Journal of Sustainable Metallurgy (2024) 10:250–266
https://doi.org/10.1007/s40831-024-00785-7

RESEARCH ARTICLE

Blast Furnace Thermal State Prediction Based on Multiobjective Evolutionary Ensemble Neural Networks

Tenghui Hu[1,4] · Xianpeng Wang[2,3] · Xiangman Song[1]

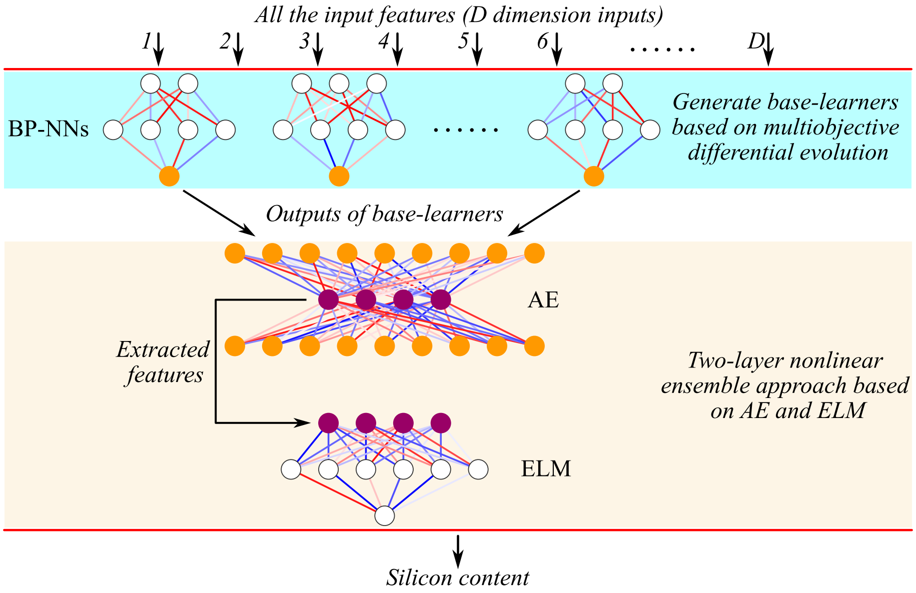
Received: 29 August 2023 / Accepted: 31 December 2023 / Published online: 8 February 2024

Abstract

Blast furnace ironmaking is the largest energy-consuming and greenhouse gas-emitting process in the iron and steel industry. As a key indicator of blast furnace operation status and energy consumption level, silicon content prediction has been very important for blast furnace operators to save energy and ensure stable ironmaking production. Traditional data-driven methods usually use feature selection when building models, which cannot guarantee high accuracy and generalization ability of the model due to the high correlation between features. To solve this problem, we propose an ensemble model based on multiobjective differential evolution, which consists of two stages. In the first stage, a multiobjective differential evolution is employed to evolve the input feature weights of each base-learner, and feature pruning is embedded in the evolution process to achieve sparse selection of all available features for the base-learners. In the second stage, a two-layer ensemble method based on autoencoder and extreme learning machine is proposed to achieve nonlinear ensemble of good candidate base-learners. Experimental results based on actual production data show that the developed model outperforms previous studies and can achieve more accurate values of silicon content, which in turn helps to reduce energy consumption in actual production.

Graphical Abstract

The modelling process of the proposed silicon content prediction model consists of two stages. In the first stage, a multiobjective algorithm is employed to evolve the input feature weights of each base-learner. In the second stage, a two-layer ensemble method is proposed to achieve nonlinear ensemble the base-learners.


All the input features (D dimension inputs)
1
2
3
4
5
6
......
D
BP-NNs
......
Generate base-learners based on multiobjective differential evolution
Outputs of base-learners
AE
Extracted features
Two-layer nonlinear ensemble approach based on AE and ELM
ELM
Silicon content

The contributing editor for this article was Adam Clayton Powell.

Extended author information available on the last page of the article

Keywords Blast furnace ironmaking · Evolutionary ensemble learning · Feature weight evolution · Silicon content prediction

Introduction

A typical blast furnace ironmaking system is shown in Fig. 1, which includes many subsystems such as feeding system, hot stove system, iron slag system, and so forth [1]. In the ironmaking process, raw materials containing iron ore are loaded from the top and move downward, while high-temperature hot air mixed with pulverized coke is blown in from the bottom and moves upward [1]. After some heat and mass transfer, chemical reactions, and phase changes [2], the main product, i.e., pig iron and two by-products, namely slag and blast furnace gas, are obtained.

In the era of emission peak and carbon neutrality, blast furnace ironmaking, as one of the processes with the highest energy consumption and carbon dioxide emissions in iron &steel companies, is the key to achieving green, healthy and sustainable development. Many companies are therefore committed to intelligent blast furnace ironmaking. An intelligent blast furnace system based on data-driven technologies consists of components such as key indicators prediction, blast furnace stability evaluation, and blast furnace operation optimization. The relationship of these components is shown in Fig. 2. It can be seen that the key indicator prediction models are at the center, as these models can not only predict important indicators during the blast furnace operation process, but also serve as a basis for other models.

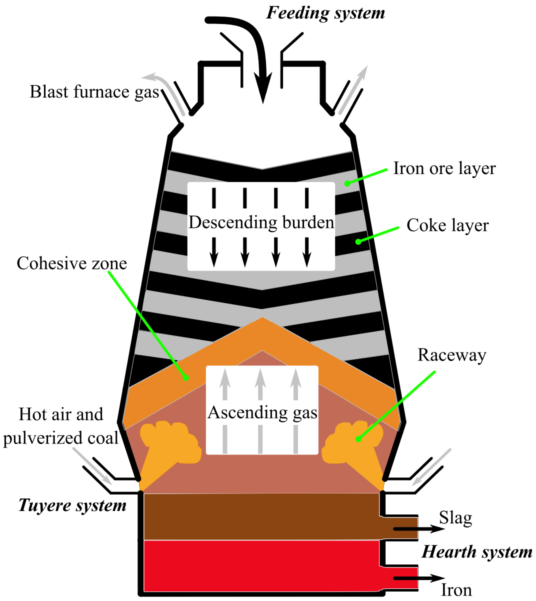
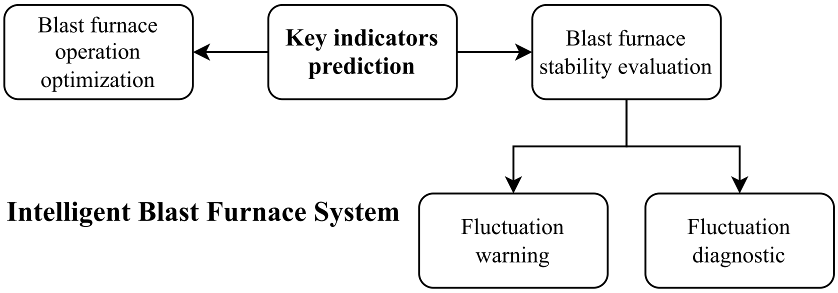


Fig. 1 A simple diagram of the blast furnace ironmaking

The prediction of silicon content in hot metal is a research hotspot in the field of key indicators prediction because of its close correlation with the thermal state of BF, which is crucial for the smooth and efficient operation of the smelting process [3, 4]. A decreasing [Si] indicates the blast furnace is cooling, which can lead to industrial accidents and blast furnace shutdowns; while a high [Si] indicates excessive generation of heat, wasting unnecessary fuels and raw materials [2]. Both result in additional CO_2 emissions. If silicon content can be accurately predicted, blast furnace operators can take timely action to avoid fluctuations and reduce unnecessary greenhouse gas emissions.

Since blast furnace ironmaking is a complex industrial process involving many systems, building an accurate mechanistic model of the process is very difficult. Therefore, similar to other scenarios of modern industrial production [5–8], data-driven modeling approaches have been investigated by many scholars in the field of [Si] prediction [9]. Yang et al. [10] proposed a soft-sensing model based on extreme learning machine (ELM) that utilized a pruning algorithm to optimize the weights of original ELM. Considering that different blast furnace operational conditions may occur in the ironmaking process and each condition has its own characteristics, Hua et al. employed hard C-means clustering [11]

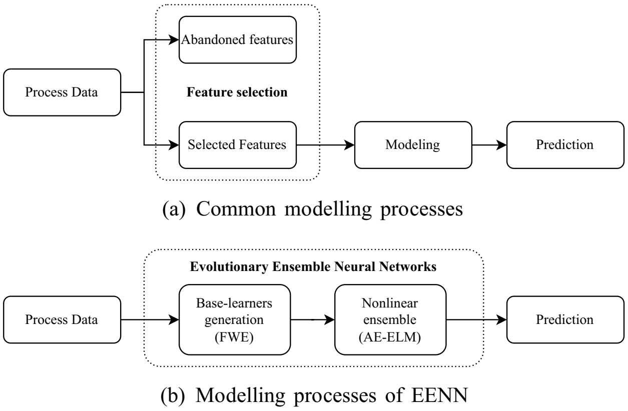
Fig. 2 The main components of an intelligent blast furnace system


Blast furnace operation optimization
Key indicators prediction
Blast furnace stability evaluation
Intelligent Blast Furnace System
Fluctuation warning
Fluctuation diagnostic

to separate the data into different groups and trained multiple support vector machines (SVRs) as predictors on these data groups. To have a more comprehensive monitoring of molten iron quality, Zhou et al. [12] presented a multioutput model for the multivariate prediction of quality parameters of molten iron based on online sequential random vector functional-link networks.

