha_i \le C, \ i = 1, 2, \ldots, l$$

$$0 \le \alpha_i^* \le C, \ i = 1, 2, \ldots, l$$

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0,$$

where α_i and α_i^* represent Lagrange coefficients; C is the penalty factor; and $K(\cdot)$ denotes the kernel function. The regression function $f(x)$ is

$$f(x) = \sum_{i=1}^{N_{SV}} \mathrm{sv}(\alpha_i^* - \alpha_i)K(x_i, x), \quad (11)$$

where N_{SV} denotes the number of SVs.

2.7 AdaBoost

In view of multiple variations of AdaBoost, its R^2 regression is illustrated as an example here [34]. Denoting $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ as the sample set. The iteration number of the weak learner is K.

The initial weight of the sample set is expressed as follows:

$$D(1) = (\omega_{11}, \omega_{12}, \ldots, \omega_{1m}); \quad \omega_{1i} = \frac{1}{m}; \quad i = 1, 2, \ldots, m.$$

Supporting $k = 1, 2, \ldots, K$:

(1) Gain a weak learner $G_k(x)$ by training the sample set with weights D_k.

(2) Calculate the maximum error of the training set.

$$E_k = \max|y_i - G_k(x_i)|, \quad i = 1, 2, \ldots, m. \quad (12)$$

(3) Calculate the relative error for each sample. Exponent error is expressed as follows:

$$e_{ki} = 1 - \exp\left(\frac{-|y_i - G_k(x_i)|}{E_k}\right). \quad (13)$$

(4) Calculate the regression error rate as follows:

$$e_k = \sum_{i=1}^{m}\omega_{ki}e_{ki}. \quad (14)$$

(5) Calculate the coefficient of the weak learner as follows:

$$\alpha_k = \frac{e_k}{1 - e_k}. \quad (15)$$

(6) Update the weight distribution of the sample set as follows:

$$\omega_{k+1,i} = \frac{\omega_{ki}}{Z_k}\alpha_k^{1-e_{ki}}. \quad (16)$$

where the normalization factor Z_k is expressed as follows:

$$Z_k = \sum_{i=1}^{m}\omega_{ki}\alpha_k^{1-e_{ki}}. \quad (17)$$

Finally, the strong learner is constructed as follows:

$$f(x) = \left[\sum_{k=1}^{K}\left(\ln\frac{1}{\alpha_k}\right)\right]g(x), \quad (18)$$

where $g(x)$, $k = 1, \ldots, K$, is the median value of all $\alpha_k G_k(x)$.

2.8 RF

RF [35] is an ensemble of B trees $\{T_1(X), \ldots, T_B(X)\}$, where $X = \{x_1, \ldots, x_p\}$ is a p-dimensional input vector [36]. The ensemble produces B outputs $\{\hat{Y}_1 = T_1(X), \ldots, \hat{Y}_B = T_B(X)\}$, where $\hat{Y}_b$, $b = 1, \ldots, B$, is a prediction value of the output variable generated by the b-th tree. The outputs of all

trees are aggregated to generate one final prediction, $\hat{Y}$. In regression, it is the average prediction of all the individual trees.

The training set, $D = \{(X_1, Y_1),\ldots,(X_n, Y_n)\}$, where X_i, $i = 1,\ldots, n$, is an input vector and Y_i is an output vector. The training process is as follows:

(1) A bootstrap sample with replacement is randomly drawn from the training data including n samples.
(2) For each bootstrap sample, a tree will be grown by the following process: at each node, the optimal segmentation is chosen among the subsets, including random m_{try} (rather than all) variables. Here, m_{try} is the only tunable parameter in RF. The tree is grown to its maximum size (i.e., until no further splits are possible) and not pruned back.
(3) Repeat the above steps until B (a sufficiently large number) trees are grown.
(4) Average the predictions of the individual trees to obtain the final prediction as follows:

$$\hat{Y} = \frac{1}{B}\sum_{b=1}^{B}\widehat{Y_b}. \tag{19}$$

2.9 KNN-AdaBoost, LR-AdaBoost, and RF-AdaBoost

Ensemble learning algorithms can significantly promote the generalization ability of learning systems by training a certain number of weak learners and combining them into a strong learner. KNN-AdaBoost, LR-AdaBoost, and RF-AdaBoost are all ensemble algorithms that improve AdaBoosts by training KNN, LR, or RF as weak learners, respectively [37–39]. The training process is the same as AdaBoost, and only one difference can be found in the first step. KNN, RF, or LR algorithms are selected as the weak learners $G_k(x)$, and the sample set is trained with the weights D_k in iterations.

2.10 GBR

GBR fits the negative gradient by iterative process [40,41]. Considering a regression question, n samples, (x_1, y_1), (x_2, y_2), …, (x_n, y_n) given are fitted into a function $F(x)$ with minimal error.

Set the fitted values $\hat{y} = F(x)$, square error function $L(y, \hat{y}) = (y - \hat{y})^2/2$, and the total error for all samples $J = \sum_{i=1}^{n} L(y_i, \hat{y}_i)$.

The target value y_i of the sample is known, and the final result of error J is a numerical scalar. Error J is a function of the n-variable $(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$ for a total of n samples.

Calculate the gradient of the function J as follows:

$$\nabla J = \left(\frac{\partial J}{\partial \hat{y}_1} \cdots \frac{\partial J}{\partial \hat{y}_i} \cdots \frac{\partial J}{\partial \hat{y}_n}\right), \tag{20}$$

$$\frac{\partial J}{\partial \hat{y}_i} = \frac{\partial \sum_{i=1}^{n} L(y_i, \hat{y}_i)}{\partial \hat{y}_i} = \frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i} = \frac{\partial((y_i - \hat{y}_i)^2/2)}{\partial \hat{y}_i}$$
$$= -(y_i - \hat{y}_i),$$

$$y_i - \hat{y}_i = -\frac{\partial J}{\partial \hat{y}_i}. \tag{21}$$

It can be seen from the above derivation that the residual error is equal to the negative gradient $y - \hat{y} = -\nabla J$.

The weak models are established through iteration, gradually enhanced (boosting) and combined into a strong model.

2.11 ERT

ERT, proposed by Geurts et al. [42] in 2006, is very similar to the RF algorithm, which is composed of many decision trees [43,44]. The two major differences between the two algorithms are described as follows:

(1) The bagging model is applied in the RF algorithm. While all samples are trained to generate each decision tree in ERT, i.e., each decision tree is constructed with the same training samples.
(2) A random subset of RF needs to find the best bifurcation attribute. While the bifurcation value of ERT is obtained completely at random, so as to realize the bifurcation of a decision tree. The ERT model is trained with random features and threshold values. Therefore, the training process of ERT is much faster than RF.

