

Predicting Temperature of Molten Steel in LF-Refining Process Using IF–ZCA–DNN Model



ZI-CHENG XIN, JIANG-SHAN ZHANG, JUN-GUO ZHANG, JIN ZHENG, YU JIN,
and QING LIU

Controlling the temperature of molten steel in ladle furnace (LF)-refining process is one of the main tasks to ensure that the steelmaking-continuous casting process runs smoothly. In this work, a hybrid model based on metallurgical mechanism, isolation forest (IF), zero-phase component analysis whitening (ZCA whitening), and a deep neural network (DNN) was established to predict the temperature of molten steel in LF-refining process. The metallurgical mechanism, Pearson correlation coefficient, ZCA whitening, IF, and *t*-distributed stochastic neighbor embedding (*t*-SNE) were used to obtain the main factors affecting the temperature, analyze the correlation between two random variables, eliminate the correlation among the input variables, reduce the abnormal data of the original datasets, and visualize high-dimensional data, respectively. The single-machine-learning (ML) models, ZCA–ML models, and IF–ZCA–DNN model were comparatively examined by evaluating the coefficient of determination (R^2), root-mean-square error (RMSE), mean absolute error (MAE), and hit ratio. The optimal structure of IF–ZCA–DNN model had 4 hidden layers, 45 hidden layer neurons, a learning rate of 0.03, regularization coefficient of 2×10^{-4} , batch size of 128, leaky-rectified linear unit activation function, and an optimization algorithm of mini-batch stochastic gradient descent with momentum. The R^2 , RMSE, and MAE of the IF–ZCA–DNN model were 0.916, 2.827, and 2.048, respectively. Meanwhile, the prediction hit ratio for the temperature of IF–ZCA–DNN model in the error ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$ were 77.9, 92.3, and 99.6 pct, respectively. This study will be beneficial to realize precise control of temperature of molten steel in LF-refining process.

<https://doi.org/10.1007/s11663-023-02753-0>

© The Minerals, Metals & Materials Society and ASM International 2023

I. INTRODUCTION

WITH the technological advancement of the steel industry, greenization and intelligence have become the main development direction of steelmaking plants.^[1] Establishing reasonable control over the molten steel temperature in ladle furnace (LF) refining is key to ensuring the smooth running of the continuous casting process^[2]. However, LF refining is a complicated metallurgical reaction process at high temperature. At present, the temperature of molten steel is controlled mainly by multiple temperature measurements and the

experience of the operators, which leads to low precision for temperature control of molten steel and even affects the smooth running of the continuous casting process. Therefore, the establishment of a reasonable temperature prediction model in LF refining has become the research focus for metallurgists to realize a “narrow window” control of temperature of molten steel.

Among the research on the temperature prediction of molten steel in LF-refining process, three strategies can be developed to establish the temperature prediction model. The first strategy establishes a mechanistic model using the metallurgical mechanism. Camdali and Tunc^[3] established a mathematical mode of heat loss in the LF unit based on a heat transfer analysis and suggested some strategies to decrease heat losses. Volkova and Janke^[4] proposed a mathematical model of ladle heat transfer based on the heat transfer mechanism and Fourier differential equations, and the calculated results were in good agreement with the measured results. Wu *et al.*^[5] derived a heating rate model of molten steel based on the energy balance. However, such models are mostly used in laboratory research, or many parameters of the mechanism model are obtained through field experience due to

ZI-CHENG XIN, JIANG-SHAN ZHANG, and QING LIU are with the State Key Laboratory of Advanced Metallurgy, University of Science and Technology Beijing, Beijing, 86, P.R. China. Contact e-mails: zjsustb@163.com; qliu@ustb.edu.cn JUN-GUO ZHANG, JIN ZHENG, and YU JIN are with the Tangshan Branch, Hebei Iron and Steel Co., Ltd., Tangshan, 86, Hebei, P.R. China.

Manuscript submitted December 15, 2022; accepted February 16, 2023.

Article published online March 14, 2023.

complex field conditions, which leads to some limitations in the practical application of these models. With the rapid development of machine-learning (ML) technology, its application in the field of metallurgy advances rapidly. Hong *et al.*^[6] developed an automatic detection system based on a convolutional recurrent neural network model to detect whether the exit hole of a basic oxygen furnace was blocked during the tapping operation. Wang *et al.*^[7] established a calcium yield prediction model based on an artificial neural network (ANN) during calcium treatment process in steelmaking, which improved the calcium yield and the stability of calcium content. Yang *et al.*^[8] developed a hydrogen content prediction model based on thermodynamics, kinetics, and deep neural network (DNN) for the vacuum oxygen decarburization process. Myers and Nakagaki^[9] proposed a DNN-based model to predict the nucleation lag time for iron and steelmaking slags; this model could rapidly design, analyze, and optimize novel slag compositions. Thakur *et al.*^[10] established DNN-based integrated flow stress and roll force models to accurately predict the plate-rolling of microalloyed steel. Kwon *et al.*^[11] developed an ANN-based model to predict the high-temperature ductility of steels, which can guide the setting of steel-casting operating conditions to ensure that steel maintains high ductility. The second strategy involves the use of ML algorithms to establish a data-driven model to predict the temperature of molten steel. For example, Yang *et al.*^[12] developed a model for LF–VD–CC route using the neural network method. In this case, the prediction hit ratio in the error ranges of 0 °C to 7 °C was 85 pct. Li *et al.*^[13] established a model in LF-refining process based on a genetic algorithm, particle swarm optimization, and a back propagation (BP) neural network algorithm. Wang *et al.*^[14] used partial least squares and support vector machines, obtaining prediction hit ratios of 85 and 95 pct for the temperature in the error ranges of [– 7, 7] and [– 10, 10], respectively. However, the data-driven model relies excessively on data while ignoring the metallurgical mechanism, which limits its practical application.

The third strategy establishes a hybrid model by considering the metallurgical mechanism and ML algorithms. This modeling strategy overcomes not only the difficulty in determining the key parameters of the mechanism model but also the deficiency of data-driven model, so it has been widely used to predict the temperature of molten steel in LF-refining process.^[15–18] For example, Fu *et al.*^[19] established a temperature prediction model based on the mechanism model and BP neural network model to predict the temperature of molten steel in LF-refining process, obtaining a prediction hit ratio above 95 pct for the temperature in the error range of [– 5, 5]. Lv *et al.*^[20–22] established a series of temperature prediction models of molten steel in LF based on the combination of optimally pruned bagging and partial linear extreme learning machine (ELM), the partial linear regularization networks combined with sparse representation technique, and the pruning bagging method, respectively. He *et al.*^[23] developed a case-based reasoning (CBR) based on two-step retrieval approach and the correlation-based feature weighting to predict the temperature of molten steel in LF, indicating that the CBR

model was better than the BP neural network model. In another study, Nath *et al.*^[24] developed a model based on physics, material and heat balance, and statistical analysis to predict and control the temperature and composition of molten steel in LF refining. Tian *et al.*^[25–27] established a series of temperature prediction models of molten steel in LF based on the metallurgical mechanism combined with the ELM and AdaBoost.RT algorithm, the combination of ELM, incremental learning and updating method, and the combination of BP and AdaBoost.IR algorithm, respectively. However, in the aforementioned studies, the number of datasets varied between 200 and 2000, where the majority of the datasets were small. In the case of small datasets, if appropriate measures (*e.g.*, data amplification, data enhancement, regularization) are not taken, over-fitting and reduced generalization of the established model may result. Meanwhile, the choice of ML algorithms has an important influence on the prediction accuracy of ML models. Therefore, taking a comparative approach, we employed a large dataset and several optimized ML algorithms to establish an optimal temperature prediction model for molten steel.

Furthermore, most of the previous research focused on the selection and optimization of the ML algorithms. However, no matter which modeling strategy is used for temperature prediction, the validity and accuracy of the data are also crucial on the basis of sufficient volume of data.^[28] Therefore, data preprocessing, which mainly includes data normalization (*e.g.*, min–max normalization and Z-score standardization), feature transformation, data cleaning, is also essential to constructing the prediction model. When a correlation between the input data exists, it is redundant to directly use the data for training model, which may lead to poor model convergence. Generally, principal component analysis (PCA) is used for data processing in the case of data redundancy.^[29–31] However, when the PCA is used to eliminate the principal components with a small contribution ratio, the integrity of the dataset is easily damaged, even to the point of deleting important information in the sample. As a solution, zero-phase component analysis (ZCA) whitening has been used. ZCA whitening not only reduces the redundancy of the data and removes the correlation between two random variables but also ensures data integrity and tries to make the whitened data close to the original data.^[32] In addition, data cleaning (abnormal data detection) has also been the focus of data-preprocessing research. In general, the clustering algorithm is mainly used to distinguish abnormal data through comparing similarity based on characteristics of outlier data.^[33] However, this algorithm has advantages in identifying global abnormal data. When there is a large amount of data, local abnormal data cannot be effectively identified.^[34] The isolated forest (IF) algorithm^[35] can filter out abnormal data by using a random hyperplane to split the data space. Meanwhile, IF has both high computational efficiency and accuracy with respect to abnormal recognition.