A point worth noting is that all of these methods mentioned above, as well as many other methods [3, 13, 14], employ Pearson correlation coefficient (PCC) or principal component analysis (PCA) to pre-process the input data with a large number of features, both of which expect a subset to represent the entire original dataset (as shown in Fig. 3a), when modeling the variation pattern of silicon content. Normally, PCC and PCA can serve to reduce model complexity, speed up training, and improve accuracy [11, 15]. However, due to the characteristics of high vanadium-titanium blast furnaces (HVT-BFs) that differ from normal BFs, these feature selection strategies may not perform as well as usual because they only take into account the linear relationship between variables. In addition to PCC and PCA, human experience and process knowledge are also used in variable selection [15, 16], but these methods are not universally applicable to the data-driven models because each blast furnace has its own unique properties. In addition, due to the high TiO_2 of the iron ore used in HVT-BFs, the smelting process is prone to phenomena such as slag stickiness and slag–iron separation issues. These issues make HVT-BFs more prone to fluctuations compared to common blast furnaces.

In order to fully tap the information contained in the data of BF production process, this paper proposes an ensemble learning method based on multiobjective optimization, which takes neural network as its base-learner. In the proposed evolutionary ensemble neural networks (EENN) model, the weights of input features of base-learners are taken as decision variables in the multiobjective evolutionary algorithm (MOEA). In combination with the feature pruning strategy, a set of base-learners with different input features can be obtained through the optimization of the MOEA. Furthermore, a nonlinear ensemble method based on autoencoder (AE) and extreme learning machine (ELM) [17] is proposed to construct the ensemble model. The main contributions of this paper can be summarized as follows.



(a) Common modelling processes

Evolutionary Ensemble Neural Networks
Process Data
Base-learners generation (FWE)
Nonlinear ensemble (AE-ELM)
Prediction

(b) Modelling processes of EENN

Fig. 3 Modeling process for EENN and common prediction models. EENN does not rely on a separate feature selection process and can use all features

1. Feature weight evolution (FWE) and feature pruning (FP) are proposed to deal with the input feature selection problem of [Si] prediction models in HVT-BFs. Through the application of FWE and FP, the derived ensemble model is prevented from being too redundant while allowing all features to play a role in it.
2. A nonlinear ensemble method based on AE and ELM is proposed for the integration of the base-learners generated through the co-evolutionary approach. Compared to linear methods such as weighted average, this method can better discover and exploit the connections between base-learners to enable better performance of the ensemble model.
3. Comprehensive experiments are conducted on a real-world industrial dataset to investigate the effects of different input feature selection methods on the modeling of [Si]. The results show that the commonly used PCC and PCA methods do not always perform well.

The remaining part of this paper is organized as follows. Section 2 provides some basic concepts about evolutionary ensemble learning as well as some other related work. The proposed prediction model is described in details in Sect. 3. In Sect. 4, some experiments and discussion are presented. Finally, Sect. 5 concludes the paper.

Related Works

Multiobjective Differential Evolution

Differential evolution (DE), proposed by Storn and Price, is a robust population-based global optimization method [18]. Because of its vector-based evolution operators, DE is particularly suitable for solving real-valued problems [19]. One of the most widely used DE variants is DE/rand/1/bin, which generates offspring individuals as follows:

1. Mutation: For a individual $\mathbf{x}_{i,G}$ (assuming it is a D-dimensional vector) in the G-th generation, a mutant vector is generated according to

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{p,G} + F \cdot \left(\mathbf{x}_{q,G} - \mathbf{x}_{r,G}\right) \tag{1}$$

where F is the differential weight, $\mathbf{x}_{p,G}$, $\mathbf{x}_{q,G}$, and $\mathbf{x}_{r,G}$ are three individuals randomly selected from the G-th generation population and satisfy $i \neq p \neq q \neq r$.

2. Crossover: The offspring individual $\mathbf{u}_{i,G+1}$ corresponding to $\mathbf{x}_{i,G}$:

$$\mathbf{u}_{i,G+1} = \left(u_{1i,G+1}, u_{2i,G+1}, \cdots, u_{Di,G+1}\right) \tag{2}$$

is generated, where

$$u_{di,G+1} = \begin{cases} v_{di,G+1}, & \text{if}(\mathit{uniform}(d) \leq CR)\text{or}(d = \mathit{rand}(i)) \\ x_{di,G}, & \text{if}(\mathit{uniform}(d) > CR)\text{and}(d \neq \mathit{rand}(i)) \end{cases} \tag{3}$$

In Equation (3), $d = 1, 2, \cdots, D$, $\mathit{uniform}(d)$ returns a uniform random number in the range $[0, 1]$, CR is the crossover constant and $\mathit{rand}(i)$ returns a random number $\in \{1, 2, \cdots, D\}$.

Multiobjective differential evolution (MODE) is an extended version of DE for multiobjective optimization problems [20, 21] and has been successfully applied in many fields [22]. In the proposed EENN, a MODE combining the main algorithmic flow of NSGA-II and the evolution operators of DE is adopted as the optimization method for generating base-learners. The general flow of the utilized MODE is shown in Fig. 4.

Evolutionary Ensemble Learning

As a class of powerful global optimization methods, population-based evolutionary algorithms have been widely used in various practical applications [23]. Among them, evolutionary ensemble learning (EEL) is a typical application that exploits the strong search ability of evolutionary algorithms for constructing ensemble models and has demonstrated excellent performance in many real-world problems [24, 25]. EEL usually consists of two phases: base-learners generation and base-learners integration.

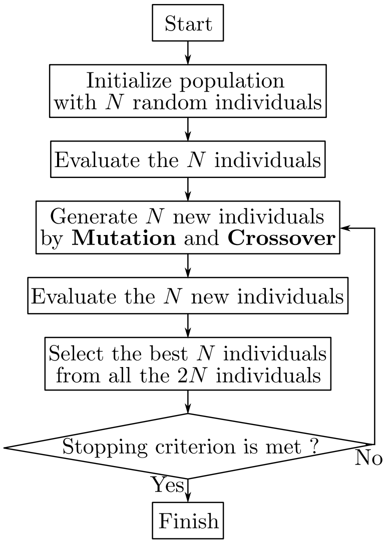

Start
Initialize population with N random individuals
Evaluate the N individuals
Generate N new individuals by **Mutation** and **Crossover**
Evaluate the N new individuals
Select the best N individuals from all the $2N$ individuals
Stopping criterion is met ?
No
Yes
Finish

Fig. 4 Flow chart of multiobjective differential evolution

In the first phase, base-learners generation is modeled as a two-objective optimization problem that is solved by MOEAs, while base-learners are encoded as individuals in the algorithm population. To ensure that base-learners have high accuracy, the first objective of the multiobjective optimization problem is prediction accuracy. The second objective could be the diversity, complexity, or sparsity of base-learners.

Upon completion of the first phase, a set of base-learners can be obtained and used in full or in part for subsequent ensemble. The most common ensemble methods are simple average and weighted average.

Proposed Algorithm

From the modeling process shown in Fig. 3b, it can be seen that the proposed model does not rely on a separate feature selection method, but takes all available features as inputs. The proposed FWE takes these inputs as the basis for generating multiple base-learners using MODE combined with back-propagation (BP) algorithm. Thereafter, the obtained base-learners are combined in a nonlinear way. Several characteristics of the proposed algorithm are shown below.

1. The weights of the input features are taken as individuals in the MODE population to achieve the automatic selection of input feature weights for each base-learner.
2. Before training the base-learner, the training data are selected and scaled according to the corresponding feature weights. Therefore, the training data vary for each base-learner.
3. A nonlinear ensemble method based on AE and ELM is employed instead of the commonly used linear ensemble method. And all individuals in the final population are used instead of only Pareto individuals.

The rest part of this section first gives the overall framework of the algorithm, followed by some detailed explanations.

Algorithm Framework

Algorithm 1 presents the framework of our proposed algorithm. Lines 1-16 are the base-learners generation part. The main process of this part is the same as that of NSGA-II except the *Decoding* operation is required before evaluating the prediction accuracy of individuals. The individuals in the population are one-dimensional real vectors of length *n*, where each value represents the weight of a corresponding input feature. Since the individuals are vectors encoded by real numbers, *Mutation* and *Crossover* operations from DE/rand/1/bin [18] are employed to generate new individuals.

After the base-learners generation part is completed, all the individuals in the final population $\mathcal{P}$ are used for subsequent ensemble, as shown in Line 17. These individuals are first decoded into neural networks; then, some key features of the output of these base-learners are extracted by AE and finally combined using ELM.

Encoding and Decoding of Individuals

The encoding and decoding of individuals in the population are depicted in Fig. 5. As mentioned in Sect. 3.1, the individuals in the population are one-dimensional real vectors ($\mathbf{x}$ in Fig. 5). In order to evaluate their prediction accuracy and individual diversity, they need to be decoded into corresponding neural networks first. The *Decoding* operation is shown in Algorithm 2.