3 Evaluation criteria

The four evaluation criteria, including root mean square error (RMSE), correlation coefficient (R), mean absolute error (MAE), and R^2 (coefficient of determination), were employed to evaluate the results of TFCV, model prediction, and external validation. They are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{22}$$

$$R = \frac{\sum_{i=1}^{N}(y_i - m_0)(\hat{y}_i - \hat{m}_0)}{\sqrt{\sum_{i=1}^{N}(y_i - m_0)^2}\sqrt{\sum_{i=1}^{N}(\hat{y}_i - \hat{m}_0)^2}}, \quad (23)$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \quad (24)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - m_0)^2}, \quad (25)$$

where N, y_i, $\hat{y}_i$, m_0, and $\hat{m}_0$ denote the number of samples, the measured value, the predicted value, the mean value of y_i, and the mean value of $\hat{y}_i$, respectively.

4 Results and discussions

In this section, the genetic algorithm (GA) and recursive feature elimination (RFE) are adopted to select variables to form the optimal feature sets. Thirteen technologies, including MLR, CART, Lasso, EN, KNN, SVR, AdaBoost, RF, KNN-AdaBoost, GBR, LR-AdaBoost, ERT, and RF-AdaBoost, were used for predicting the CCR of the industrial BF. The comparative analysis of the above techniques is presented as follows.

4.1 Dataset

The dataset consists of two parts: the training set for modeling and the test set for validation. After excluding the outliers caused by the abnormal state of BF (such as blowing down, record fault, or overhaul) and data with missing values, the training set with 326 samples was established by collecting the historical data throughout a year from an industrial BF (an internal volume of 2,000 m^3) with a sampling interval of 1 day. The test set includes 87 samples collected from the same BF in the first 3 months of next year. The disciplinarian of BF burden distribution remained the same during the period of data collection [17].

The dependent variable of the dataset is CCR ($kg{\cdot}t^{-1}$). The data for every CCR is the average of a day. According to the experience of BF experts, 36 process variables (shown in Table 1) were chosen as the independent variables of the dataset. The quantity and quality of data are two key factors influencing industrial optimization and constructing a reasonable model. A common thought is that the amount of data needed to construct a plausible data-driven model is at least three times of feature number. The quantity of data in the work meets the above requirement. The quality of the data depends on the spatial coverage of the target variable and the uncertainties associated with the data [45]. Normally, data with a normal distribution is in favor of establishing a reasonable model. Data uncertainty, such as experimental error and input error, could influence the quality of the data. To further analyze the quality of the data, the distribution curves of CCR and the independent variables were plotted in Figures S1 and S2 of Supporting Information, respectively. It can be seen from the two figures that all variables are basically consistent with normal distribution,

Table 1: The list of the independent variables

No.	Meanings	Features	No.	Meanings	Features
1	Grade of iron (%)	GI	19	Iron losses (%)	LI
2	Pig iron [Ti] (%)	PI_{Ti}	20	Unit consumption of nut coke ($kg{\cdot}t^{-1}$)	Ic
3	Pig iron [Si] (%)	PI_{Si}	21	Small sinter ($kg{\cdot}t^{-1}$)	SS
4	Blast tmperature (°C)	T_B	22	Comprehensive ironmaking strength ($t/m^3{\cdot}d$)	SC
5	Top gas pressure (MPa)	P_{Top}	23	Feed batch	FB
6	Blast volume ($m^3{\cdot}min^{-1}$)	Q_B	24	Gas utilization rate (%)	R_{Gas}
7	Burden ratio ($t{\cdot}t^{-1}$)	P	25	Unit consumption of iron ore ($kg{\cdot}t^{-1}$)	U_{IO}
8	Utilization coefficient ($t{\cdot}m^{-3}{\cdot}day^{-1}$)	η_v	26	Permeability index	IP
9	Slag ratio ($kg{\cdot}t^{-1}$)	R_{Sl}	27	Gray iron ratio (%)	R_{GI}
10	Basicity of slag (%)	R_{Bas}	28	Top temperature (°C)	T_{Top}
11	Oxygen-enriched rate (%)	R_{Ox}	29	Sinter of each batch ($t{\cdot}batch^{-1}$)	S_B
12	Coke ash (%)	CA	30	Small sinter of each batch ($t{\cdot}batch^{-1}$)	SS_B
13	Coke sulfur (%)	CS	31	Pellet 1 of each batch ($t{\cdot}batch^{-1}$)	P_{1B}
14	Coke M40 (%)	M_{40}	32	Pellet 2 of each batch ($t{\cdot}batch^{-1}$)	P_{2B}
15	Coke M10 (%)	M_{10}	33	Huili mine of each batch ($t{\cdot}batch^{-1}$)	M_B
16	Coke CSR (%)	CSR	34	Batch weight of coke ($t{\cdot}batch^{-1}$)	C_{BW}
17	Coke <25 mm (%)	C_{25}	35	Blast speed ($m{\cdot}s^{-1}$)	V_B
18	Clinker rate (%)	R_{CL}	36	Coal injection ratio ($kg{\cdot}t^{-1}$)	Y_{Coal}

and the stability of the data is higher because the data show relative concentration. Most of the data measurement instruments have random and systematic errors, so the data collected from the BF system has some degree of uncertainty. On the whole, the quality of the data set used is suitable for data mining and industrial optimization, although the BF data have some inevitable errors [17,19].

4.2 Selecting variables

Before establishing the models, variable selection can not only reduce the dimension of feature space to further decrease the risk of over fitting but also better remove some variables unrelated to the target variable and noise interference. Meanwhile, it can also greatly reduce training time and further improve the prediction accuracy and generalization performance of the model [17,46]. Apparently, the model's performance will be harshly affected, and the model is too complex if the variables irrelevant to the output are not deleted before training a model.

In the work, the primary feature screening was conducted using the Pearson correlation coefficients between the features. First, the max relevance min redundancy (mRMR) method [47] was employed to sort the features. The sorting result is represented in Figure S3. Second, the Pearson correlation coefficients between the features were calculated. Its matrix is shown in Figure S4. If the correlation score between the features is higher than 0.9, the feature with the lower mRMR score will be omitted. After PI_{Si}, R_{CL}, and SS_B were omitted, the 33 features were retained by the primary screening.

GA based on the learners and RFE based on the base_estimators were used to further optimize the feature sets, respectively. GA, first put forward by Holland [48], is a global optimization algorithm of random search that simulates the processes of inheritance and evolution in natural conditions. It has highly collateral, random, and adaptive features, is an effective method to remove from local optima present on the response surface, and can solve a wide variety of optimization problems with the requirement of no knowledge about the response surface or gradient present on it [49].

RFE is a method that works by selecting data features recursively based on the smallest feature value [50]. In the RFE concept, RFE based on the base_estimators works by eliminating irrelevant features in each iteration, namely the lowest-weight feature. This method is divided into three stages as described in ref. [51]. The two feature selection processes are related to the learners or the base_estimators and are automatically performed during the training of the learners or the base_estimators.

Figure 2 illustrates the implementation processes of variable selection of GA based on the seven learners, respectively. The learners employed are CART, KNN, SVR, KNN-AdaBoost, GBR, LR-AdaBoost, and RF-AdaBoost. Genetic algebra and the best feature set for each GA-learner are listed in Table 2. Figure 3 shows the processes of feature selection of RFE based on the six base_estimators, namely, MLR, Lasso, EN, AdaBoost, RF, and ETR. Table 3 illustrates the best feature set for each RFE – base_estimator. In the variable screening process, the optimal feature sets will be found when R is the maximum or RMSE is the minimum. It can be seen from Tables 2 and 3 that the best feature set is different with different learners or base_estimators, and the feature number for different optimal feature set varies greatly. The least feature number is 4, and the most is 18.