ML algorithms are suitable for solving complex nonlinear industrial problems.^[12–14,20–22] Meanwhile, the influence of metallurgical mechanism and data quality on the prediction accuracy of temperature prediction model cannot be ignored. In this study, an

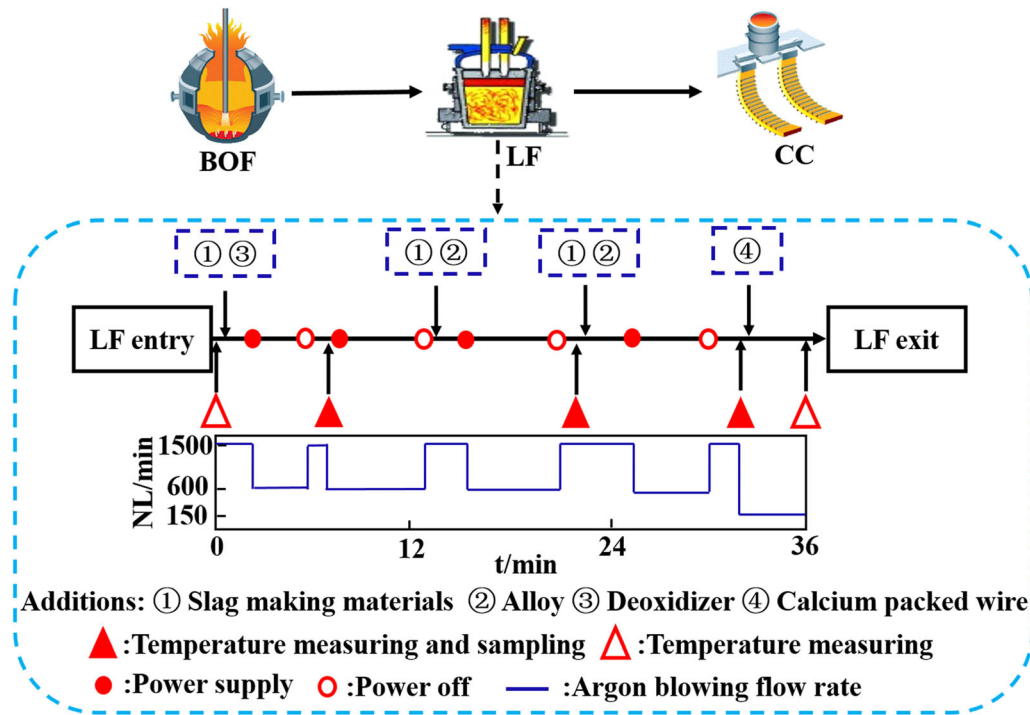


Fig. 1—The whole flow chart of LF-refining process (Adapted with permission from Ref. 36. Copyright 2021, copyright Taylor & Francis).

optimal hybrid model was established to predict the temperature of molten steel in LF-refining process using the metallurgical mechanism, Pearson correlation coefficient, ZCA whitening, IF, and ML algorithms. Firstly, the main factors influencing the temperature of molten steel were obtained by analyzing the metallurgical mechanism and LF-refining process. Then, temperature prediction models of molten steel based on different ML algorithms were proposed. Moreover, correlations between input variables were analyzed by conducting a correlation analysis, and the redundancy of data was eliminated using the ZCA whitening. In addition, ZCA-ML models were established, and their performance was compared with the ML models. Finally, to further improve the prediction accuracy of the ZCA-DNN model, IF was carried out to filter out the abnormal data. This proposed modeling method not only serves as a good reference for the construction of temperature prediction models of molten steel in LF-refining process, but can also be applied to other metallurgical processes, such as sintering, ironmaking, steelmaking, and continuous casting.

II. ANALYSIS OF LF-REFINING PROCESS

LF refining is a complex metallurgical reaction process that involves high-temperature conditions and different operations. The LF-refining process includes temperature measuring, sampling, slagging, adjusting and homogenizing the temperature and composition of molten steel, feeding calcium packed wire, and argon blowing during the whole LF-refining process (different operations corresponding to different argon blowing

intensity, *i.e.*, various additions adding and after molten steel heating corresponding to strong stirring; during heating of molten steel corresponding to medium stirring; soft blowing corresponding to weak stirring). Figure 1 shows a flow chart depicting the entire LF-refining process for a steelmaking plant in China.

Taking the LF, molten steel, and slag as a research system, the energy input is equal to the energy output in LF-refining process according to the principle of energy conservation. The input energy includes the heat gain of electric arc and the heat effects of various additions. The output energy includes the heat storage of the ladle shell, the heat exchange between LF and the surroundings, the heat loss of the bottom blowing argon and slag surface, and the heat required to heat the molten steel. Based on the analysis of the LF-refining process and energy conservation, the heat gain of the electric arc was related to the electric arc-heating time; the temperature change caused by adding alloy and slag-making materials into the molten steel can be calculated according to the temperature effect coefficients of different additions obtained by using statistical analysis. The heat storage of the ladle shell was related to the turnover time of the ladle; the heat exchange between the LF and the surroundings and the heat loss from the slag surface increased with the increase of refining time; and the heat loss of the bottom blowing argon was reflected by the argon consumption. In addition, the initial energy of the molten steel was determined by its initial temperature and weight. To sum up, the main factors affecting the end point temperature of the molten steel in LF refining are as follows: initial temperature of molten steel, turnover time of ladle, refining time, weight of molten steel, heating time, argon consumption, and addition

Table I. Temperature Effect Coefficients When Charging 150 t LF with Various Additions

Addition	Temperature Effect Coefficient $\times 10^{-2}$ ($^{\circ}\text{C}/\text{kg}$)	Addition	Temperature Effect Coefficient $\times 10^{-2}$ ($^{\circ}\text{C}/\text{kg}$)
C	- 4.30	slag-making materials	- 2.00
MCFeMn	- 1.20	FeSi	+ 1.10
HCFeMn	- 1.30	FeTi	- 0.38
Al	- 0.5	CaSi	- 1.10

Note MCFeMn represents the medium carbon ferromanganese; HCFeMn represents the high carbon ferromanganese.

Table II. Descriptive Statistics of All Input and Output Variables for Temperature Prediction Models

Variables	Description of Variables	Units	Minimum	Maximum	Mean	Standard Deviation
X_1	turnover time of ladle	min	22	90	64.35	13.62
X_2	weight of molten steel	t	127.42	160.00	152.97	4.22
X_3	initial temperature of molten steel	$^{\circ}\text{C}$	1516	1615	1561.05	19.52
X_4	refining time	min	15	90	38.48	9.86
X_5	heating time	s	155	1773	784.71	239.67
X_6	argon consumption $\times 10^4$	NL	1.0	6.9	2.42	0.72
$\Delta T_{\text{addition}}$	temperature change caused by additions	$^{\circ}\text{C}$	- 60.53	1.08	- 16.69	9.58
T_{measured}	measured temperature of molten steel	$^{\circ}\text{C}$	1545	1620	1587.17	13.20
T_1	$T_{\text{measured}} - \Delta T_{\text{addition}}$	$^{\circ}\text{C}$	1576.01	1665.95	1603.86	10.70

amounts of various additions. Finally, taking the main factors affecting the temperature of the molten steel as input variables of the ML model, a hybrid prediction model of molten steel temperature was established. The temperature effect coefficients of the alloy and slag-making materials to molten steel are listed in Table I. The calculation formulas of the temperature change ($\Delta T_{\text{addition}}$) caused by additions and the T_1 are shown in Eqs. [1] and [2], respectively.

$$\Delta T_{\text{addition}} = \sum_i G_i q_i, \quad [1]$$

$$T_1 = T_{\text{measured}} - \Delta T_{\text{addition}}, \quad [2]$$

where i expresses addition i (alloy or slag-making materials); G_i represents the weight of addition i (kg); q_i represents the temperature effect coefficient of addition i ($^{\circ}\text{C}/\text{kg}$); T_{measured} is the measured temperature of molten steel (end point temperature of molten steel in LF refining).

The production data, about 9764 heats, were collected from an industrial LF-refining process for steel production in China. Table II summarizes the statistical analysis of all input and output variables for temperature prediction models, including minimum, maximum, mean, and standard deviation. The turnover time of ladle ranged from 22 to 90 minutes. The range of the weight of molten steel was 127.42 t to 160.00 t. The range of initial temperature of molten steel was from 1516 $^{\circ}\text{C}$ to 1615 $^{\circ}\text{C}$. The value of refining time ranged from 15 to 90 minutes. The heating time ranged from 155 to 1773 seconds. The argon consumption ranged from 1.0×10^4 NL to 6.9×10^4 NL. The $\Delta T_{\text{addition}}$

ranged from - 60.53 $^{\circ}\text{C}$ to 1.08 $^{\circ}\text{C}$. The range of the T_{measured} was 1545 $^{\circ}\text{C}$ to 1620 $^{\circ}\text{C}$. The T_1 ranged from 1576.01 $^{\circ}\text{C}$ to 1665.95 $^{\circ}\text{C}$.

III. TEMPERATURE PREDICTION OF MOLTEN STEEL USING ML MODELS

This section introduces the temperature prediction model construction method (Section III-A), data-processing methods (Section III-B), the various ML algorithms (Section III-C), and the model evaluation methods (Section III-D) in predicting the temperature of molten steel in LF-refining process.