Algorithm 1: Framework of EENN

```
Input: training data set $\Omega$, population size $N$, maximum iterative number $maxIter$, hidden layer
       node numbers $h$, learning rate $\eta$, weight threshold $\alpha$;
Output: ensemble model;
1  Initialize population $\mathcal{P}$ with $N$ random individuals;
2  for each individual $\mathbf{x}$ in $\mathcal{P}$ do
       // Algorithm 2
3      $M \leftarrow Decoding(\mathbf{x}, \Omega, \eta, \alpha, h)$;
4      $CalAccuracy(M, \Omega)$;                                        // Equation(4)
5  $t = 0$;
6  while $t < maxIter$ do
7      $t = t + 1$;
8      $\mathcal{U} \leftarrow \mathcal{P}$;                        // copy all the individuals in $\mathcal{P}$ to $\mathcal{U}$
9      for each individual $\mathbf{x}$ in $\mathcal{P}$ do
10         $\mathbf{v} \leftarrow Mutation(\mathbf{x})$;                       // Equation(1)
11         $\mathbf{u} \leftarrow Crossover(\mathbf{x}, \mathbf{v})$;                 // Equation(3)
12         $M \leftarrow Decoding(\mathbf{u}, \Omega, \eta, \alpha, h)$;
13         $CalAccuracy(M, \Omega)$;
14         $\mathcal{U} \leftarrow \mathcal{U} \cup \{\mathbf{u}\}$;                  // add $\mathbf{u}$ to $\mathcal{U}$
15     $CalDiversity(\mathcal{U})$;                                   // Equation(5)
       // the same as that in NSGA-II
16     $\mathcal{P} \leftarrow GetNextPopulation(\mathcal{U})$;
17 $Ensemble(\mathcal{P})$;                                       // decode and ensemble
18 return ensemble model;
```

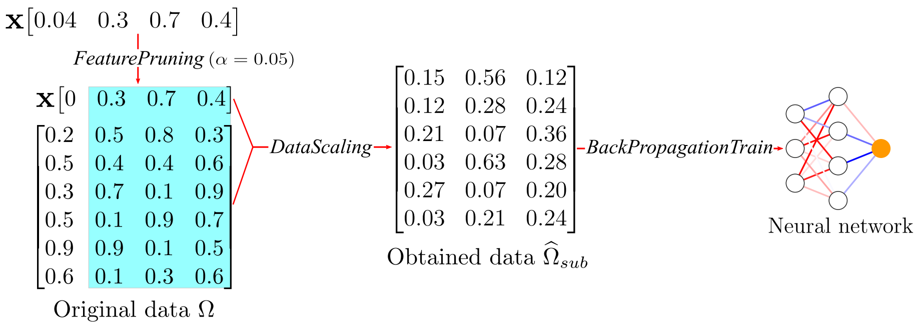


Fig. 5 The encoding and decoding of an individual. In this example, the dataset Ω has 6 samples and 4 input features. The shaded area indicates the features used and the $\widehat{\Omega}_{sub}$ is the final dataset used to train this network

```
Algorithm 2: Decoding
Input: individual $\mathbf{x}$, training data set $\Omega$, range of hidden layer node numbers $h$, learning rate $\eta$,
       weight threshold $\alpha$;
Output: neural network $M$;
$\mathbf{x} \leftarrow FeaturePruning(\mathbf{x}, \alpha)$;                        // Algorithm 3
$\widehat{\Omega}_{sub} \leftarrow DataScaling(\mathbf{x}, \Omega)$;                 // Fig.5
$M \leftarrow BackPropagationTrain\left(h, \eta, \widehat{\Omega}_{sub}\right)$;
return $M$;
```

```
Algorithm 3: FeaturePruning
Input: individual $\mathbf{x}$, weight threshold $\alpha$;
Output: modified individual $\mathbf{x}$;
for each weight $w$ in $\mathbf{x}$ do
    if $w \leq \alpha$ then
        $w = 0$;
return $\mathbf{x}$;
```

The first step in training a neural network is to determine its structure and the training data to be used, as well as some other hyperparameters, such as the learning rate η. In the proposed algorithm, the base-learners are all neural networks with a single hidden layer, which means there are three structural parameters to be determined, i.e., how many nodes there are in the input, hidden, and output layers, respectively. The number of output layer nodes equals to one, which is in line with the problem studied in this paper, while the number of hidden layer nodes is a pre-defined value h. As for the number of input layer nodes, it is equal to the number of non-zero values remaining in the individual $\mathbf{x}$ after *FeaturePruning* operation.

As illustrated in Fig. 5, obtaining training data for the individual $\mathbf{x}$ involves a two-step process. First, the data corresponding to the retained features are determined from the feature pruning results. In the instance illustrated in Fig. 5, features 2 to 4 are used. Second, the corresponding data are scaled to get the final data used to train this neural network. This increases the diversity of the base-learners at both feature and data levels.

After obtaining all the required parameters and corresponding training data, a neural network model M can be trained using back-propagation (BP) algorithm and evaluated on the training set. The evaluation results for M are considered to be the evaluation results for individual $\mathbf{x}$.

Performance Metrics

The optimization objectives of the base-learners, i.e., the two objectives of the MOEA, are prediction accuracy and individual diversity, respectively.

Prediction Accuracy

The mean square error (MSE) is employed to evaluate the accuracy of prediction results of base-learners. Therefore, the prediction accuracy of i-th base-learner is defined as:

$$PA_i = \frac{1}{L} \sum_{l=1}^{L} \left(y_l - \hat{y}_l\right)^2 \tag{4}$$

where L represents the total number of samples, y_l and $\hat{y}_l$ are the true and predicted values of the l-th sample, respectively.

Individual Diversity

The well-tested idea, negative correlation learning (NCL) [26], is employed as the second objective to promote the diversity among base-learners. Its definition is shown below:

$$ID_i = \sum_{l=1}^{L} \left|y_l^i - y_l^{\mathcal{U}}\right| \left(\sum_{j=1, j \neq i}^{2N} \left|y_l^j - y_l^{\mathcal{U}}\right| \right) \tag{5}$$

where y_l^i and y_l^j denote the predicted values of the i-th and j-th individuals in the population $\mathcal{U}$ for the l-th sample,

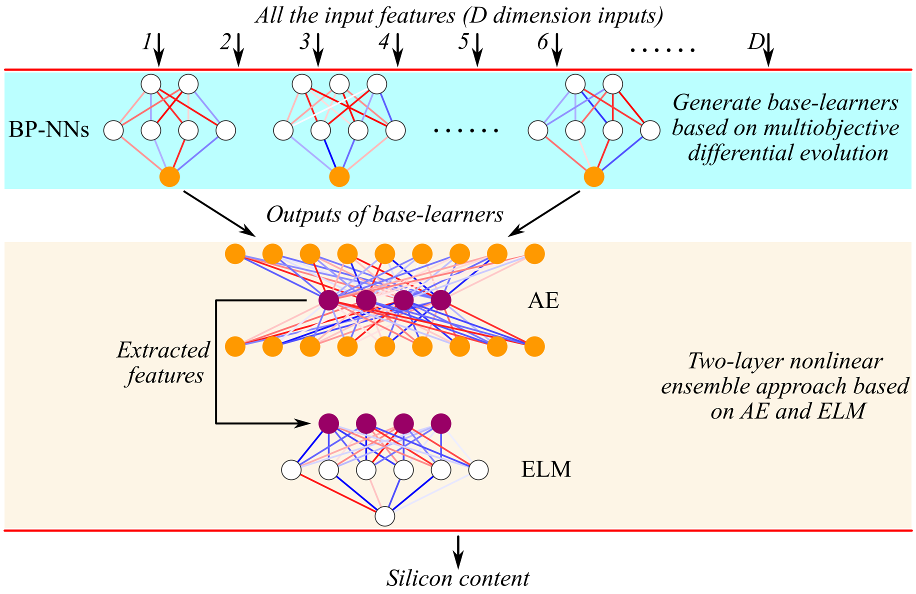

All the input features (D dimension inputs)
1
2
3
4
5
6
......
D
BP-NNs
......
Generate base-learners based on multiobjective differential evolution
Outputs of base-learners
AE
Extracted features
Two-layer nonlinear ensemble approach based on AE and ELM
ELM
Silicon content

Fig. 6 Modeling process of the proposed silicon content prediction model

respectively, and $y_l^{\mathcal{U}}$ is the average output of all individuals for the l-th sample.

After evaluating all the $2N$ individuals in the population $\mathcal{U}$, the best N individuals are selected from them as the next generation according to the *GetNextPopulation* operation (Line 16 in Algorithm 1).

Nonlinear Ensemble Method Based on AE and ELM

Most evolutionary ensemble learning algorithms use only Pareto individuals in the ensemble phase [27, 28], but because all individuals in the population are generated collaboratively, using only Pareto individuals for ensemble runs the risk of wasting information. In addition, in the ensemble learning scenario, since the base-learners are trained on the same training set, the outputs of the obtained base-learners are inevitably linearly correlated with each other even with various measures to enhance their diversities. This leads to a very redundant input to the final ensemble model, which in turn affects its accuracy and generalization ability. To address these issues, an ensemble approach based on AE and ELM (referred to as AE-ELM) is proposed in this paper.