The top three most frequency features selected by the various GA-learners are P, U_{IO}, and Y_{Coal} in the feature subsets. P refers to the amount of ore smelted by unit coke, whose higher value may result in lower CCR. U_{IO} is the amount of iron ore consumption for generating 1 t of pig iron, and the larger U_{IO} content may lead to the higher CCR. In terms of the mechanism of BF operation, Y_{Coal} has large contributions to CCR and their relationship is related to the quality and dosage of pulverized coal and coke.

As shown in Figure 4, the scatter plots between the top important features and CCR are generated for further analysis based on the training data. Figure 4(a) indicates that the larger P tends to result in a smaller CCR value and has a negative correlation with CCR. Meanwhile, Figure 4(b) shows the positive correlation between U_{IO} and CCR, and Figure 4(c) reveals the positive one. Therefore, the relationships between the three features and CCR are consistent with the mechanism of BF operation.

4.3 CCR prediction

The parameters of the models were optimized with the grid search technique and TFCV. The results were compared by three evaluation criteria, namely RMSECV, MAE_CV, and R^2_CV, where "CV" represents "cross validation." The three methods with the top performances were discovered by TFCV and used to establish the models. The generalization capabilities and robustness of the models were evaluated by using RMSE, MAE, R, and R^2.

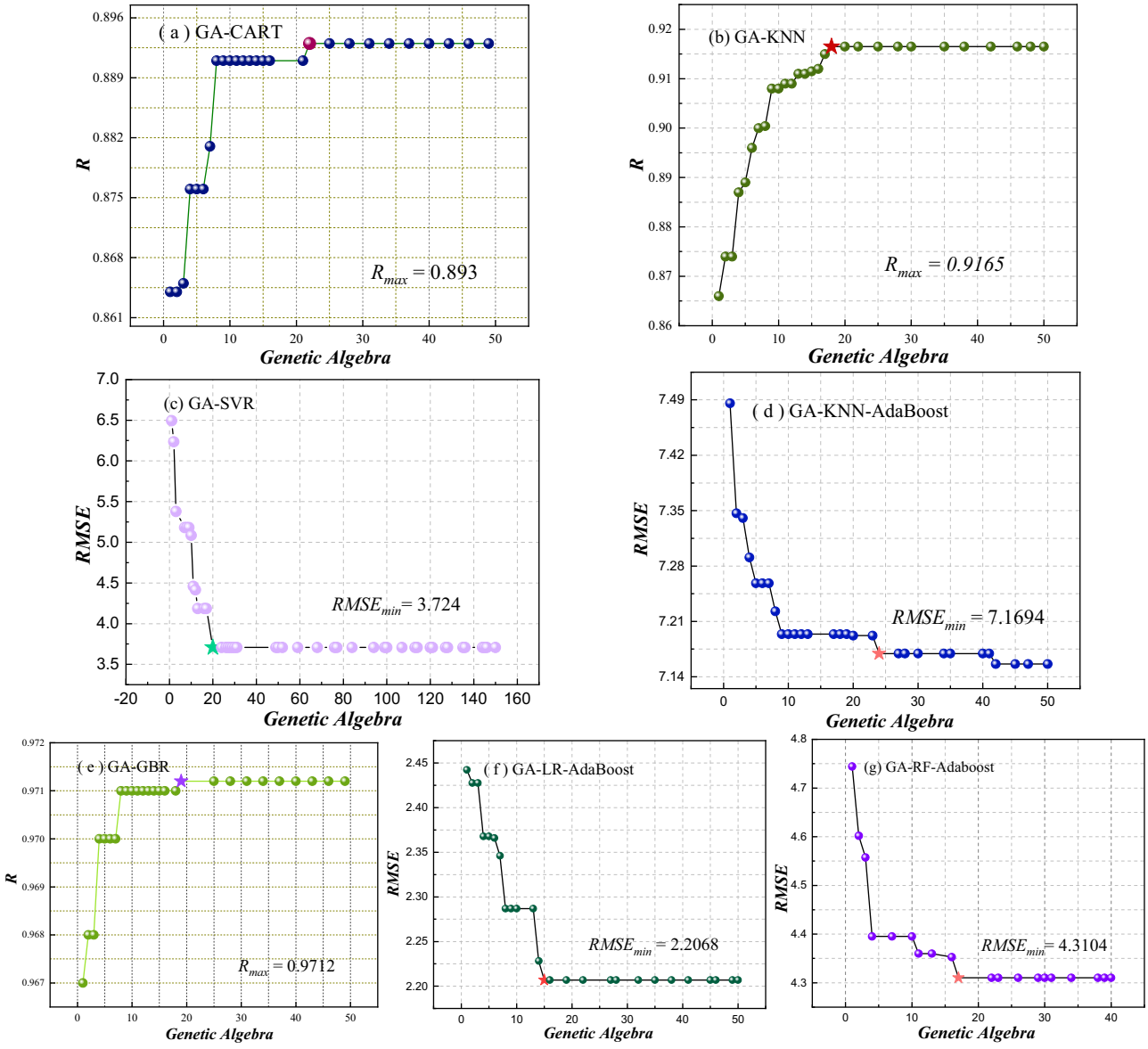

Figure 2: The procedures of variable selection in GA-learners: (a) GA-CART, (b) GA-KNN, (c) GA-SVR, (d) GA-KNN-AdaBoost, (e) GA-GBR, (f) GA-LR-AdaBoost, and (g) GA-RF-AdaBoost.

Table 2: Genetic algebra and the optimal feature sets in GA-learners

Learner	Genetic algebra	Selected features
CART	22	GI, T_B, P_{Top}, P, CA, C_{25}, SC, FB, U_{IO}, S_B, P_{2B}, M_B, Y_{Coal}
KNN	18	P, η_v, R_{SI}, CSR, LI, Ic, SC, FB, R_{Gas}, U_{IO}, IP, Y_{Coal}
SVR	20	P, M_{10}, U_{IO}, Y_{Coal}
KNN-AdaBoost	24	GI, P, η_v, R_{SI}, R_{Bas}, R_{Ox}, CA, CS, M_{40}, CSR, LI, Ic, FB, R_{Gas}, U_{IO}, S_B, P_{1B}, Y_{Coal}
GBR	19	PI_{Ti}, P_{Top}, P, R_{Ox}, CS, M_{40}, M_{10}, CSR, C_{25}, Ic, SS, SC, FB, R_{Gas}, U_{IO}, IP, T_{Top}, M_B, C_{BW}, Y_{Coal}
LR-AdaBoost	15	GI, PI_{Ti}, P, R_{SI}, U_{IO}, R_{GI}, T_{Top}, S_B, P_{1B}, P_{2B}, Y_{Coal}
RF-AdaBoost	17	P_{Top}, P, η_v, R_{SI}, M_{40}, SC, U_{IO}, P_{2B}, Y_{Coal}