A. Modeling with ML Models

The temperature prediction models based on the metallurgical mechanism, data-preprocessing methods, and ML algorithms were developed by analyzing the LF-refining process. This study consists of three main parts. Figure 2 presents a schematic diagram illustrating the design process for the prediction model, where each part involves the follows:

Part 1 the temperature prediction models of molten steel were established based on the production data and different single ML algorithms (e.g., DNN; K-Nearest Neighbor, KNN; Regularized ELM, RELM; Bayesian-regularization BP neural network, BR-BP). Then, 10-fold cross-validation^[37] was used to optimize the parameters of different ML models, and the optimal parameters of different ML models were obtained. Finally, the performance of the ML models was evaluated in terms of the coefficient of determination (R^2), mean absolute error (MAE), root-mean-square error (RMSE), and hit ratio.

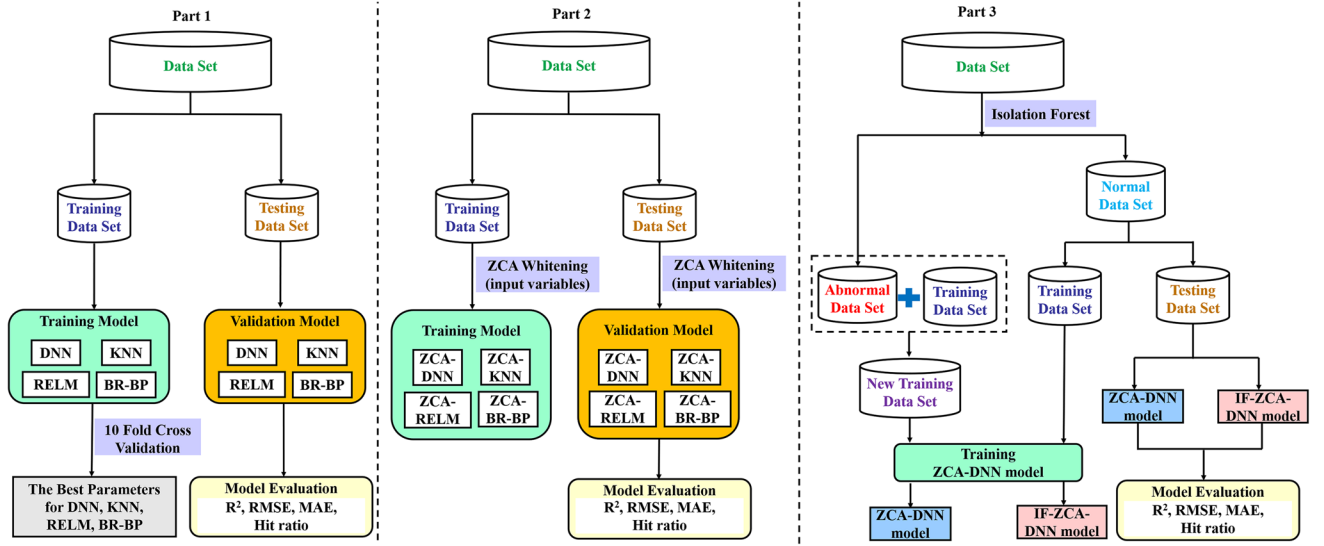


Fig. 2—Schematic diagram of the design method and model validation.

Part 2 when the input variables are correlated, it is redundant to directly use the original data for the training model. In this study, ZCA whitening was applied to eliminate correlation and redundancy between input variables. Then, the performance of the ML models was compared with that of the ZCA-ML models using the same test set.

Part 3 abnormal value may affect the prediction results of the ML models. The IF algorithm was used to detect the abnormal value, and then the training dataset with and without the abnormal value were used for model training. Finally, the performance of the ZCA-DNN model was compared with that of the IF-ZCA-DNN model using the same test set.

B. Data-Processing Methods

Data analysis and data processing are the primary tasks of modeling. This section mainly introduces the visualization method for the high-dimensional data (Section III-B-1), processing method for data redundancy (Section III-B-2), and processing method for abnormal data (Section III-B-3). The data were processed using the above methods to obtain high-quality data, which gave full play to the value of data.

1. *t*-Distributed stochastic neighbor embedding (*t*-SNE)

It is difficult for us to directly observe the distribution of high-dimensional data. In general, the method of dimensionality reduction is used to visualize the high-dimensional data and then to analyze it. The *t*-SNE algorithm, an advanced nonlinear dimensionality reduction and visualization algorithm, was proposed by Maaten and Hinton.^[38,39] The *t*-SNE was mainly used to visualize the high-dimensional data in two- and three-dimensional space, so as to reveal hidden information in the data.

The process of *t*-SNE was shown as follows:^[40]

Step 1 calculating the conditional probability density (p_{ji}) between any two data points x_i and x_j in high-dimensional dataset, as shown in Eq. [3].

$$p_{ji} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}, \quad [3]$$

where x_k refers to other data points except x_i in the high-dimensional dataset; σ_i represents the variance of the Gaussian distribution of data point x_i .

Step 2 calculating the joint probability density (p_{ij}) of the high-dimensional sample, as shown in Eq. [4].

$$p_{ij} = (p_{ji} + p_{ij}) / 2n, \quad [4]$$

where n is the number of data points.

Step 3 initializing low-dimensional data ($Y^{(0)}$) randomly, as shown in Eq. [5].

$$Y^{(0)} = \{y_1, y_2, \dots, y_n\}. \quad [5]$$

Step 4 the joint density (q_{ij}) and gradient ($\frac{\partial C}{\partial y_i}$) of sample points in low-dimensional space were calculated by t distribution with degrees of freedom 1, as shown in Eqs. [6] and [7].

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_k - y_l\|^2)^{-1}}, \quad [6]$$

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}, \quad [7]$$

where C is the cost function defined by Kullback-Leibler distance

Step 5 calculating the low-dimensional data ($Y^{(l)}$), as shown in Eq. [8].

$$Y^{(t)} = Y^{(t-1)} + \eta \frac{\partial C}{\partial Y} + \mu(t)(Y^{(t-1)} - Y^{(t-2)}), \quad [8]$$

where t is iterations; η is learning rate; $\mu(t)$ is momentum factor.

Step 6 repeating steps 4–5 until the number of iterations met the requirements.

2. Zero-phase component analysis whitening (ZCA whitening)

When a correlation exists between the input data, it is redundant to directly use the data for the training model, as it will lead to poor model convergence. Pearson correlation analysis^[41] was used to analyze the correlation between two random variables, and the formula is shown in Eq. [9]. ZCA whitening was used to eliminate the data redundancy of the input variables. Meanwhile, the features were less correlated with each other, and the features all had the same variance after whitening.^[32]

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad [9]$$

where \bar{x} is the mean of variable x ; \bar{y} is the mean of variable y ; x_i is the i th value of variable x ; y_i is the i th value of variable y .

The process of ZCA whitening was shown as follows:

Step 1 calculating the covariance matrix (Σ) of the dataset, as shown in Eq. [10].

$$\Sigma = \frac{1}{m} \sum_{i=1}^m (x^{(i)})(x^{(i)})^T, \quad [10]$$

where m is the number of samples; $x^{(i)}$ is the vector composed by physical layer parameter.

Step 2 after calculating the covariance matrix of the dataset, the vector (U) is obtained by singular value decomposition (SVD). Then, $U^T x$ is used to obtain x_{rot} , as shown in Eq. [11].

$$x_{\text{rot}} = U^T x = \begin{bmatrix} u_1^T x \\ u_2^T x \\ \vdots \\ u_n^T x \end{bmatrix}, \quad [11]$$

where $u_n^T x$ is the projection amplitude of sample point x on feature u_n .

Step 3 the dataset was whitened using PCA, as shown in Eq. [12].

$$x_{\text{PCA},i} = \frac{x_{\text{rot},i}}{\sqrt{\lambda_i + \varepsilon}}, \quad [12]$$

where λ_i is the value of the diagonal element in the covariance matrix (x_{rot}). During the ZCA whitening process, some eigenvalues may be close to zero, so the scaling process was divided by the values close to 0. Therefore, the regularization (ε) was used to achieve the scaling process in practical application. Generally, the value of ε is 1×10^{-5} .

Step 4 the vector (U) was multiplied by PCA whitening matrix to obtain the ZCA whitening matrix, as shown in Eq. [13].

$$x_{\text{ZCA},i} = U x_{\text{PCA},i}. \quad [13]$$

3. Isolation forest (IF)

The IF algorithm^[35,42] was used to detect the outlier in the dataset. The data space was continuously split using a random hyperplane until every data in the data space became a data node and formed an isolated Tree (iTree). Then, the path length $h(i)$ between the root node and the i th data node was used to determine whether the data node (i) was an outlier.

The process of IF algorithm was shown as follows:

Step 1 samples (n) were randomly selected from the dataset (X) as training sub-dataset (X') and put into the root node of iTree.

Step 2 a variable dimension was randomly selected from the variables, and a break point K was randomly generated on this dimension. Meanwhile, the break point K was between the maximum and minimum value of the selected dimension.

Step 3 the break point K was extended into hyperplane, and the space of dataset was divided into two spaces. And then, dividing values less than K to the left branch and greater than K to the right branch.

Step 4 repeating steps 2 and 3 to continuously split data space using hyperplane until every data in the data space became a data node (iTree) or iTree reached a height limit.

Step 5 the iForest training iteration stopped to obtain t iTrees. Then the generated iForest was used to calculate the outlier score of the data, so as to detect whether the data were outliers.