Commonly used autoencoders usually have a three-layer structure, including an input layer, a hidden layer, and an output layer. And the number of nodes in these three layers satisfies: input layer = output layer > hidden layer. Due to the special structure of autoencoders, it is often used to reduce data dimensionality and extract key features in unsupervised learning. In the proposed ensemble approach, the output of each base-learner is taken as an input to the AE, which can then extract a number of key features from the outputs of numerous base-learners. These extracted key features are used as inputs to the ELM for final prediction. In summary, the work to be done in the second phase consists of the following 3 steps:

Table 1 All the candidate input features for modeling

Input feature type	Variable name
Feeding	Burden ratio
Top gas treatment	Blast furnace gas [CO], Gas utilization rate Blast furnace gas [CO_2]
Hot blast	Blast kinetic energy, Blast volume Actual air velocity, Standard air velocity Flow rate of oxygen, Hot air temperature Oxygen-enriched pressure, Gas permeability, Pressure differential Gas volume of bosh, Bosh gas index Hot air pressure
Coal injection	Coal injection rate, Coal injection volume Coal ratio, Ratio of coal to coke
Hearth	Pig Iron [C], Pig Iron [S] Slag [TiO_2], Slag [SiO_2], Slag [Al_2O_3] D-value of Pig Iron [Ti][a]
Others	Average top gas temperature Range of top gas temperature Average throat temperature Range of throat temperature Adiabatic flame temperature Average hearth temperature[b] Average shaft temperature[b] Top pressure

[a] This variable represents the value of the change in titanium content in pig iron

[b] The average of temperatures measured by thermocouples installed in the furnace wall

1. All individuals in the final population of the base-learners generation phases are decoded into corresponding BP-NNs.
2. The outputs of these BP-NNs are processed by a AE to extract compact and representative features.
3. The ELM is employed to model the extracted key features and construct an ensemble model to obtain the final outputs.

Figure 6 illustrates the construction process of EENN, assuming a total of D input features. In most existing prediction models, a subset of the D-dimensional input variables must first be selected for modeling, which means that the information contained in the features that are not selected cannot be utilized by the model. However, the model designed in this paper is different in that EENN accepts all the features and generates a set of high-quality base-learners using feature weight evolution. These base-learners have different input features and training data, which allow them to fully mine the knowledge in the process data (i.e., D-dimensional input variables). Thus, the purpose of FWE is not to select a few important features, but to enhance the quality of base-learners. The outputs of these base-learners are further processed by the AE-ELM ensemble approach to obtain the final model. Thus, as a whole, EENN can be viewed as a model that takes all D features as inputs to make predictions about silicon content.

Supplementary Information

GetNextPopulation in Algorithm 1 is from NSGA-II without any modification. Considering the limited space of the article, no further explanation is given in the text. Interested readers can find detailed information about this method in the literature [29].

Experimental Studies

Experiment Preparation

Dataset

The data used are taken from a blast furnace at a major iron &steel company in China during the first half of 2021. The operation of the blast furnace has been relatively stable over this period, so the dataset is representative of typical operating conditions of the blast furnace. This dataset contains a total of 1950 samples, each of them consists of one target variable and 34 input variables. The target variable is [Si], while all the candidate input features are listed in Table 1. There are three main sources of the data:

1. Measured directly by sensors. For example, Hearth temperature is measured by thermocouples installed in the furnace wall; Blast Furnace Gas [CO] is measured by sensors installed at the top of the blast furnace.
2. Calculated by relevant formulas or mechanism models embedded in the control system, e.g., Adiabatic Flame Temperature and Gas Volume of Bosh.
3. Obtained from assay results of molten iron samples or slag samples, e.g., Pig Iron [C] and Slag [Al_2O_3].

In addition, some variables such as Gas Utilization Rate are also included in the input features. Although it is determined by Blast Furnace Gas [CO] and Blast Furnace Gas [CO_2], it can be viewed as a processed high-level feature. The use of such features can improve the richness of the input features and thus enhance the performance of the model.

As did in some literatures, those variables with a PCC greater than 0.3 with silicon content are selected as input features for modeling. The variables in the dataset used in this paper that fulfill this condition are bolded in Table 1. Note that some variables that are commonly thought to have a significant effect on [Si], such as Coal Rate, are not bolded. This is because each blast furnace has its own characteristics and even the same blast furnace cannot maintain the same smelting conditions all the time. The correlation coefficients are calculated for the dataset used in the experiments, and variables such as Coal Rate have correlation coefficients less than 0.3 and are therefore not bolded.

It is important to note that as part of an actual project, the data used in this paper were determined in consultation with the on-site experts. And due to limitations in data collection capabilities, not all variables affecting [Si] can be included.

A total of 400 samples are randomly selected from the dataset as test data and the remaining 1550 samples are taken as training data. Before conducting experiments, the relevant data are normalized according to Equation (6).

$$\mathbf{x}_{norm} = \frac{\mathbf{x} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}} \tag{6}$$

Model Performance Evaluation

After the ensemble model is trained on the training set, its performance is evaluated on the test set data. Three model evaluation criteria, hit rate (HR), coefficient of determination (R^2), and root mean square error (RMSE), are used, and their calculation formulas are shown in Equations (7), (8) and (9), where L is the number of samples, y_l is the true value of the l-th sample, $\hat{y}_l$ is the predicted value of the model for the l-th sample, and $\bar{y} = \frac{1}{L}\sum_{l=1}^{L} y_l$.

Table 2 Test results on the prediction accuracy of some existing models when using different input feature schemes

Prediction models	Input features	HR		R^2		RMSE	
		Mean	Std	Mean	Std	Mean	Std
HCM-SVRs	All features[1]	9.00*e*-01	5.39*e*-03	6.56*e*-01	1.15*e*-02	6.14*e*-02	1.02*e*-03
	PCA features[2]	8.97*e*-01	2.78*e*-03	6.11*e*-01	5.91*e*-03	6.53*e*-02	4.99*e*-04
	PCC features[3]	8.92*e*-01	3.85*e*-03	6.38*e*-01	4.19*e*-03	6.30*e*-02	3.64*e*-04
Bagging-NN	All features	9.12*e*-01	1.91*e*-03	6.72*e*-01	2.24*e*-03	5.99*e*-02	2.05*e*-04
	PCA features	8.99*e*-01	1.90*e*-03	6.45*e*-01	2.05*e*-03	6.23*e*-02	1.80*e*-04
	PCC features	8.82*e*-01	2.63*e*-03	6.31*e*-01	1.56*e*-03	6.36*e*-02	1.35*e*-04
dGRU-RNN	All features	9.05*e*-01	7.02*e*-03	6.54*e*-01	1.18*e*-02	6.16*e*-02	1.04*e*-03
	PCA features	8.90*e*-01	6.74*e*-03	6.12*e*-01	7.73*e*-03	6.52*e*-02	6.50*e*-04
	PCC features	8.83*e*-01	4.29*e*-03	6.22*e*-01	6.29*e*-03	6.43*e*-02	5.32*e*-04
Modified-ELM	All features	9.27*e*-01	1.23*e*-02	6.39*e*-01	1.07*e*-01	6.30*e*-02	9.15*e*-03
	PCA features	9.04*e*-01	1.20*e*-02	6.58*e*-01	3.46*e*-02	6.17*e*-02	2.25*e*-03
	PCC features	9.18*e*-01	1.27*e*-02	6.77*e*-01	4.21*e*-02	5.99*e*-02	4.24*e*-03
NEA-ANCL	All features	8.91*e*-01	5.47*e*-03	6.41*e*-01	6.58*e*-03	6.27*e*-02	5.72*e*-04
	PCA features	7.52*e*-01	8.19*e*-03	3.58*e*-01	1.60*e*-02	8.39*e*-02	1.04*e*-03
	PCC features	8.79*e*-01	4.86*e*-03	6.03*e*-01	7.54*e*-03	6.59*e*-02	6.26*e*-04

[1] The test results obtained when using all the input features

[2] The test results obtained when using the PCA reduced-dimensional features

[2]The test results obtained when using some of the features selected by PCC

$$H_l = \begin{cases} 1, & |\hat{y}_l - y_l| < 0.1 \\ 0, & others \end{cases}$$
$$\mathrm{HR} = \frac{1}{L}\left(\sum_{l=1}^{L} H_l\right) \tag{7}$$

$$\mathrm{R}^2(y, \hat{y}) = 1 - \frac{\sum_{l=1}^{L} (y_l - \hat{y}_l)^2}{\sum_{l=1}^{L} (y_l - \bar{y})^2} \tag{8}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{L}\sum_{l=1}^{L} (y_l - \hat{y}_l)^2} \tag{9}$$