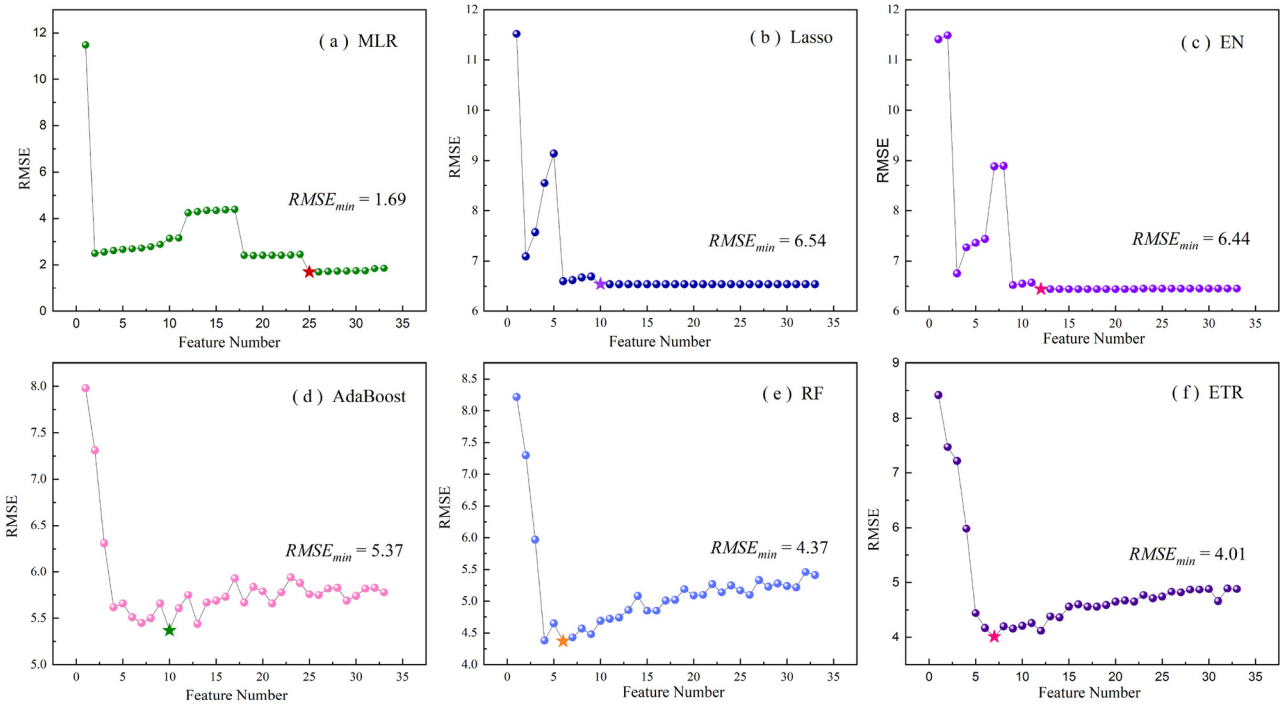

Figure 3: The procedures of feature selection in RFE based on the base_estimators: (a) MLR, (b) Lasso, (c) EN, (d) AdaBoost, (e) RF, and (f) ETR.

Table 3: The optimal feature sets in RFE based on the base_estimators

Base_estimator	Selected features
MLR	PI_{Ti}, P_{Top}, R_{Ox}, M_{10}, C_{25}, Ic, SS, SC, FB, R_{Gas}, U_{IO}, S_B, P_{1B}, P_{2B}, M_B, C_{BW}, Y_{Coal}
Lasso	Q_B, CSR, Ic, FB, U_{IO}, IP, R_{GI}, T_{Top}, V_B, Y_{Coal}
EN	Q_B, CSR, Ic, FB, R_{Gas}, U_{IO}, IP, R_{GI}, T_{Top}, C_{BW}, V_B, Y_{Coal}
AdaBoost	PI_{Ti}, P, η_v, R_{SI}, LI, Ic, SC, U_{IO}, C_{BW}, Y_{Coal}
RF	PI_{Ti}, P, η_v, LI, SC, U_{IO}, Y_{Coal}
ETR	P, η_v, LI, SC, U_{IO}, C_{BW}, Y_{Coal}

4.3.1 Model validation

The TFCV results of six conventional and seven ensemble methods constructed with the optimal features obtained from feature selection are shown in Table 4. Compared with the other methods, KNN-AdaBoost has an inferior prediction performance, for it achieved the largest RMSECV and MAE_CV and the lowest R^2_CV in TFCV. MLR, SVR based on RBF, and LR-AdaBoost exhibit the best predictive properties among all the technologies. LR-AdaBoost is an ensemble learning algorithm that improves AdaBoost with linear regression as weak learner. According to the excellent performance of the MLR and LR-AdaBoost models, a conclusion can be drawn that a certain linear relationship exists between the target variable and the independent variables. Due to sufficiently high R^2_CV and very low RMSECV and MAE_CV, the three models have excellent generalization performance. Therefore, the further optimization of other methods would not be considered.

Figure 5 reveals the box plots of RMSECV and R^2_CV for six conventional methods. In Figure 5(a), the technologies are ranked from low to high according to the center of RMSECV: MLR, SVR, Lasso, KNN, EN, and CART. Especially, the center values of RMSECV distribution of the MLR and SVR models are far less than other methods. As shown in Figure 5(b), the zones of R^2_CV for MLR, SVR, and Lasso are much narrower than those of the other methods, and their center values are much higher than those of the others. According to the above analysis, MLR, SVR, and Lasso are more reliable and more precise for predicting CCR than the other techniques.

Figure 6 represents the box plots of RMSECV and R^2_CV of TFCV for seven ensemble methods. In Figure 6(a), the methods are ordered according to the central values of RMSECV in ascending order: LR-AdaBoost, GBR, RF, ERT, RF-AdaBoost, AdaBoost, and KNN-AdaBoost. The RMSECV range of LR-AdaBoost is the narrowest among the seven methods. In Figure 6(b), the

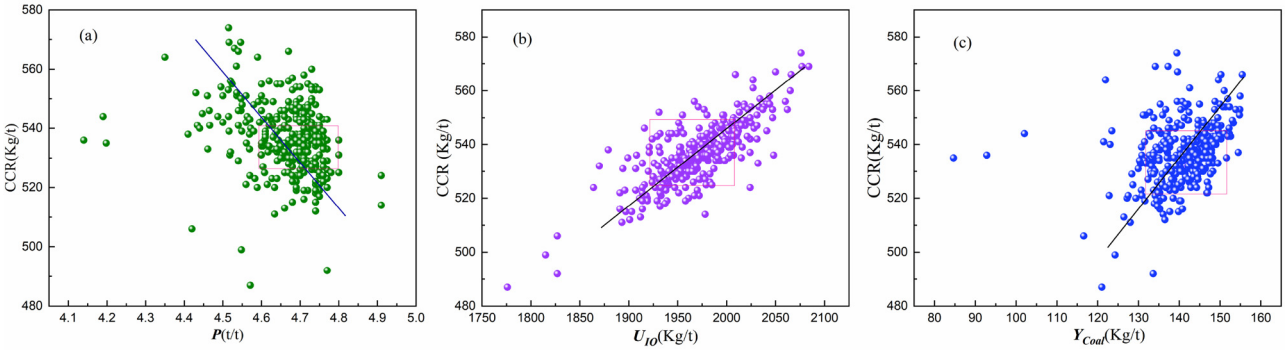

Figure 4: Relationships between the top three most important features: (a) P, (b) U_{IO}, and (c) Y_{Coal} and CCR.