C. ML Algorithms

After obtaining a dataset, it is necessary to select appropriate ML algorithms. This study, which has both input and output values, corresponds to supervised learning, and needs regression algorithms to predict the temperature of molten steel in LF-refining process. To predict the temperature of molten steel, five ML algorithms were compared in this study, including (i) Bayesian-regularization BP neural network (BR-BP), (ii) RELM, (iii) KNN, and (iv) DNN. The descriptions of these algorithms were shown as follows.

1. Bayesian-regularization BP neural network (BR-BP)

The BP neural network is a multilayer feedforward network with error BP and has strong nonlinear mapping ability.^[43] The traditional BP neural network is prone to over-fitting, which reduces the generalization ability of the network. However, Bayesian regularization, which imparts high generalization ability to the network by improving the objective function, can solve these problems. The mean-square error was used as the objective function in the traditional BP neural network,^[44] as shown in Eq. [14]. The penalty function was

introduced in the objective function of the traditional BP neural network to construct the modified objective function of the BR–BP neural network,^[45] as shown in Eqs. [15] and [16]. In order to ensure the rationality of parameters α and β , the Bayesian regularization algorithm can adaptively adjust the values of α and β to achieve optimal results and improve the generalization ability of the network according to the training results of the network.

$$E_d = \frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2, \quad [14]$$

where n is the number of samples; t_i is the actual value; y_i is the output value of the neural network.

$$J_w = \alpha E_w + \beta E_d, \quad [15]$$

$$E_w = \frac{1}{m} \sum_{j=1}^m w_j^2, \quad [16]$$

where α and β are regularization coefficients; E_w is the mean-square error of all network weights; m is the total number of network weights; w_j is the network weight.

2. Regularized extreme learning machine (RELM)

Huang *et al.*^[46] proposed a single-hidden layer feed-forward neural network algorithm called ELM, which has fast computation speed and does not require any iterative adjustment for parameter determination. In the traditional ELM model,^[47] based on the principle of empirical risk minimization, the minimum training error is taken as the criterion, which does not consider the structural risk. It is easy to overfit when there are too many nodes in hidden layer.^[48] In order to solve the

deficiencies of the ELM model, the structural risk and regularization coefficient were introduced to construct the RELM model. The regularization coefficient was applied to adjust the proportion of empirical risk and structural risk to improve the generalization ability of the traditional ELM model.^[49–51]

3. *K*-nearest neighbor (KNN)

KNN is a typical supervised learning algorithm^[52] based on the core idea that if a sample has the smallest distance from K samples in the feature space, the sample value is the arithmetic mean of K samples. The only hyperparameter of the KNN algorithm is K . If the K value is too small, it is easy to lead to over-fitting of the KNN model; if the K value is too large, it is easy to lead to under-fitting of the KNN model. In general, the value of K is an integer not greater than 20.^[53]

The process of KNN algorithm^[53] was shown as follows:

Step 1 inputting the test data after the feature and label values of the training dataset were given.

Step 2 calculating the distance (European distance or Manhattan distance) between the feature value of the test data and the corresponding feature value of the training set,

Step 3 the data were sorted according to the distance, and the first K data for the feature values of the training set closest to the feature values of the test set were selected.

Step 4 the average value of the label values corresponding to the first K data was the predicted value corresponding to the test data.

4. Deep neural network (DNN)

With the advent of the era of big data and ML, DNN has been developed rapidly and has become a key technology in the field of artificial intelligence.^[54] The DNN consists of three layers: input layer, hidden layer

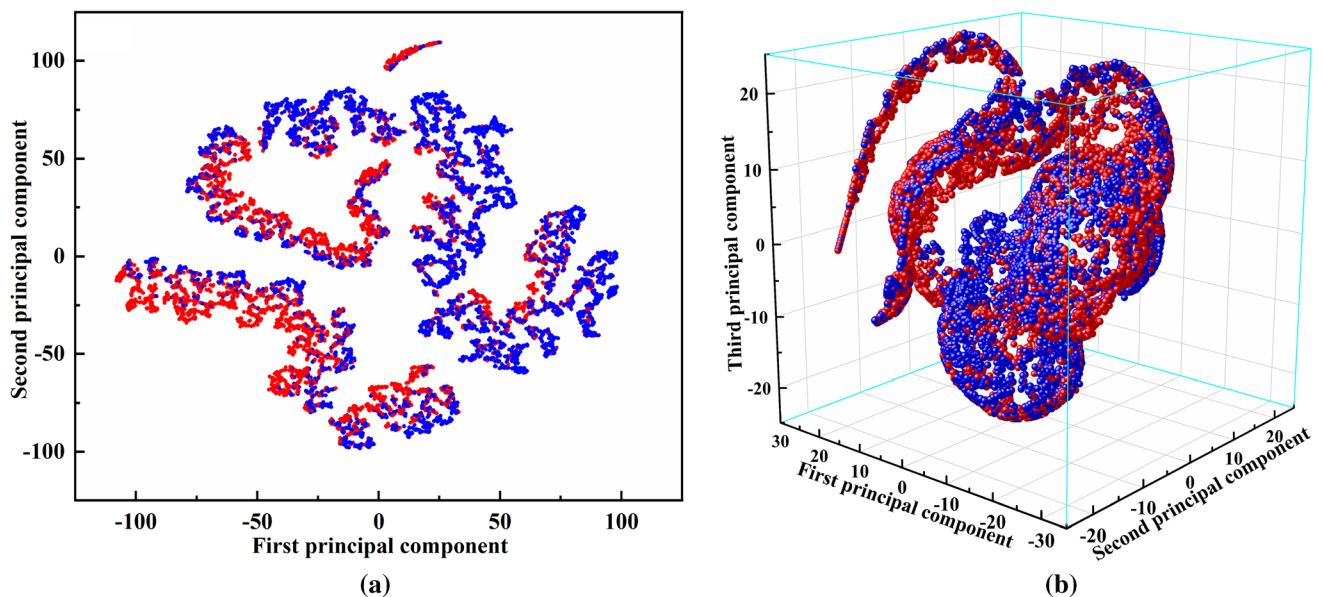


Fig. 3—Visualization of original data in (a) two-dimensional space, and (b) three-dimensional space. Higher than average temperature is represented in red, while lower than average temperature is represented in blue (Color figure online).

(≥ 3), and output layer.^[55] Meanwhile, the adjustment of hyperparameters for DNN is essential, where the main hyperparameters include the activation function, learning rate, number of hidden layers, number of nodes in the hidden layer, batch size, and the optimization algorithms. The activation function is applied to increase the nonlinearity of DNN model. The learning rate is used to determine how fast parameters are updated.^[56] The number of hidden layers and nodes are used to determine the complexity of the network structure and the learning ability of the DNN.^[15] The batch size influences the generalization ability and training time of the DNN model. In our previous paper,^[15] we described the DNN model and the roles of different hyperparameters in detail, which will not be repeated here.

D. Model Evaluation

In order to compare the generalization ability of different ML models, the performances of the ML models were evaluated in terms of the R^2 , MAE, RMSE, and hit ratio, respectively. The “hit ratio” is the proportion of test heats within the error range of temperature to the total test heats. The high R^2 and hit ratio values and the lower MAE and RMSE values indicate better precision of the ML models. The R^2 , MAE, RMSE, and hit ratio values were calculated using the following equations:

$$R^2 = \frac{\sum_{i=1}^{N_p} (y_i^{\text{act}} - \bar{y}_m)^2 - \sum_{i=1}^{N_p} (y_i^{\text{cal}} - y_i^{\text{act}})^2}{\sum_{i=1}^{N_p} (y_i^{\text{act}} - \bar{y}_m)^2}, \quad [17]$$

$$\text{MAE} = \sum_{i=1}^{N_p} |y_i^{\text{cal}} - y_i^{\text{act}}| / N_p, \quad [18]$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N_p} (y_i^{\text{cal}} - y_i^{\text{act}})^2 / N_p}, \quad [19]$$

$$\text{Hit ratio} = \frac{N_{(|\text{Calculated Value} - \text{Actual Value}| \leq \varepsilon)}}{N_{(\text{Total test heats})}} \times 100 \text{ pct}, \quad [20]$$

where N_p is the number of the data; y_i^{act} represents the actual value of the temperature; y_i^{cal} represents the calculated value of the temperature; \bar{y}_m represents the average value of the temperature; $N_{(|\text{Calculated Value} - \text{Actual Value}| \leq \varepsilon)}$ is the number of the test heats within the error range of temperature; $\varepsilon = 3 \text{ }^\circ\text{C}$, $5 \text{ }^\circ\text{C}$, and $10 \text{ }^\circ\text{C}$ in temperature prediction; $N_{(\text{Total test heats})}$ is the number of the total test heats.

IV. RESULTS AND DISCUSSION

A. High-Dimensional Data Visualization

The purpose of LF refining is to produce molten steel that satisfies the temperature, composition, and cleanliness requirements to ensure the smooth running of the continuous casting process within a specified time. LF refining is a complex metallurgical reaction process under high-temperature condition. Based on the analysis of the LF-refining process and energy conservation, seven factors affect the end point temperature of molten steel in LF refining. The original data belong to the high-dimensional data which are difficult to intuitively display the data distribution. Therefore, the original data were visualized in two and three dimensions using the t -SNE algorithm, as shown in Figure 3. The red and blue dots in the figure are intertwined, indicating the complexity of these data structures. This motivates the application of ML algorithms to explore relationships between complex data. Therefore, a variety of ML algorithms were applied to predict the temperature of molten steel.