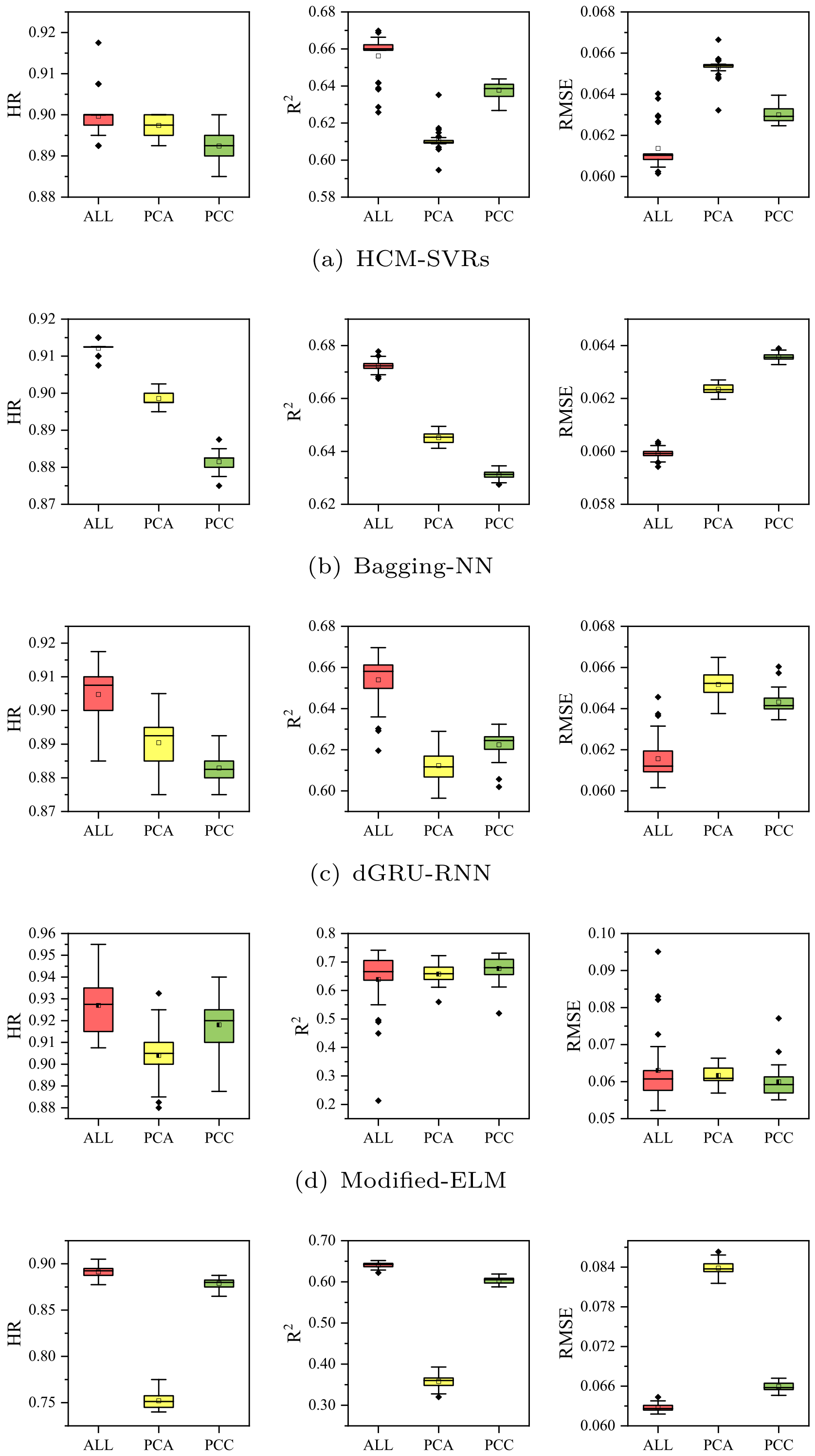
HR is a common criterion used by on-site operators to evaluate the accuracy of silicon content prediction models and has also been used in many existing research papers. As for R^2 and RMSE, two commonly used criteria for evaluating regression models in the machine learning community are also employed to compare the performance of silicon content prediction models from different perspectives.

Experiments of Different Models with Different Inputs

This section gives the test results of some silicon content prediction models as well as an evolutionary ensemble model with different input feature schemes. A brief description of these models is given below.

- HCM-SVRs: This model first divides the dataset into different groups using hard C-means clustering and then trains multiple SVRs on these data groups for subsequent prediction [11].
- Bagging-NN: The classical Bagging algorithm with neural networks as the base-learners [30].
- dGRU-RNN: An enhanced recurrent neural network that simplifies the update and reset gates in gated recurrent unit (GRU) into one disposition gate (dGRU) to achieve simpler structure and higher efficiency [3].
- Modified-ELM: An improved ELM algorithm that employs a modified pruning method to optimize the weights of original ELM [10].
- NEA-ANCL: A niching evolutionary ensemble algorithm with adaptive negative correlation learning (ANCL). The unique feature of this algorithm is the modified dynamical fitness sharing method to preserve the diversity of population and ANCL to balance the accuracy and diversity of base-learners [31].

The three input feature schemes used in the experiments are all features, PCA features, and PCC features. In PCA features, the original 34 features are processed by PCA. And 10 principal components are retained as the final inputs, with a combined explained variance ratio of 91.25%. For PCC features, the correlation coefficient between each feature in



(a) HCM-SVRs

(b) Bagging-NN

(c) dGRU-RNN

(d) Modified-ELM

(e) NEA-ANCL

Fig. 7 Boxplots of the test results on the performance of some existing models when using different input feature schemes. In the horizontal coordinate of each figure, ALL represents the test results using all of the input features, PCA represents the test results using the PCA features, and PCC represents the test results using the PCC features

the original dataset and the silicon content is first calculated, and those features with a correlation coefficient greater than 0.3 are used as inputs (bolded in Table 1).

Each experiment is run 30 times independently to ensure that the results obtained are credible. Table 2 shows the performance of different prediction models on the test data with different input feature schemes, in which the mean (*Mean*) and standard deviation (*Std*) of each evaluation criterion are calculated. The larger *Mean* of HR and R^2, the smaller *Mean* of RMSE, indicating the higher prediction

Table 3 Best performance count for each input feature scheme

	Mean count	Std count
All feature	13	2
PCA	0	5
PCC	2	8

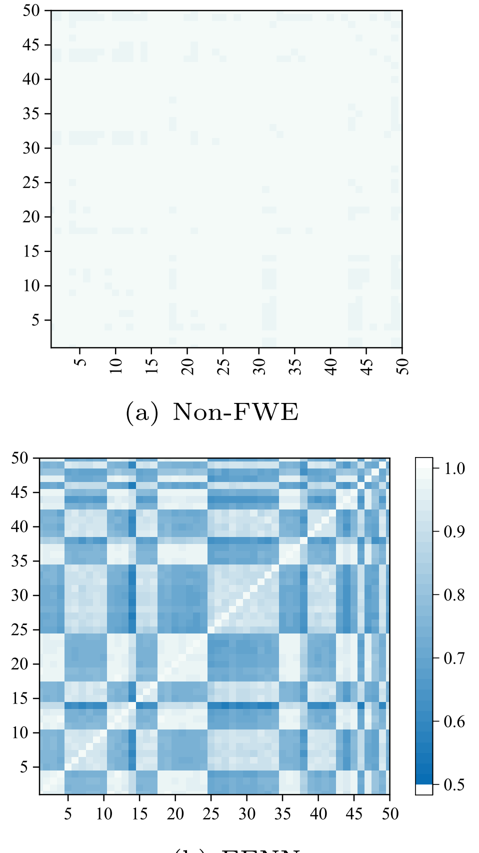
accuracy of the model; the smaller *Std*, the better stability of the model. In addition, the distribution of the results of these 30 repeated experiments for each model is shown in the boxplots in Fig. 7.

The best performance of each model under different input feature schemes is statistically analyzed from two aspects of accuracy and stability, and the results are shown in Table 3. As can be seen, the *Mean* count of All feature reaches 13, which means that when all the features are used for modeling, these models tend to achieve higher accuracy. This result indicates that our inference that PCC and PCA do not perform as well as before in the HVT-BF scenario is correct, because some nonlinear information implied in the dataset may be lost in the process of linear data dimensionality reduction. However, in terms of *Std* count, the opposite is true. When modeling with PCA features or PCC features, most of the models exhibited better robustness. Therefore, if the on-site operator has high requirements for model stability, using PCC or PCA to pre-process data is still a reliable method.

Validation of the Proposed Strategies

The effectiveness of the proposed strategies in this paper is verified in this section. All statistical results in this section are obtained from 30 independent experiments, and the Wilcoxon rank-sum test is conducted at a significance level of 0.05 to determine whether the performance differences of different models are statistically significant. Symbols "+" and "−" denote that our proposed model is significantly better or worse than the rivals, respectively. Symbol "=" indicates that the difference between the involved models is not significant.

The main parameters of the proposed algorithm are set as follows: in the base-learners generation phase, $N = 50$, $maxIter = 80$, $h = 30$, $\eta = 0.001$, $\alpha = 0.05$, and the initial weights of the input features are generated by normal distribution with mean of 1 and variance of 0.5; while in the ensemble phase, the network structures (i.e., the number of nodes per layer) of AE and ELM are $50 - 35 - 5 - 35 - 50$ and $5 - 10 - 1$, respectively.