Table 4: The TFCV results of the thirteen methods

Methods	RMSECV	MAE_CV	R^2_CV
MLR	1.668	1.079	0.973
CART	5.630	3.860	0.757
Lasso	3.018	2.365	0.928
EN	4.776	3.635	0.819
KNN	4.233	2.890	0.882
SVR	1.878	1.158	0.975
AdaBoost	5.073	3.808	0.803
RF	3.711	2.741	0.885
KNN-AdaBoost	6.404	4.882	0.674
GBR	3.459	2.578	0.904
LR-AdaBoost	2.276	1.638	0.953
ERT	4.053	2.976	0.867
RF-AdaBoost	4.226	2.931	0.858

centers of LR-AdaBoost, RF, GBR, ERT, and RF-AdaBoost are much higher than those of the others. The methods with a narrow distribution of R^2_CV include LR-AdaBoost, RF, and ERT. On the whole, LR-AdaBoost is the most accurate and stable technique for predicting CCR among all the ensemble techniques.

In addition, it can be seen from the above two figures that MLR has the superior prediction performance, while SVR and LR-AdaBoost trail in second and third, respectively.

4.3.2 Model training and testing

Based on the above comparison and analysis of the thirteen methods in TFCV, MLR, SVR, and LR-AdaBoost have the top three performances for CCR prediction. This section will illustrate how they would be employed to establish the models and how well the models performed in external validation.

By setting a regression equation, the MLR model can estimate the dependent variable with multiple independent

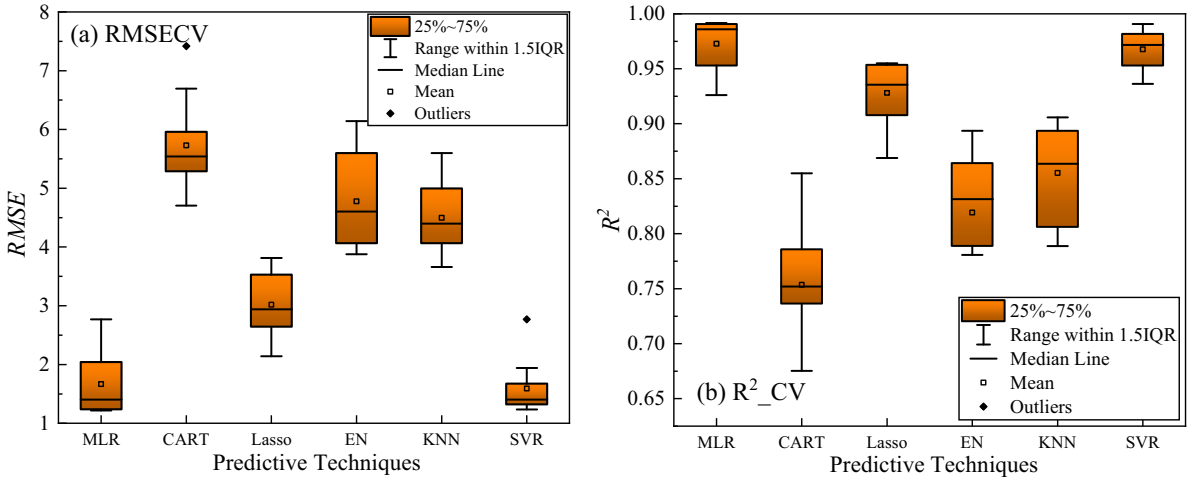

Figure 5: Box plots of (a) RMSECV and (b) R^2_CV for six conventional methods.

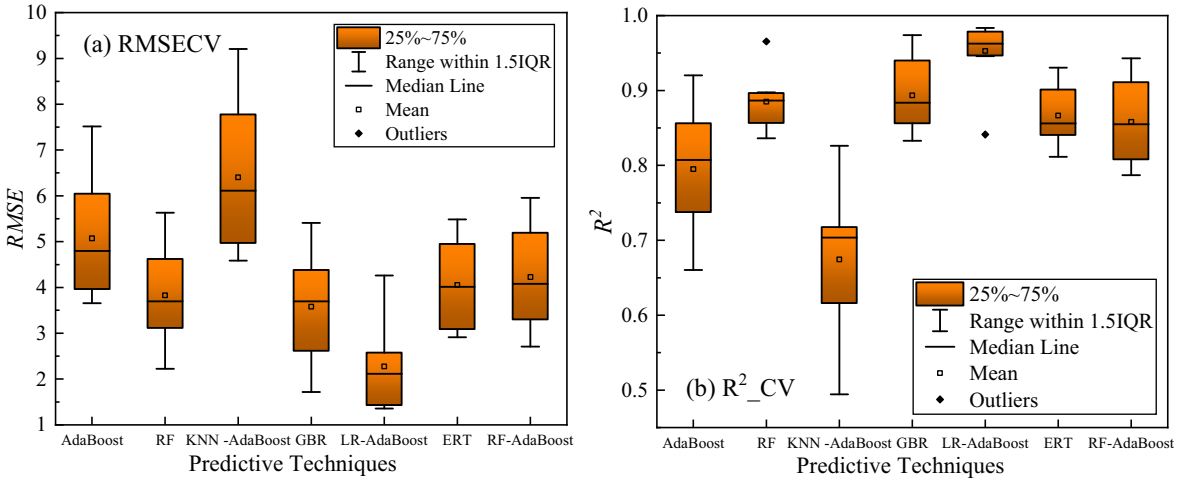

Figure 6: Box plots of (a) RMSECV and (b) R^2_CV for seven ensemble methods.

variables and then predict the dependent variable. The independent variables were standardized because of their different units. The larger the value of the standardized regression coefficient, the greater the influence of the independent variable on the dependent variable. The MLR model constructed with the optimal feature set is shown as follows:

$$\begin{aligned} \mathrm{CCR} = &-4.16\mathrm{PI_{Ti}} + 9.05P_{\mathrm{Top}} - 1.64R_{\mathrm{Ox}} + 0.82M_{10} \\ &- 0.19C_{25} + 0.24\mathrm{Ic} - 0.05\mathrm{SS} + 287.4\mathrm{SC} \\ &- 1.88\mathrm{FB} - 0.41R_{\mathrm{Gas}} + 0.25U_{\mathrm{IO}} - 9.05S_{\mathrm{B}} \\ &- 8.76P_{1\mathrm{B}} - 9.5P_{2\mathrm{B}} - 9.61M_{\mathrm{B}} + 9.25C_{\mathrm{BW}} \\ &+ 0.21Y_{\mathrm{Coal}} + 348.52. \end{aligned} \quad (26)$$

Parameter tuning of SVR and LR-AdaBoost is a crucial step to develop models with high generalization performance. The three hyperparameters, C, ε, and σ need to be optimized using the grid search method based on leave-one-out cross-validation before SVR modeling [17]. In the work, the range of ε was from 0.01 to 0.1 with a step of 0.01; C changed from 1 to 100 with an interval of 2; σ varied from 0.5 to 1.4 with a step of 0.1. As can be seen from Figure 7, the higher the C and σ, the larger the RMSE. The RMSE value ranges from 1.867 to 5.470. Obviously, the reasonable selection of C, ε, and σ has a great influence on the prediction performance of the SVR model. After conducting the grid search shown in Figure 7, the optimal C, ε, and σ were determined to be 23, 0.01, and 0.7, respectively, when RMSE (equal to 1.867) was lowest.