B. Correlation Analysis and ZCA whitening

Figure 4 displays a heat map of the Pearson correlation coefficients. In Figure 4(a), the Pearson correlation coefficients between the refining time and the argon consumption was 0.417, indicating that the argon consumption increased with increasing the refining time. The Pearson correlation coefficients between the initial temperature of molten steel and the heating time were -0.505 , indicating that the heating time increased with decreasing the initial temperature of molten steel. The Pearson correlation coefficients between the refining time and the heating time was 0.753, indicating that the heating time increased with increasing the refining time. According to the correlation analysis, there are outstanding linear relationships between input variables, which may lead to data redundancy and even result in poor convergence of the prediction models in this study. ZCA whitening not only reduces the data redundancy but also tries to make the whitened data close to the original data. Therefore, the data were processed using ZCA whitening, as shown in Figure 4(b). It can be seen that the correlation coefficients between any two input variables were zero, *i.e.*, the correlation between input variables was eliminated.

C. Comparison of Prediction Performance of Single ML Models

In order to obtain the best parameters for various ML algorithms (DNN, BR-BP, RELM, and KNN), a 10-fold cross-validation^[37] was applied based on the training dataset (8264 heats) and testing dataset (1500 heats). Note that the testing dataset was not included in the learning of the ML model. Table III lists the best parameters for various ML algorithms within the parameter boundaries. In Section III-C, the various ML algorithms were introduced and were mainly

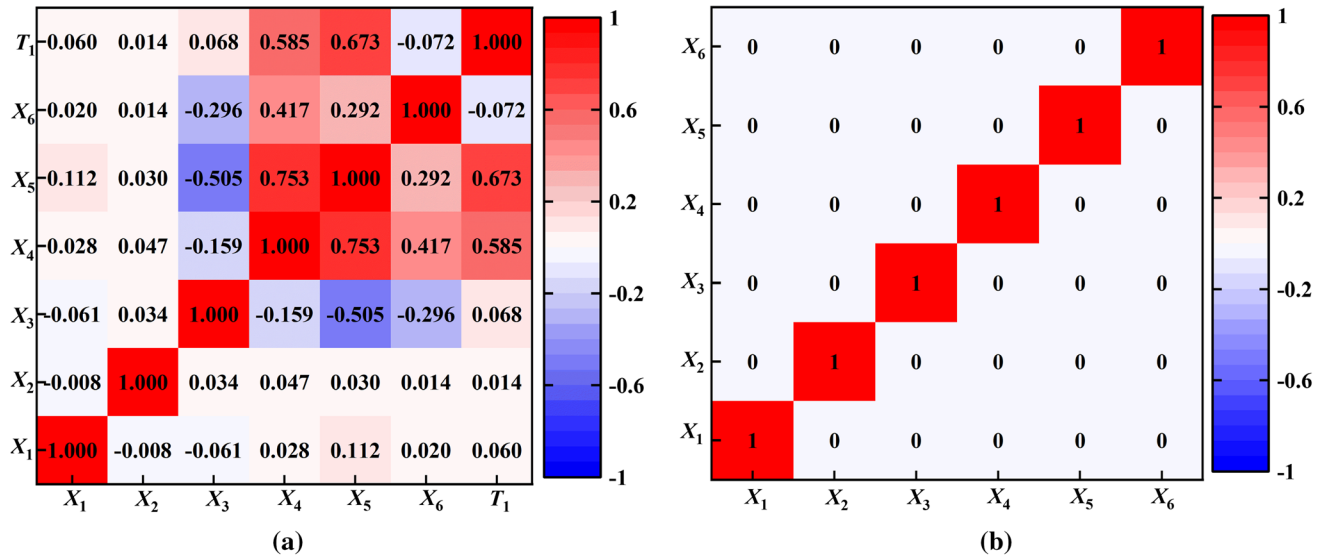


Fig. 4—Heat map of Pearson correlation coefficient of (a) original data, and (b) data of ZCA whitening.

Table III. The Best Parameters for Various ML Algorithms Within the Parameter Boundaries

Method	Parameters Name	Parameters Boundaries	Best Parameters
DNN	num_hidden layer	[3, 6]	4
	num_hidden layer neurons	[20, 60]	45
	learning_rate	(0, 1)	0.03
	L2 regularization coefficient	$[1 \times 10^{-4}, 1 \times 10^{-2}]$	2×10^{-4}
	batch_size	64, 128, 256, 512	128
	activation function	—	LReLU
	optimization algorithms	—	mini-batch SGD with momentum (momentum parameter defaults to 0.9)
KNN	n_neighbors	[3, 20]	10
RELM	num_hidden layer neurons	[10, 500]	300
	regularization coefficient	(0, 1)	0.3
	Activation function	sin, sig, hardlim	sin
BR-BP	num_hidden layer neurons	[3, 13]	10
	learning_rate	(0, 1)	0.2
	training function	—	trainbr

constructed using the scikit-learn module of Python. For the DNN model, the leaky-rectified linear unit (LReLU) was selected as the activation function. The mini-batch stochastic gradient descent (SGD) with momentum and the L2 regularization coefficient of 2×10^{-4} were integrated and utilized to improve the convergence speed and prevent getting into local optimum and over-fitting for the traditional DNN. The integrated optimization algorithm utilized to modify the traditional DNN can be found in our previous work.^[15] Meanwhile, the other optimal hyperparameters are given in Table III. The DNN was configured with four hidden layers, and the number of hidden layer neurons was 45. The learning rate and the batch size were 0.03 and 128, respectively.

In addition, the statistical evaluation indexes (e.g., R^2 , r , RMSE, MAE) and hit ratio were used to evaluate the generalization performance of the ML models. In Figure 5(a), the R^2 and r values of DNN model were 0.026 and 0.014 higher than those of the KNN model, respectively; 0.027 and 0.015 higher than those of the RELM model, respectively; 0.042 and 0.023 higher than those of the BR-BP model, respectively; and 0.137 and 0.078 higher than those of the MLR model, respectively. The RMSE and MAE values of the DNN model were 0.427 and 0.372 lower than those of the KNN model, respectively; 0.370 and 0.337 lower than those of the RELM model, respectively; 0.563 and 0.564 lower than those of the BR-BP model, respectively; and 1.618 and 1.456 lower than those of the MLR model, respectively.

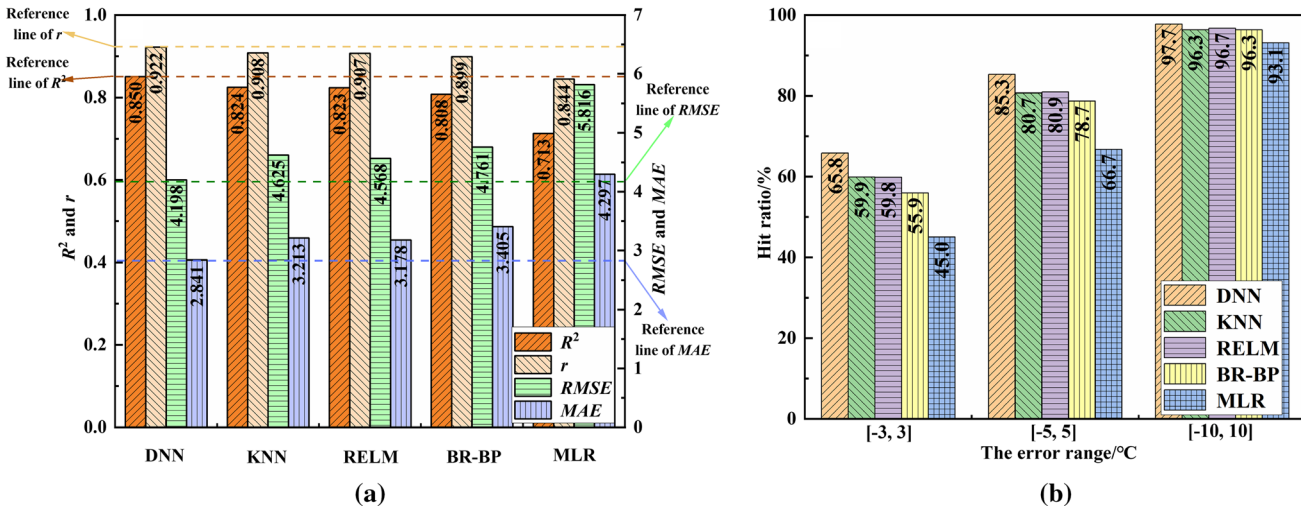


Fig. 5—(a) The statistical evaluation indexes and (b) hit ratio of the various prediction models.

Table IV. Comparison of Various Models

Indices	Various Models									
	DNN		KNN		RELM		BR-BP		MLR	
	INIT	ZCA	INIT	ZCA	INIT	ZCA	INIT	ZCA	INIT	ZCA
R^2	0.850	0.875	0.824	0.852	0.823	0.827	0.808	0.825	0.713	0.713
r	0.922	0.935	0.908	0.923	0.907	0.909	0.899	0.908	0.844	0.844
RMSE	4.198	3.895	4.625	4.258	4.568	4.517	4.761	4.537	5.816	5.816
MAE	2.841	2.650	3.213	2.892	3.178	3.136	3.405	3.227	4.297	4.297

Note INIT and ZCA denote the ML models without and with ZCA whitening, respectively.