(a) Non-FWE

(b) EENN

Fig. 8 Correlation coefficient heat maps of the outputs of the base-learners. **a** The case of Non-FWE with a VIF equal to 1464.72. **b** The case of EENN with a VIF equal to 402.08

Table 4 Statistical results of EENN and Non-FWE in terms of mean and standard deviation of prediction performance

	HR	R^2	RMSE
EENN	$9.16e\text{-}01_{5.02e\text{-}03}$	$7.04e\text{-}01_{9.94e\text{-}03}$	$5.70e\text{-}02_{9.55e\text{-}04}$
Non-FWE	$9.16e\text{-}01_{6.06e\text{-}03}^{=}$	$6.95e\text{-}01_{7.33e\text{-}03}^{+}$	$5.78e\text{-}02_{6.97e\text{-}04}^{+}$

Feature Weight Evolution

To verify the validity of feature weighting evolution, an algorithm (denoted as non-FWE) that is identical to Algorithm 1 except for the absence of FWE is developed. As shown in Algorithm 4, the developed algorithm retains the multiobjective selection mechanism and the nonlinear ensemble method, which makes the validation results for FWE convincing.

```
Algorithm 4: Framework of Non-FWE
  Input: training data set $\Omega$, population size $N$, maximum iterative number $maxIter$, hidden layer
         node numbers $h$, learning rate $\eta$;
  Output: ensemble model of Non-FWE;
  Train $N$ neural networks on $\Omega$ as the initial population and evaluate their accuracy according to
    Equation(4);
  $t = 0$;
  while $t < maxIter$ do
      $t = t + 1$;
      $\mathcal{U} \leftarrow \mathcal{P}$;
      for $i := 1$ to $N$ do
          $M \leftarrow BackPropagationTrain(h, \eta, \Omega)$;
          $CalAccuracy(M, \Omega)$;
          $\mathcal{U} \leftarrow \mathcal{U} \cup \{M\}$;                          // add $M$ to $\mathcal{U}$
      $CalDiversity(\mathcal{U})$;                                   // Equation(5)
      $\mathcal{P} \leftarrow GetNextPopulation(\mathcal{U})$;
  $Ensemble(\mathcal{P})$;                                          // According to Fig.6
  return ensemble model;
```

First, in the base-learners generation phase, we examined the correlation relationship among the outputs of the base-learners obtained by each of these two algorithms. Figure 8 shows the correlation coefficient heat maps of the outputs of the base-learners for the two cases of absence and presence of FWE. As can be seen from Fig. 8, the application of FWE greatly reduces the correlation among the outputs of the base-learners, which means that the diversity of the obtained base-learners is greatly improved. More precisely, the variance inflation factor (VIF) [32] is reduced from 1464.72 to 402.08, indicating that FWE considerably alleviates the multicollinearity problem in the output of the base-learners.

Then, the base-learners obtained by these two algorithms are combined separately using the nonlinear ensemble method described in Sect. 3.4, and the statistical results of the means and standard deviations are listed in Table 4. As can be seen in Table 4, EENN performs similarly to non-FWE in terms of HR, but statistically outperforms the latter in terms of R^2 and RMSE. This indicates that the prediction results obtained by EENN have smaller errors with the true [Si] and are much closer to its actual variation pattern.

Table 5 Statistical results of different ensemble methods in terms of mean and standard deviation of prediction performance

	HR	R^2	RMSE
AE-ELM	$9.16e\text{-}01_{5.02e\text{-}03}$	$7.04e\text{-}01_{9.94e\text{-}03}$	$5.70e\text{-}02_{9.55e\text{-}04}$
Only-ELM	$9.14e\text{-}01_{6.87e\text{-}03}^{=}$	$6.90e\text{-}01_{1.52e\text{-}02}^{+}$	$5.83e\text{-}02_{1.43e\text{-}03}^{+}$
AE-ELM$_P$	$9.15e\text{-}01_{7.69e\text{-}03}^{=}$	$6.96e\text{-}01_{9.58e\text{-}03}^{+}$	$5.77e\text{-}02_{9.09e\text{-}04}^{+}$
WA	$9.17e\text{-}01_{7.09e\text{-}03}^{=}$	$6.95e\text{-}01_{8.87e\text{-}03}^{+}$	$5.78e\text{-}02_{8.35e\text{-}04}^{+}$

The above experimental results demonstrate the effectiveness of FWE in modeling [Si]. The application of FWE enables the base-learners to adjust the weights of their input features by MOEA. And different weight combinations of input features make the base-learners tend to acquire different aspects of the dataset, thus enhancing the diversity among base-learners and helping to improve the accuracy and generalization ability of the ensemble model.

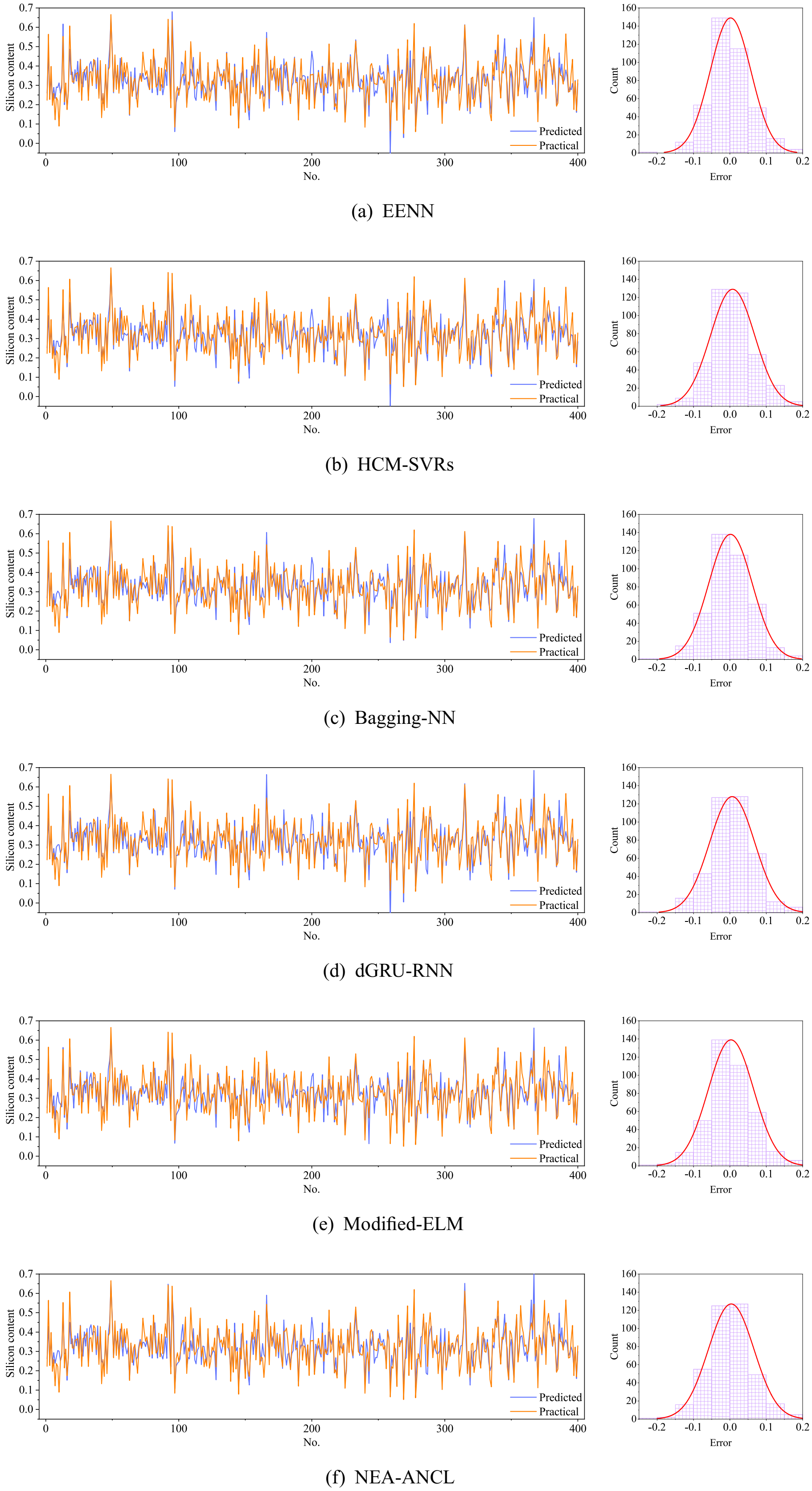
Ensemble Method Based on AE and ELM

It can be seen from the analysis in Sect. 4.3.1 that although FWE has enhanced the diversity of base-learners, there is still a strong linear correlation between their outputs. Therefore, it is necessary to process the original outputs of the base-learners before ensemble. In the experiments in this subsection, we employ an AE with three hidden layers to extract 5-dimensional features from the output of 50 base-learners.

Table 5 gives the experimental results of the proposed ensemble method compared with the commonly used weighted average (WA) method using only Pareto individuals. In WA, the weight of each Pareto individual is obtained by a differential evolution algorithm. In addition, the experimental results of some variants of AE-ELM are listed in Table 5 for reference. Only-ELM refers to the direct ELM ensemble without feature extraction via AE, where the structure of ELM is set to $50 - 60 - 1$. AE-ELM$_P$ refers to the ensemble with only Pareto individuals, and since the number of Pareto individuals is relatively small, an AE with only one hidden layer is used to extract 5-dimensional features from their outputs.