The SVR model constructed with the optimal C, ε, and σ is shown as follows:

$$y = \sum_{i=1}^{n} \beta_i \cdot \exp(-0.7 \cdot ||x - x_i||^2) + 0.64286, \quad (27)$$

where x is the unknown vector, x_i is the support vector of the SVR model, n is the corresponding sample number, and β_i is the Lagrange multiplier of the support vector.

The two hyperparameters of the LR-AdaBoost model, namely n_estimators and learning_rate need to be tuned by using the grid search method. Here, n_estimators is the number of weak learners. Generally, too small n_estimators could cause under-fitting, otherwise could cause over-fitting. Furthermore, learning_rate is the weight reduction factor of the weak learner. In this work, n_estimators varied from 1 to 10 with a step of 1 and from 20 to 100 with a step of 10. Learning_rate changed from 0.1 to 1 with an interval of 0.1. The result indicated that n_estimators and learning_rate were equal to 2 and 0.1, respectively, when RMSE was at a minimum. The optimization process of the

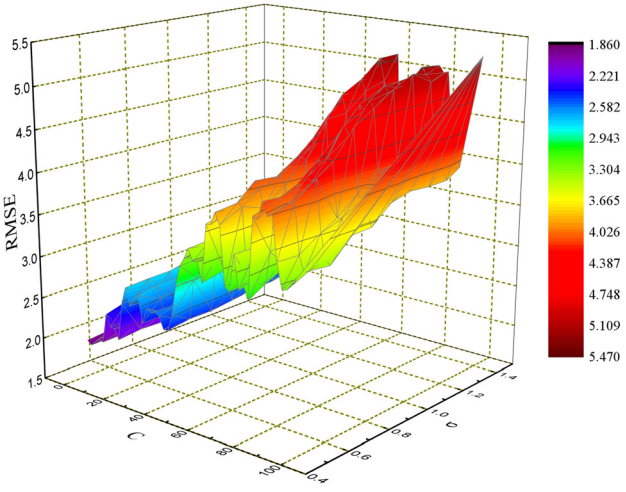

Figure 7: The variety of RMSE with σ and C in optimizing SVR's hyperparameters.

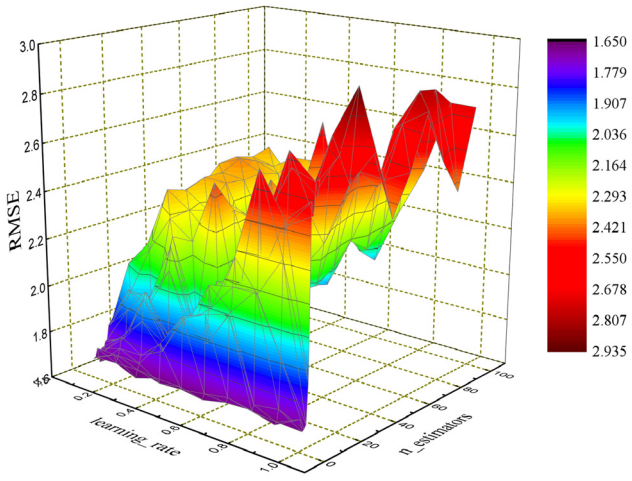

Figure 8: The variety of RMSE with learning_rate and *n*_estimators in optimizing LR-AdaBoost's hyperparameters.

Table 5: The prediction results of the MLR, SVR, and LR-AdaBoost models

Indicator	MLR	SVR	LR-AdaBoost
$RMSE_{TR}$	1.337	1.812	1.855
MAE_{TR}	1.037	1.084	1.123
R_{TR}	0.993	0.988	0.986
R^2_{TR}	0.987	0.976	0.973
$RMSE_{TE}$	1.593	2.464	1.863
MAE_{TE}	1.328	1.887	1.508
R_{TE}	0.989	0.981	0.987

parameters is shown in Figure 8, from which it can be found that *n*_estimators have a more influential impact on RMSE than learning_rate. When *n*_estimators are less than 20, a smaller RMSE can be obtained.

The MLR, SVR, and LR-AdaBoost models were employed to predict the samples in the test set. The evaluation criteria of the three models, RMSE, MAE, *R*, and R^2 are listed in Table 5. The subscripts "TR" and "TE" represent "training" and "test," respectively. It can be seen from the table that MLR has the lowest $RMSE_{TR}$ and MAE_{TR}, and the highest R_{TR} and R^2_{TR}. Moreover, it also performs best in the external test. In model training, SVR is superior to LR-AdaBoost. The reverse is true for the performance indicators generated by the external test of the two technologies. This shows that LR-AdaBoost has stronger extrapolation performance than SVR. Figure 9 represents the scatter plots of the three models. It can be seen from the figure that they could primely fit the experimental data since the predictive data are closely distributed along the diagonal direction. In particular, the training result of SVR is slightly better than LR-AdaBoost, but LR-AdaBoost in external validation is more reliable than SVR. It can be seen from the above analysis that the three methods all possess high prediction accuracy and practical values.

5 Conclusions

In the work, a comparative study of data-driven prediction methods for the CCR of BF was carried out. Six conventional methods, including MLR, CART, Lasso, EN, KNN, and SVR, and seven ensemble methods, namely, AdaBoost, RF, KNN-AdaBoost, GBR, LR-AdaBoost, ERT, and RF-AdaBoost, were investigated from the application point of view. The TFCV results found that KNN-AdaBoost had the lowest competitiveness for predicting CCR. Meanwhile, MLR had the optimal prediction performance among all the methods, while SVR and LR-AdaBoost trailed in second and third place. Furthermore, the SVR model possessed better training performance than the LR-AdaBoost model, but the LR-AdaBoost method appeared more reliable for external validation. On the whole, the three methods owned high practical values and generalization performance