In Figure 5(b), the prediction hit ratio of the DNN model for the temperature in the error ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$ were 65.8, 85.3, and 97.7 pct, respectively. The prediction accuracies of the DNN model were 5.9, 4.6, and 1.4 pct higher than those of the KNN model; 6.0, 4.4, and 1.0 pct higher than those of the RELM model; 9.9, 6.6, and 1.4 pct higher than those of the BR-BP model; and 20.8, 18.6, and 4.6 pct higher than those of the MLR model in the ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$, respectively. In summary, the DNN model was superior to the other models in predicting the temperature of molten steel, achieving the highest R^2 value of 0.850, highest r value of 0.922, lowest RMSE of 4.198, and lowest MAE of 2.841. The goodness of fit decreased in the following order: DNN model, KNN model, RELM model, BR-BP model, and MLR model.

D. Comparison of Prediction Performance of ZCA-ML Models

The original industrial data contain amount of redundant, and thus, using the original industrial data to establish the model would lead to poor convergence and low accuracy. In order to further improve the prediction accuracy of the ML models, ZCA whitening was employed for data processing. To evaluate the

impact of ZCA whitening on the ML models, the INIT-ML models (*i.e.*, the ML models without ZCA whitening) and ZCA-ML models (*i.e.*, the ML models with ZCA whitening) were compared according to their statistical evaluation indexes and hit ratios. Table IV lists the performance of the various models. It can be seen that the R^2 and r values of the ZCA-ML models were better than those of the INIT-ML models. The ZCA-DNN model achieved the highest R^2 value of 0.875, highest r value of 0.935, lowest RMSE of 3.895, and lowest MAE of 2.650.

In addition, the R^2 , RMSE, MAE, and hit ratio in different error ranges of the INIT-ML models and ZCA-ML models (the ML models included the DNN, KNN, and RELM models) are illustrated in Figure 6. In Figure 6(a), the R^2 value of the ZCA-DNN model was 0.025 higher than that of the INIT-DNN, and the RMSE and MAE values of the ZCA-DNN model were 0.303 and 0.191 lower than those of the INIT-DNN. The R^2 value of the ZCA-KNN model was 0.028 higher than that of the INIT-KNN, and the RMSE and MAE values of the ZCA-KNN model were 0.367 and 0.321 lower than those of the INIT-KNN. The R^2 value of the ZCA-RELM model was 0.004 higher than that of the INIT-RELM, and the RMSE and MAE values of the ZCA-RELM model were 0.051 and 0.042 lower than those of the INIT-RELM. In Figure 6(b), the

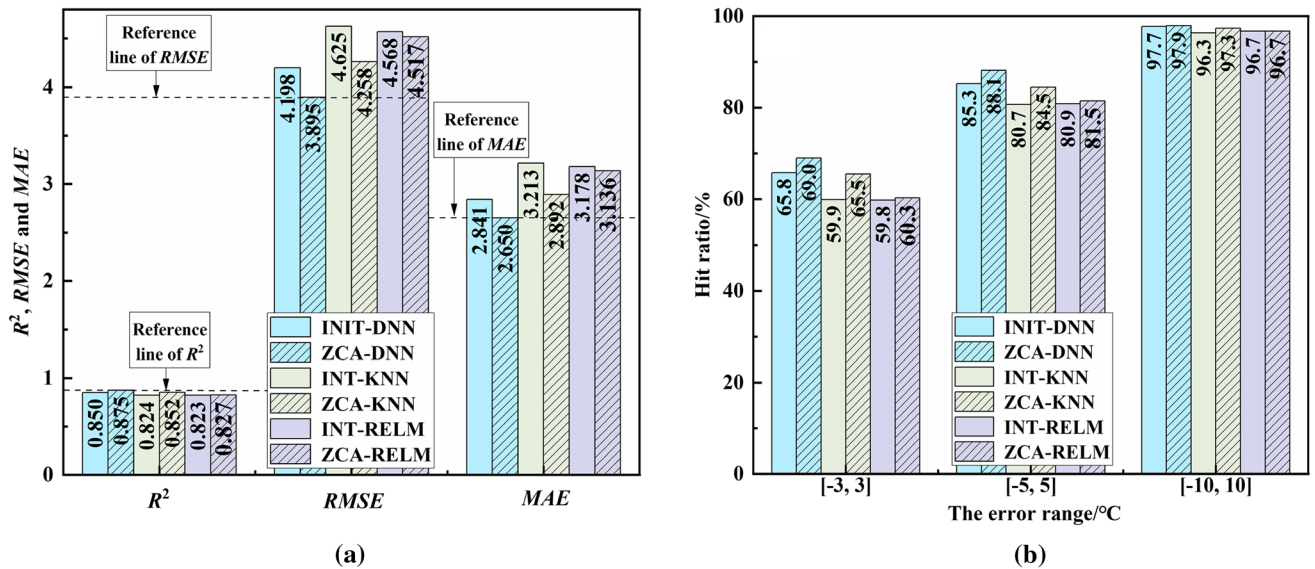


Fig. 6—(a) The statistical evaluation indexes and (b) hit ratios of the best-three models.

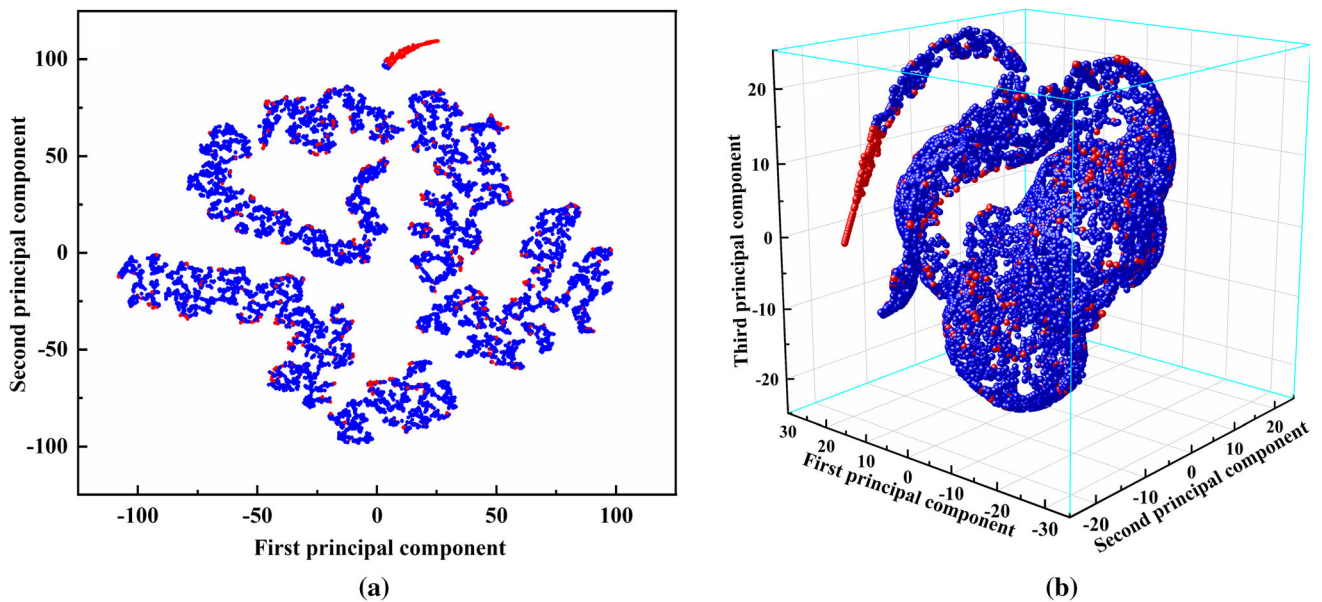


Fig. 7—Visualization of abnormal data and normal data in (a) two-dimensional space, and (b) three-dimensional space (Color figure online).

ZCA-DNN model achieved the highest hit ratios of 69.0, 88.1, and 97.9 pct in the ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$, respectively. Meanwhile, the prediction accuracies of the ZCA-DNN model were 3.2, 2.8, and 0.2 pct higher than those of the INIT-DNN model in the ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$, respectively. The prediction accuracies of the ZCA-KNN model were 5.6, 3.8, and 1.0 pct higher than those of the INIT-KNN model in the ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$, respectively. The prediction accuracies of the ZCA-RELM model were 0.5 pct, 0.6 pct, and 0 higher than those of the INIT-RELM model in the ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$, respectively. In summary, the ZCA-DNN

model exhibits the optimal performance. Meanwhile, for most of the ML models (*e.g.*, DNN, KNN, RELM, BR-BP), the effectiveness of ZCA whitening was reflected in the temperature prediction in LF-refining process.