Only-ELM performs worse compared to AE-ELM, indicating that the redundant inputs are detrimental to the

Fig. 9 Predicted curves and corresponding error distribution of each prediction model on the test set data



(a) EENN

(b) HCM-SVRs

(c) Bagging-NN

(d) dGRU-RNN

(e) Modified-ELM

(f) NEA-ANCL

Table 6 Performance comparison of the proposed ensemble prediction model with some existing prediction models

	HR	R^2	RMSE
EENN	$9.16e-01_{5.02e-03}$	$7.04e-01_{9.94e-03}$	$5.70e-02_{9.55e-04}$
HCM-SVRs	$9.00e-01_{5.39e-03}{}^{+}$	$6.56e-01_{1.15e-02}{}^{+}$	$6.14e-02_{1.02e-03}{}^{+}$
Bagging-NN	$9.12e-01_{1.91e-03}{}^{+}$	$6.72e-01_{2.24e-03}{}^{+}$	$5.99e-02_{2.05e-04}{}^{+}$
dGRU-RNN	$9.05e-01_{7.02e-03}{}^{+}$	$6.54e-01_{1.18e-02}{}^{+}$	$6.16e-02_{1.04e-03}{}^{+}$
Modified-ELM	$9.18e-01_{1.27e-02}{}^{=}$	$6.77e-01_{4.21e-02}{}^{+}$	$5.99e-02_{4.24e-03}{}^{+}$
NEA-ANCL	$8.91e-01_{5.47e-03}{}^{+}$	$6.41e-01_{6.58e-03}{}^{+}$	$6.27e-02_{5.72e-04}{}^{+}$

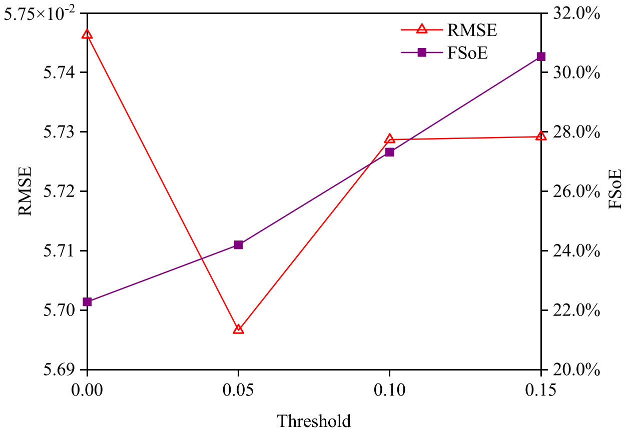


Fig. 10 Variation of prediction accuracy and FSoE with weight threshold α

prediction accuracy of the final ensemble model. This result demonstrates the need for the AE-based feature extraction. In the other case, if only Pareto individuals are used for feature extraction and ensemble (AE-ELM$_P$), the performance of the obtained model is also inferior to that of AE-ELM, indicating that although Pareto individuals are more representative, other individuals in the population can also contribute to the ensemble model. The proposed ensemble method is also superior to WA as shown in Table 5.

Experimental Results of the Proposed Prediction Model

This section gives the experimental results of the proposed model in comparison with other rivals. From the results in Sect. 4.2, among these comparative models, HCM-SVRs, Bagging-NN, dGRU-RNN, and NEA-ANCL have higher accuracy when using all features for modeling; for Modified-ELM, on the whole, the model using PCC features performs better. Therefore, for these models, only their best results are presented.

Figure 9 shows the predicted curves and corresponding error distribution of each model on the test set data. From the curve trend of the predicted values by different models, the proposed method has stronger tracking ability for the changes of actual values. And the approximate error distribution of EENN is thinner and taller than the others, which indicates that the predicted values of EENN are closer to the practical values of silicon content.

The statistical results of thirty repeated experiments of each model are shown in Table 6. Comparing the results obtained by these models, we can see that the performance of EENN in HR is roughly equivalent to that of modified-ELM and superior to other methods. In terms of R^2 and RMSE, EENN significantly outperforms other comparison methods. These results prove that the proposed improved strategies are very effective for improving the accuracy of silicon content prediction.

As we know from Sect. 4.2, for some silicon content prediction models, using all available features for modeling can fully exploit the information contained in the process data and thus improve the prediction accuracy. And the method proposed in this paper tackles the input feature issue with a novel ensemble modeling approach, i.e., an evolutionary algorithm is used to determine the input features and their corresponding weights for each base-learner so that each base-learner has a different combination of input features. This method can enhance the diversity among base-learners while ensuring that the final ensemble model utilizes all available input features, thus further improving the accuracy of silicon content prediction, which is the main reason why it achieves better performance than its competitors.

From a practical application point of view, the EENN proposed in this paper provides a new scheme for building silicon content prediction models that employs an evolutionary algorithm to search the input feature weights of base-learners for the purpose of using all process data properly, avoiding the information wastage problem in PCC and PCA feature selection methods. And more precise silicon content values help on-site operators to get a more accurate indication of the thermal state of the blast furnace. In addition, the proposed model can be automated for modeling without relying on expert experience and easily extended to other industrial application scenarios.

Further Analysis on FWE and FP

Section 4.3.1 presents an analysis of the effect of FWE on the outputs of base-learners, where the threshold of FP

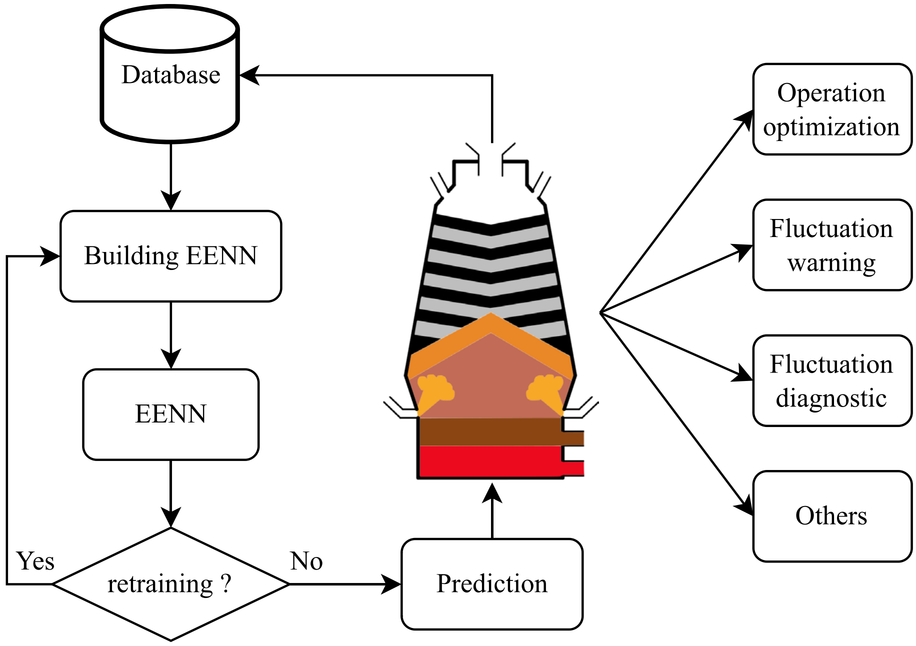


Fig. 11 Online running process of the proposed model

$\alpha = 0.05$. This section focuses on the sparsity generated by FWE and FP in the input of the base-learners, and the performance of obtained ensemble models under different weight thresholds.

In the FWE scenario, the sparsity of the input features of a base-learner (FS) can be defined as follows:

$$FS = \frac{\text{Number of features with a weight of 0}}{\text{Number of features in the original data}} \times 100\% \quad (10)$$

Note that the actual input features of the base-learners do not include these features with a weight of 0, as shown in Fig. 5.

Based on the definition in Equation(10), the feature sparsity of an ensemble model (FSoE) is defined as follows:

$$FSoE = \frac{\sum_{i=1}^{N} FS_i}{N} \quad (11)$$

where N is the number of base-learners involved in the ensemble, and FS_i is the feature sparsity of the i-th base-learner.

The experimental results are shown in Fig. 10, where the prediction accuracy of the model is represented by its RMSE. The RMSE and FSoE of the model under each weight threshold in Fig. 10 are the average results of 30 repeated experiments. The purple polyline in Fig. 10 illustrates the variation of FSoE with the weight threshold α. From the experimental results, it is obvious that even without FP strategy ($\alpha = 1.00e - 10$ in the experiments), the evolution of feature weights can already bring significant sparsity to the ensemble model. As the weight threshold α increases from 0.00 to 0.15, FSoE increases accordingly, which is in good agreement with intuition.

The prediction accuracy of the obtained ensemble model with different weight thresholds is shown by the red polyline in Fig. 10. When the weight threshold is set to 0.05, the prediction accuracy of the ensemble model is significantly improved over that without FP strategy. This result demonstrates the effectiveness of FP strategy. The possible reason is that the FP filters out the features with too small weights, which, after scaling with tiny weights, become almost noise for the base-learners. However, as the threshold continues to increase, the model performance starts to deteriorate, indicating that some useful information is blocked from the learning process by the inappropriately large threshold.

Combining the analysis in Sect. 4.3.1 and Sect. 4.5, it is clear that the role of FWE and the affiliated FP is mainly in two aspects:

1. Some sparsity is brought to the input of the base-learners by evolving and filtering the feature weights. Appropriate sparsity can reduce the structural complexity of the ensemble model and help improve its generalization ability, so as to better cope with the complex and variable blast furnace ironmaking production process.
2. Different combinations of feature weights enhance the diversity among the base-learners. Each combination of input features prompts the corresponding base-learners to mine unique information about the silicon content variation pattern from the process data and plays an

irreplaceable role in the final ensemble model, greatly enhancing the diversity among the base-learners.