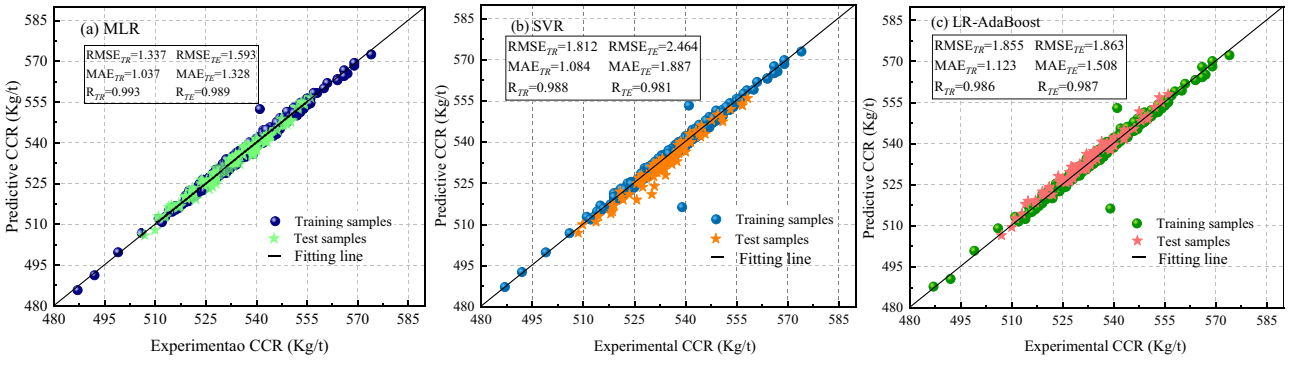

Figure 9: The scatter plots of the (a) MLR, (b) SVR, and (c) LR-AdaBoost models.

because of their models' extremely high R and very low RMSE and MAE. The study integrated 13 techniques to solve the industrial optimization problem with strong noise and multivariable couple. The method outlined here can provide valuable hints into industry optimization with the assistance of machine learning and has important instructions and practical value for assisting managers to control BF parameters and detect BF status.

Acknowledgments: The authors gratefully acknowledge the financial supports from the Sichuan Science and Technology Program (No.2022YFG0318) and Sichuan Technology & Engineering Research Program for Vanadium Titanium Materials of China (No. 2020-2FTGC-YB-01).

Funding information: The work was supported by the Sichuan Science and Technology Program (No. 2022YFG0318) and Sichuan Technology & Engineering Research Program for Vanadium Titanium Materials of China (No. 2020-2FTGC-YB-01).

Author contributions: X.Z. contributed to the writing of the original draft, review, and editing and assisted with the methodology, formal analysis, validation, resources, and funding acquisition. M.C. contributed to the writing of the original draft and assisted with the formal analysis, visualization, project administration, and investigation.

Conflict of interest: The authors state that there is no conflict of interest.

Data availability statement: All data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Kuang, S., Z. Li, and A. Yu. Review on modeling and simulation of blast furnace. *Steel Research International*, Vol. 89, No. 1, 2017, id. 1700071.

[2] Li, J. P., C. C. Hua, Y. N. Yang, and X. P. Guan. Fuzzy classifier design for development tendency of hot metal silicon content in blast furnace. *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 3, 2018, pp. 1115–1123.

[3] Roche, M., M. Helle, J. van der Stel, G. Louwerse, L. Shao, and H. Saxen. On-line estimation of liquid levels in the blast furnace hearth. *Steel Research International*, Vol. 90, No. 3, 2019, id. 1800420.

[4] Nielson, S., T. Okosun, B. Damstedt, M. Jampani, and C. Q. Zhou. Tuyere-level syngas injection in the blast furnace: a computational fluid dynamics investigation. *Processes*, Vol. 9, No. 8, 2021, id. 1447.

[5] Li, Z. N., M. S. Chu, Z. G. Liu, G. J. Ruan, and B. F. Li. Furnace heat prediction and control model and its application to large blast furnace. *High Temperature Materials and Processes*, Vol. 38, 2019, pp. 884–891.

[6] Dong, X. F., P. Zulli, and M. Biasutti. Prediction of blast furnace hearth condition: Part II – A transient state simulation of hearth condition during blast furnace shutdown. *Ironmaking & Steelmaking*, Vol. 47, No. 5, 2020, pp. 561–566.

[7] Bernasowski, M., A. Klimczyk, and R. Stachura. Support algorithm for blast furnace operation with optimal fuel consumption. *Journal of Mining and Metallurgy, Section B: Metallurgy*, Vol. 55, No. 1, 2019, pp. 31–38.

[8] Guha, M. Revealing cohesive zone shape and location inside blast furnace. *Ironmaking & Steelmaking*, Vol. 45, No. 9, 2018, pp. 787–792.

[9] La, G. H., J. S. Choi, and D. J. Min. Investigation on the reaction behaviour of partially reduced iron under blast furnace conditions. *Metals-Basel*, Vol. 11, No. 5, 2021, id. 839.

[10] Li, S., J. C. Chang, M. S. Chu, J. Li, and A. M. Yang. A blast furnace coke ratio prediction model based on fuzzy cluster and grid search optimized support vector regression. *Applied Intelligence*, Vol. 52, 2022, pp. 13533–13542.

[11] Roche, M., M. Helle, J. van der Stel, G. Louwerse, L. Shao, and H. Saxen. Off-line model of blast furnace liquid levels. *ISIJ International*, Vol. 58, No. 12, 2018, pp. 2236–2245.

[12] Shiau, J. S. and C. K. Ho. A visualization technique to predict abnormal channeling phenomena in the blast furnace operation. *Mining, Metallurgy & Exploration*, Vol. 36, No. 2, 2019, pp. 423–430.

[13] Zhang, X., M. Kano, and S. Matsuzaki. A comparative study of deep and shallow predictive techniques for hot metal temperature prediction in blast furnace ironmaking. *Computers & Chemical Engineering*, Vol. 130, 2019, id. 106575.

[14] Zhang, X., M. Kano, and S. Matsuzaki. Ensemble pattern trees for predicting hot metal temperature in blast furnace. *Computers & Chemical Engineering*, Vol. 121, 2019, pp. 442–449.

[15] Li, J. P., C. C. Hua, Y. N. Yang, and X. P. Guan. Data-driven Bayesian-based Takagi-Sugeno fuzzy modeling for dynamic prediction of hot metal silicon content in blast furnace. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 2, 2022, pp. 1087–1099.

[16] Sun, W. Q., Z. H. Wang, and Q. Wang. Hybrid event-, mechanism- and data-driven prediction of blast furnace gas generation. *Energy*, Vol. 199, 2020, id. 117497.

[17] Zhai, X. Y., M. T. Chen, and W. C. Lu. Fuel ratio optimization of blast furnace based on data mining. *ISIJ International*, Vol. 60, No. 11, 2020, pp. 2471–2476.

[18] Li, J. L., R. J. Zhu, P. Zhou, Y. P. Song, and C. Q. Zhou. Prediction of the cohesive zone in a blast furnace by integrating CFD and SVM modelling. *Ironmaking & Steelmaking*, Vol. 48, No. 3, 2021, pp. 284–291.

[19] Hu, Y., H. Zhou, S. Yao, M. Kou, Z. Zhang, L. P. Wang, et al. Comprehensive evaluation of the blast furnace status based on data mining and mechanism analysis. *International Journal of Chemical Reactor Engineering*, Vol. 20, No. 2, 2022, pp. 225–235.

[20] Li, W. B. and Z. Y. Chen. Breathing rate estimation based on multiple linear regression. *Computer Methods in Biomechanics and Biomedical Engineering*, Vol. 25, No. 7, 2022, pp. 772–782.