E. Comparison of Prediction Performance of IF-ZCA-DNN Model

In order to further improve the prediction accuracy of the ZCA-DNN model, the IF algorithm was applied to discover the abnormal values in the dataset, and 802 groups of abnormal values were discovered in the original dataset. The *t*-SNE algorithm was used to

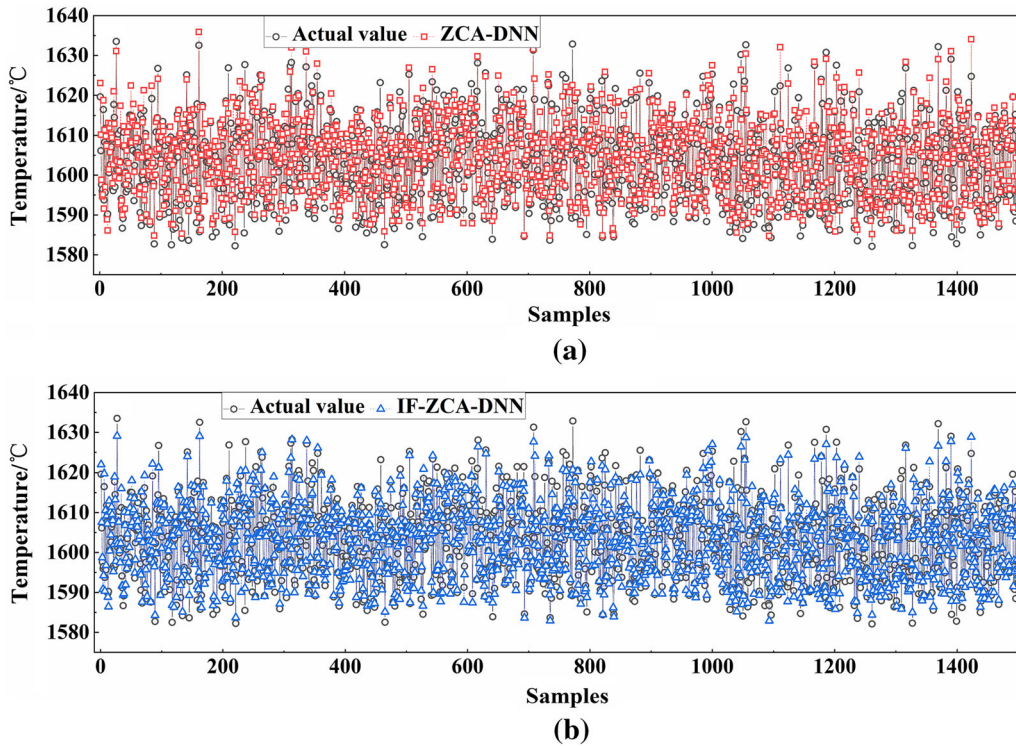


Fig. 8—Comparison between actual and predicted values obtained by (a) ZCA–DNN model and (b) IF–ZCA–DNN model.

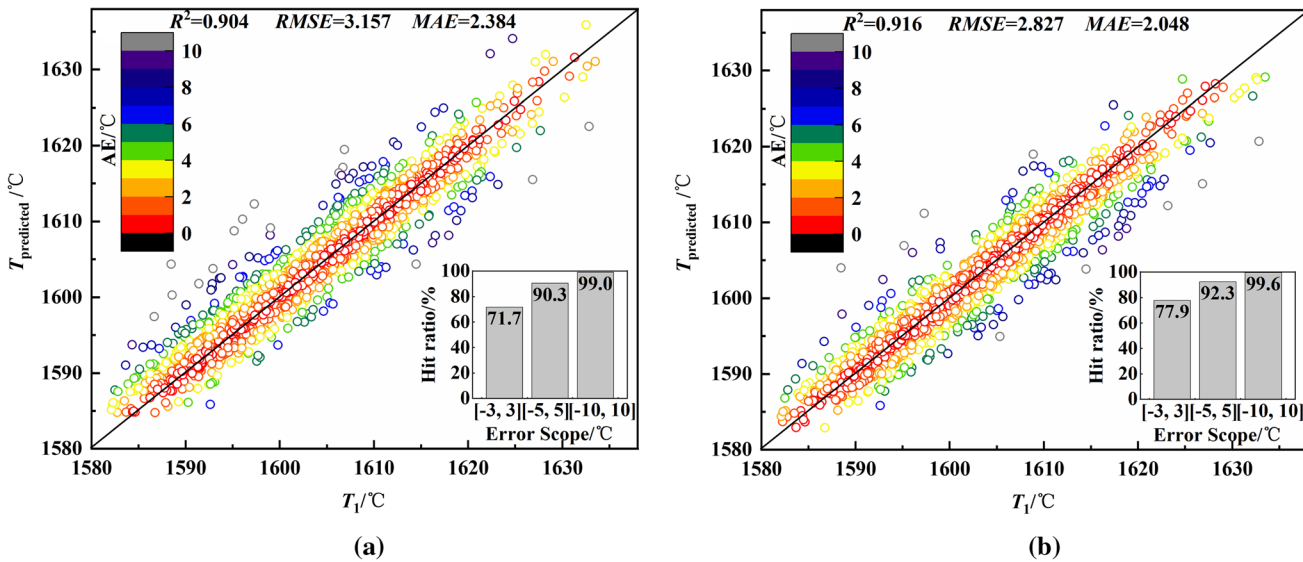


Fig. 9—Comparison between T_1 and $T_{\text{predicted}}$ obtained by (a) ZCA–DNN model and (b) IF–ZCA–DNN model.

visualize the distribution of abnormal data and normal data, as shown in Figure 7. Abnormal data are represented in red, and normal data are represented in blue. In the next section, the training datasets with and without the abnormal values were constructed to train the model, and the ZCA–DNN model was evaluated using the same test dataset.

Figure 8 presents the prediction results of ZCA–DNN and IF–ZCA–DNN models. The abscissa represents the heat number, and the ordinate represents the temperature. Each heat number corresponds to the actual temperature value and the predicted temperature value represented by different colors and symbols. It can be seen from Figure 8 that the calculated temperature values of the ZCA–DNN and IF–ZCA–DNN models fit

well with the actual temperature values. In the next section, other statistical evaluation indexes were used to evaluate the ZCA–DNN and IF–ZCA–DNN models.

To evaluate the prediction accuracy and generalization performance of the ZCA–DNN and IF–ZCA–DNN models more intuitively, scatterplots were drawn, and the R^2 , RMSE, MAE, and hit ratio in different error ranges were calculated, as shown in Figure 9. It can be seen that the RMSE and MAE of the IF–ZCA–DNN model decreased from 3.157 and 2.384 of the ZCA–DNN model to 2.827 and 2.048. The R^2 of the IF–ZCA–DNN model increased to 0.916. In Figure 9, the 45 deg diagonal line was used to express that T_{actual} was equal to $T_{\text{predicted}}$. A color scale was applied to express the different absolute errors (AE) between the actual values and predicted values, and a histogram was used to express the distribution of the error scope, as shown in the graphs in the upper left corner and lower right corner, respectively. The overall scatter plot of the IF–ZCA–DNN model was closer to the 45 deg diagonal line than that of the ZCA–DNN model, which indicated that the error between the actual and predicted values of IF–ZCA–DNN model was smaller than that of the ZCA–DNN model. The prediction hit ratio of the ZCA–DNN model for the temperature in the error ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$ were 71.7, 90.3, and 99.0 pct, respectively. The prediction hit ratio of the IF–ZCA–DNN model for the temperature in the error ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$ were 77.9, 92.3, and 99.6 pct, respectively. The prediction accuracies of the IF–ZCA–DNN model were 6.2, 2.0, and 0.6 pct higher than those of the ZCA–DNN model in the ranges of $[-3, 3]$, $[-5, 5]$, and $[-10, 10]$, respectively. Based on these prediction results, the temperature prediction model established using the IF–ZCA–DNN algorithm was more accurate than that established using the ZCA–DNN algorithm. Therefore, owing to its high accuracy and strong generalization performance, the IF–ZCA–DNN algorithm was the most recommended method for the end-temperature prediction of molten steel in LF-refining process.

This study indicates that the DNN algorithm is superior to KNN, RELM, BR–BP, and MLR in predicting the temperature of molten steel in LF-refining process. Moreover, ZCA whitening is good for improving the prediction performance of most ML models (e.g., DNN, KNN, RELM, BR–BP). Furthermore, the data processing based on the IF algorithm can also improve the prediction accuracy of the ZCA–DNN model. Based on current research results, there is room for further improving the performance of IF–ZCA–DNN model, which is mainly expounded from the following two aspects: (1) in recent years, the explainable ML has become an important research direction of ML. SHapley Additive exPlanation (SHAP),^[57–59] which can explain the output of any ML model, is a model interpretation module developed by Python. Therefore, the SHAP can be applied to improve the interpretability of the DNN model in the next step, further improving the prediction accuracy of the DNN model and realizing a “narrow window” control of molten steel temperature. (2) With the increase of data

volume, the improvement of computational efficiency of the DNN model will be considered on the basis of ensuring the prediction accuracy of the model.^[60] With the rapid development of parallel computing, the calculation speed of the DNN model can be accelerated using a high-performance graphic processing unit (GPU)^[61,62] to meet the current high-efficiency development requirements of steel plants.

V. CONCLUSIONS

In this study, prediction models based on the metallurgical mechanism, large production dataset, data analysis methods, and different ML algorithms were established to predict the temperature of molten steel in LF. The main conclusions are as follows:

- (1) Through the analysis of the LF-refining process and energy conservation, the main factors affecting the end point temperature of molten steel were the initial temperature of molten steel, turnover time of ladle, refining time, weight of molten steel, heating time, argon consumption, and addition amounts of various additions.
- (2) The optimal structures of the temperature prediction model (DNN model) possessed 4 hidden layers, 45 hidden layer neurons, a learning rate of 0.03, L2 regularization coefficient of 2×10^{-4} , batch size of 128, an LReLU activation function, and the optimization algorithms of mini-batch SGD with momentum. For most of the ML models (e.g., DNN, KNN, RELM, BR–BP), the effectiveness of the ZCA whitening was reflected on the temperature prediction in LF-refining process.
- (3) By comparing the performances of the temperature prediction models, it was found that IF–ZCA–DNN had a higher prediction accuracy than both the single ML algorithms and ZCA–ML algorithms. In conclusion, both the selection of ML algorithms and the data processing significantly influence the performance of the prediction model. The modeling method and workflow presented in this study are also applicable to other fields.