Both of the above enhancements to the base-learners are beneficial to reduce the structural risk of the ensemble model and improve its generalization ability.

Applications

In the project in which the research of this paper is carried out, the application of EENN can be simplified to the process shown in Fig. 11. First, the proposed algorithm loads relevant data from the database and builds the prediction model. The obtained model can predict the silicon content of the molten iron during the operation of the blast furnace. The on-site operators can refer to the predicted values to assess the thermal state of the blast furnace and make timely adjustments in case of fluctuating trends. In addition, the model can be automatically updated at a pre-set frequency to match the latest smelting conditions. And the update does not rely on feature selection methods or expert experience.

In addition to helping operators assess the thermal state of the blast furnace, potential applications for the EENN include the following:

1. Automatic fluctuation warning. In combination with condition monitoring methods such as T^2 and SPE [33], a fluctuation warning model can be established for the silicon content of molten iron. When the value of silicon content exceeds the normal range, the model can give an early warning.
2. Operation Optimization. Based on the accurate prediction of silicon content, an operation optimization model of the ironmaking process can be established. When the silicon content needs to be stabilized within a certain range, the operation optimization model can be used to determine the appropriate operating parameter settings.
3. Application to prediction problems of other key indicators. Since the proposed model does not rely on a specific feature selection method or expert experience for modeling, it can be easily applied to the prediction problems of other key indicators, such as blast furnace gas, molten iron temperature, etc.

Conclusion

This paper investigates the silicon content prediction problem in the high vanadium-titanium blast furnace (HVT-BF) smelting process. Experimental results show that the commonly used Pearson correlation coefficient (PCC) and principal component analysis (PCA) feature selection methods do not always perform well in improving prediction accuracy in the HVT-BF scenario, and some existing models achieve relatively high prediction accuracy when modeling with all available features. But PCC and PCA can enhance the robustness of these models to some extent.

In order to further improve the prediction accuracy of silicon content, an ensemble model based on multiobjective optimization is proposed, which contains two main improvement strategies, namely feature weight evolution (FWE) and nonlinear ensemble. And the nonlinear ensemble is based on autoencoder (AE) and extreme learning machine (ELM). Among them, FWE and the affiliated feature pruning (FP) can enhance the diversity of base-learners and alleviate the multicollinearity problem in the output of base-learners; AE-ELM can effectively reduce the redundancy of the meta-learner inputs while fully utilizing the information contained in the outputs of base-learners. The effectiveness of these two strategies and the performance of the overall ensemble model are verified on real-world production data. The superiority of the new scheme of using all available features to build a silicon content prediction model was verified.

Finally, we discuss the effect of FWE and FP on the feature sparsity of the ensemble model. The results show that setting a reasonable weight threshold can improve the performance of the model while reducing its complexity.

Acknowledgements This research was supported by the Major Program of National Natural Science Foundation of China (72192830, 72192831), the Fund for the National Natural Science Foundation of China (62073067, 62303102), the 111 Project (B16009), and the Fundamental Research Funds for the Central Universities (N2128001).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Li J, Hua C, Yang Y, Guan X (2020) Data-driven bayesian-based takagi-sugeno fuzzy modeling for dynamic prediction of hot metal silicon content in blast furnace. IEEE Trans Syst, Man, Cybern: Syst 52(2):1087–1099
2. Saxén H, Pettersson F (2007) Nonlinear prediction of the hot metal silicon content in the blast furnace. ISIJ Int 47(12):1732–1737
3. Zhou H, Zhang H, Yang C (2019) Hybrid-model-based intelligent optimization of ironmaking process. IEEE Trans Ind Electron 67(3):2469–2479
4. Zhou P, Chen W, Yi C, Jiang Z, Yang T, Chai T (2021) Fast just-in-time-learning recursive multi-output lssvr for quality prediction and control of multivariable dynamic systems. Eng Appl Artif Intell 100:104168
5. Niu L, Liu Z, Zhang J, Sun Q, Schenk J, Wang J, Wang Y (2023) Prediction of sinter chemical composition based on ensemble learning algorithms. J Sustain Metall 9(3):1168–1179

6. Wang X, Wang Y, Tang L (2022) Strip hardness prediction in continuous annealing using multiobjective sparse nonlinear ensemble learning with evolutionary feature selection. IEEE Trans Autom Sci Eng 19(3):2397–2411
7. Tang L, Liu C, Liu J, Wang X (2020) An estimation of distribution algorithm with resampling and local improvement for an operation optimization problem in steelmaking process. IEEE Trans Syst, Man, Cybern: Syst. https://doi.org/10.1109/TSMC.2019.2962880
8. Liu Q, Wu J, Shao Y, Wang H, Zhu X, Liao Q (2023) Ann-based model to predict the viscosity of molten blast furnace slag at high temperatures of> 1600 k. J Sustain Metall 9:1020–1032
9. Zhao X, Fang Y, Liu L, Xu M, Zhang P (2020) Ameliorated moth-flame algorithm and its application for modeling of silicon content in liquid iron of blast furnace based fast learning network. Appl Soft Comput 94:106418
10. Yang Y, Zhang S, Yin Y (2016) A modified elm algorithm for the prediction of silicon content in hot metal. Neural Comput Appl 27(1):241–247
11. Hua C, Wu J, Li J, Guan X (2017) Silicon content prediction and industrial analysis on blast furnace using support vector regression combined with clustering algorithms. Neural Comput Appl 28(12):4111–4121
12. Zhou P, Yuan M, Wang H, Wang Z, Chai T-Y (2015) Multivariable dynamic modeling for molten iron quality using online sequential random vector functional-link networks with self-feedback connections. Inf Sci 325:237–255
13. Fontes DOL, Vasconcelos LGS, Brito RP (2020) Blast furnace hot metal temperature and silicon content prediction using soft sensor based on fuzzy c-means and exogenous nonlinear autoregressive models. Comput Chem Eng 141:107028
14. Zhou P, Guo D, Wang H, Chai T (2018) Data-driven robust m-ls-svr-based narx modeling for estimation and control of molten iron quality indices in blast furnace ironmaking. IEEE Trans Neural Netw Learn Syst 29(9):4007–4021
15. Saxén H, Pettersson F, Gunturu K (2007) Evolving nonlinear time-series models of the hot metal silicon content in the blast furnace. Mater Manuf Process 22(5):577–584
16. Tang X, Zhuang L, Jiang C (2009) Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization. Exp Syst Appl 36(9):11853–11857
17. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
18. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
19. Das S, Suganthan PN (2010) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evolut Comput 15(1):4–31
20. Wang X, Dong Z, Tang L (2020) Multiobjective differential evolution with personal archive and biased self-adaptive mutation selection. IEEE Trans Syst, Man, and Cybern: Syst 50(12):5338–5350
21. Wang X, Dong Z, Tang L (2023) Multiobjective multitask optimization - neighborhood as a bridge for knowledge transfer. IEEE Trans Evolut Comput 27(1):155–169
22. Zhen H, Gong W, Wang L (2022) Evolutionary sampling agent for expensive problems. IEEE Trans Evolut Comput 27(3):716–727
23. Cheng S, Ma L, Lu H, Lei X, Shi Y (2021) Evolutionary computation for solving search-based data analytics problems. Artif Intell Rev 54(2):1321–1348
24. Liang J, Chen G, Qu B, Yue C, Yu K, Qiao K (2021) Niche-based cooperative co-evolutionary ensemble neural network for classification. Appl Soft Comput 113:107951
25. Zhao J, Jiao L, Xia S, Fernandes VB, Yevseyeva I, Zhou Y, Emmerich MT (2018) Multiobjective sparse ensemble learning by means of evolutionary algorithms. Decis Support Syst 111:86–100
26. Liu Y, Yao X (1999) Ensemble learning via negative correlation. Neural Netw 12(10):1399–1404
27. Chandra A, Yao X (2006) Ensemble learning using multi-objective evolutionary algorithms. J Math Model Algorithms 5(4):417–445
28. Rosales-Pérez A, García S, Gonzalez JA, Coello Coello CA, Herrera F (2017) An evolutionary multiobjective model and instance selection for support vector machines with pareto-based ensembles. IEEE Trans Evolut Comput 21(6):863–877
29. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evolut Comput 6(2):182–197
30. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
31. Sheng W, Shan P, Chen S, Liu Y, Alsaadi FE (2017) A niching evolutionary algorithm with adaptive negative correlation learning for neural network ensemble. Neurocomputing 247:173–182
32. Craney TA, Surles JG (2002) Model-dependent variance inflation factor cutoff values. Qual Eng 14(3):391–403
33. Zhou P, Zhang R, Liang M, Fu J, Wang H, Chai T (2020) Fault identification for quality monitoring of molten iron in blast furnace ironmaking based on kpls with improved contribution rate. Control Eng Pract 97:104354

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Tenghui Hu[1,4] · **Xianpeng Wang**[2,3] · **Xiangman Song**[1]

✉ Xianpeng Wang
wangxianpeng@ise.neu.edu.cn

[1] Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang 110819, China

[2] Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry, Shenyang 110819, China

[3] Liaoning Key Laboratory of Manufacturing System and Logistics Optimization, Shenyang 110819, China

[4] Key Laboratory of Data Analytics and Optimization for Smart Industry (Northeastern University), Ministry of Education, Shenyang, China