[21] Al-Najjar, H. A. H. and B. Pradhan. Spatial landslide susceptibility assessment using machine learning techniques assisted by additional data created with generative adversarial networks. *Geoscience Frontiers*, Vol. 12, No. 2, 2021, pp. 625–637.
[22] Huang, F. M., Z. Ye, S. H. Jiang, J. S. Huang, Z. L. Chang, and J. W. Chen. Uncertainty study of landslide susceptibility prediction considering the different attribute interval numbers of environmental factors and different data-based models. *Catena*, Vol. 202, No. 2, 2021, id. 117406.
[23] Shahzad, S. J. H., E. Bouri, T. Ahmad, and M. A. Naeem. Extreme tail network analysis of cryptocurrencies and trading strategies. *Finance Research Letters*, Vol. 44, 2022, id. 102106.
[24] Liu, D., S. Baldi, W. W. Yu, J. D. Cao, and W. Huang. On training traffic predictors via broad learning structures: A benchmark study. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 2, 2022, pp. 749–758.
[25] Scannell Bryan, M., J. Sun, J. Jagai, D. E. Horton, A. Montgomery, R. Sargis, et al. Coronavirus disease 2019 (COVID-19) mortality and neighborhood characteristics in Chicago. *Annals of Epidemiology*, Vol. 56, 2021, pp. 47–54.
[26] Sharif, M., M. A. Khan, M. Rashid, M. Yasmin, F. Afza and U. J. Tanik. Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images. *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 33, No. 4, 2021, pp. 577–599.
[27] Ebrahimi-Khusfi, Z., R. Taghizadeh-Mehrjardi, and M. Mirakbari. Evaluation of machine learning models for predicting the temporal variations of dust storm index in arid regions of Iran. *Atmospheric Pollution Research*, Vol. 12, No. 1, 2021, pp. 134–147.
[28] Cortes, C. and V. Vapnik. Support-vector networks. *Machine Learning*, Vol. 20, No. 3, 1995, pp. 273–297.
[29] Yang, X., L. Li, Q. L. Tao, W. C. Lu, and M. J. Li. Rapid discovery of narrow bandgap oxide double perovskites using machine learning. *Computational Materials Science*, Vol. 196, 2021, id. 110528.
[30] Zhao, Q. Z., Y. Liu, W. Q. Yao, and Y. B. Yao. Hourly rainfall forecast model using supervised learning algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 60, 2022, pp. 1–9.
[31] Gu, E. X. Convolutional neural network based Kannada-MNIST classification. *2021 IEEE International Conference on Consumer Electronics and Computer Engineering*, IEEE, Guangzhou, China, 15–17 January 2021, pp. 180–185.
[32] Rodriguez-Perez, R. and J. Bajorath. Evolution of support vector machine and regression modeling in chemoinformatics and drug discovery. *Journal of Computer-Aided Molecular Design*, Vol. 36, No. 5, 2022, pp. 355–362.
[33] Zhen, Z., T. Potta, N. A. Lanzillo, K. Rege, and C. M. Breneman. Development of a web-enabled SVR-based machine learning platform and its application on modeling transgene expression activity of aminoglycoside-derived polycations. *Combinatorial Chemistry & High Throughput Screening*, Vol. 20, No. 1, 2017, pp. 41–55.
[34] Chen, W. and Y. Li. GIS-based evaluation of landslide susceptibility using hybrid computational intelligence models. *Catena*, Vol. 195, 2020, id. 104777.
[35] Du, P. J., A. Samat, B. Waske, S. C. Liu, and Z. H. Li. Random forest and rotation forest for fully polarized SAR image classification using polarimetric and spatial features. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 105, 2015, pp. 38–53.
[36] Speiser, J. L., M. E. Miller, J. Tooze, and E. Ip. A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications*, Vol. 134, 2019, pp. 93–101.
[37] Li, H. H., B. Zhang, W. W. Hu, Y. Liu, C. W. Dong, and Q. S. Chen. Monitoring black tea fermentation using a colorimetric sensor array-based artificial olfaction system. *Journal of Food Processing and Preservation*, Vol. 42, No. 1, 2018, id. e13348.
[38] Wei, M. M., H. X. Lu, and H. H. Yang. Research on boold species identification algorithm based on RF_AdaBoost model. *Chemical Journal of Chinese Universities-Chinese*, Vol. 41, No. 1, 2020, pp. 94–101.
[39] Gupta, K. K., K. Kalita, R. K. Ghadai, M. Ramachandran, and X. Z. Gao. Machine learning-based predictive modelling of biodiesel production – A comparative perspective. *Energies*, Vol. 14, No. 4, 2021, id. 1122.
[40] Touzani, S., J. Granderson, and S. Fernandes. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy and Buildings*, Vol. 158, 2018, pp. 1533–1543.
[41] Natekin, A. and A. Knoll. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, Vol. 7, 2013, id. 21.
[42] Geurts, P., D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, Vol. 63, No. 1, 2006, pp. 3–42.
[43] Saeed, U., S. U. Jan, Y. D. Lee, and I. Koo. Fault diagnosis based on extremely randomized trees in wireless sensor networks. *Reliability Engineering & System Safety*, Vol. 205, 2021, id. 107284.
[44] Wei, J., Z. Li, M. Cribb, W. Huang, W. Xue, L. Sun, et al. Improved 1 km resolution PM2.5 estimates across China using enhanced space-time extremely randomized trees. *Atmospheric Chemistry and Physics*, Vol. 20, No. 6, 2020, pp. 3273–3289.
[45] Tao, Q., P. Xu, M. Li, and W. Lu. Machine learning for perovskite materials design and discovery. *NPJ Computational Materials*, Vol. 7, No. 1, 2021, pp. 171–88.
[46] Shi, L., D. P. Chang, X. B. Ji, and W. C. Lu. Using data mining to search for perovskite materials with higher specific surface area. *Journal of Chemical Information and Modeling*, Vol. 58, No. 12, 2018, pp. 2420–2427.
[47] Yu, D. R., S. An, and Q. H. Hu. Fuzzy mutual information based min-redundancy and max-relevance heterogeneous feature selection. *International Journal of Computational Intelligence Systems*, Vol. 4, No. 4, 2011, pp. 619–633.
[48] Holland, J. H. Genetic algorithms. *Scientific American*, Vol. 267, No. 1, 1992, pp. 66–72.
[49] Holland, J. H. Building blocks, cohort genetic algorithms, and hyperplane-defined functions. *Evolutionary Computation*, Vol. 8, No. 4, 2000, pp. 373–391.
[50] Jeon, H. and S. Oh. Hybrid-recursive feature elimination for efficient feature selection. *Applied Sciences-Basel*, Vol. 10, No. 9, 2020, id. 3211.
[51] Bustamam, A., A. Bachtiar, and D. Sarwinda. Selecting features subsets based on support vector machine recursive features elimination and one dimensional-naïve Bayes classifier using support vector machines for classification of prostate and breast Cancer, *4th International Conference on Computer Science and Computational Intelligence 2019* (*ICCSCI*). pp. 450–458.