ACKNOWLEDGMENTS

This project is funded by the National Natural Science Foundation of China, under Grant Number 51974023, and the funding of State Key Laboratory of Advanced Metallurgy, University of Science and Technology Beijing, under Grant Number 41621005.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

- R.Y. Yin: *Iron Steel*, 2021, vol. 56, pp. 4–9.
- J. Li: *LF Refining Technology*, Metallurgical Industry Press, Beijing, 2012, pp. 135–36.
- U. Camdali and M. Tunc: *J. Iron Steel Res. Int.*, 2006, vol. 13, pp. 18–20.
- O. Volkova and D. Janke: *ISIJ Int.*, 2003, vol. 43, pp. 1185–90.
- Y.J. Wu, Z.H. Jiang, M.F. Jiang, W. Gong, and D.P. Zhan: *J. Iron Steel Res.*, 2002, vol. 14, pp. 9–12.
- D.G. Hong, W.H. Han, and C.H. Yim: *Metall. Mater. Trans. B*, 2021, vol. 52B, pp. 3833–45.
- W.J. Wang, L.F. Zhang, Y. Ren, Y. Luo, X.H. Sun, and W. Yang: *Metall. Mater. Trans. B*, 2022, vol. 53B, pp. 1–7.
- W.J. Yang, L.J. Wang, W. Zhang, and J.M. Li: *Metall. Mater. Trans. B*, 2022, vol. 53B, pp. 3124–35.
- C.A. Myers and T. Nakagaki: *ISIJ Int.*, 2019, vol. 59, pp. 687–96.
- S.K. Thakur, A.K. Das, and B.K. Jha: *Steel Res. Int.*, 2022, vol. 93, p. 2100479.
- S.H. Kwon, D.G. Hong, and C.H. Yim: *Ironmak. Steelmak.*, 2020, vol. 47, pp. 1176–87.
- L.J. Yang, W.Q. Chen, P. Yu, L.C. Li, and L.G. Zhu: *Iron Steel*, 2000, vol. 35, pp. 13–16.
- J. Li, D.F. He, A.J. Xu, and N.Y. Tian: *Steelmaking*, 2012, vol. 28, pp. 50–52.
- X.L. Wang, H. Zhao, and Y.G. Sun: *Metal. Ind. Autom.*, 2007, vol. 4, pp. 5–7.
- Z.C. Xin, J.S. Zhang, J. Zheng, Y. Jin, and Q. Liu: *ISIJ Int.*, 2022, vol. 62, pp. 532–41.
- K. Feng, D.F. He, A.J. Xu, and H.B. Wang: *Steel Res. Int.*, 2016, vol. 87, pp. 79–86.
- H.Y. Tang, X.C. Guo, J.L. Wang, Y. Wang, and P.F. Cheng: *Chin. J. Eng.*, 2016, vol. 38, pp. 139–45.
- X.J. Wang: *IEEE CAA J. Autom. Sin.*, 2017, vol. 4, pp. 770–74.
- G.Q. Fu, Q. Liu, Z. Wang, J. Chang, B. Wang, F.M. Xie, X.C. Lu, and Q.P. Ju: *J. Univ. Sci. Technol. B*, 2013, vol. 35, pp. 948–54.
- W. Lv, Z.Z. Mao, and P. Yuan: *J. Iron Steel Res. Int.*, 2012, vol. 19, pp. 21–28.
- W. Lv, Z.Z. Mao, and P. Yuan: *Steel Res. Int.*, 2012, vol. 83, pp. 288–96.
- W. Lv, Z.Z. Mao, P. Yuan, and M.X. Jia: *Steel Res. Int.*, 2014, vol. 85, pp. 405–14.
- F. He, A.J. Xu, H.B. Wang, D.F. He, and N.Y. Tian: *Steel Res. Int.*, 2012, vol. 83, pp. 1079–86.
- N.K. Nath, N.K. Mandal, A.K. Singh, B. Basu, C. Bhanu, S. Kumar, and A. Ghosh: *Ironmak. Steelmak.*, 2006, vol. 33, pp. 140–50.
- H.X. Tian, Z.Z. Mao, and Y. Wang: *ISIJ Int.*, 2008, vol. 48, pp. 58–62.
- H.X. Tian, Z.Z. Mao, and A.N. Wang: *ISIJ Int.*, 2009, vol. 49, pp. 58–63.
- H.X. Tian, Y.D. Liu, K. Li, R.R. Yang, and B. Meng: *ISIJ Int.*, 2017, vol. 57, pp. 841–50.
- H.X. Tian, Z.Z. Mao, and Z. Zhao: *Chin. J. Sci. Instrum.*, 2008, vol. 29, pp. 2658–62.
- C. Chen, N. Wang, and M. Chen: *ISIJ Int.*, 2021, vol. 61, pp. 1908–14.
- L.L. Zou, J.S. Zhang, Y.S. Han, F.Z. Zeng, Q.H. Li, and Q. Liu: *Metals*, 2021, vol. 11, pp. 1976–92.
- Z.C. Xin, J.S. Zhang, Y. Jin, J. Zheng, and Q. Liu: *Int. J. Miner. Metall. Mater.*, 2023, vol. 30, pp. 335–44.
- K. Sano, S. Matsuda, S. Tohyama, D. Komura, and C. Sutoh: *Sci. Rep.*, 2020, vol. 10, pp. 11714–22.
- G.Q. Huang, X.X. Zhao, and Q.Q. Lu: *J. Saf. Environ.*, 2022, vol. 22, pp. 3412–23.
- Q.S. Deng and G.P. Mei: in *2009 IEEE International Conference on Granular Computing (GRC 2009)*, China, 2009, pp. 126–31.
- F.T. Liu, K.M. Ting, and Z.H. Zhou: in *8th IEEE International Conference on Data Mining (IEEE, ICDM 2008)*, Pisa, Italy, 2008, pp. 413–22.
- Z.C. Xin, J.S. Zhang, J.G. Zhang, Y. Jin, J. Zheng, and Q. Liu: *Ironmak. Steelmak.*, 2021, vol. 48, pp. 1123–32.
- J.D. Rodriguez, A. Perez, and J.A. Lozano: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, vol. 32, pp. 569–75.
- L.V.D. Maaten and G. Hinton: *J. Mach. Learn. Res.*, 2008, vol. 9, pp. 2579–2605.
- A.C. Belkina, C.O. Ciccolella, R. Anno, R. Halpert, J. Spidlen, and J.E.S. Cappione: *Nat. Commun.*, 2019, vol. 10, pp. 5415–27.
- R.Z. Bian, J. Zhang, L. Zhou, P. Jiang, B.Q. Chen, and Y.H. Wang: *J. Comput. Aided Des. Comput. Graph.*, 2021, vol. 33, pp. 1746–54.
- K. Pearson: *Philos. Trans. R. Soc. A*, 1895, vol. 186, pp. 343–414.
- F.T. Liu, K.M. Ting, and Z.H. Zhou: *ACM Trans. Knowl. Discov. Data*, 2012, vol. 6, pp. 1–39.
- K. Hornik, M. Stinchcombe, and H. White: *Neural Netw.*, 1989, vol. 2, pp. 359–66.
- Y. Liu, Q. Zhao, W. Yao, X. Ma, and L. Liu: *Sci. Rep.*, 2019, vol. 9, pp. 19751–63.
- S. Gouravaraju, J. Narayan, R.A. Sauer, and S.S. Gautam: *J. Adhes.*, 2023, vol. 99, pp. 92–115.
- G.B. Huang, Q.Y. Zhu, and C.K. Siew: *Neurocomputing*, 2006, vol. 70, pp. 489–501.
- H.Z. Chen, J.P. Yang, X.C. Lu, X.Z. Yu, and Q. Liu: *Chin. J. Eng.*, 2018, vol. 40, pp. 815–21.
- J. Wang, S.Y. Lu, S.H. Wang, and Y.D. Zhang: *Multimed. Tools Appl.*, 2022, vol. 81, pp. 41611–60.
- Z. Zhang, L.L. Cao, W.H. Lin, J.K. Sun, X.M. Feng, and Q. Liu: *Chin. J. Eng.*, 2019, vol. 41, pp. 1052–60.
- W.Y. Deng, Q.H. Zheng, and L. Chen: in *2009 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2009)*, Nashville, TN, USA, 2009, pp. 389–95.
- Z.C. Xin, J.S. Zhang, W.H. Lin, J.G. Zhang, Y. Jin, J. Zheng, J.F. Cui, and Q. Liu: *Ironmak. Steelmak.*, 2021, vol. 48, pp. 275–83.
- T.M. Cover and P.E. Hart: *IEEE Trans. Inf. Theory*, 1967, vol. 13, pp. 21–27.
- H. Peter: *Machine Learning in Action*, Manning Publications, Greenwich, 2012, pp. 15–25.
- Y. LeCun, Y. Bengio, and G.E. Hinton: *Nature*, 2015, vol. 521, pp. 436–44.
- J.P. Yang, J.S. Zhang, W.D. Guo, S. Gao, and Q. Liu: *ISIJ Int.*, 2021, vol. 61, pp. 2100–110.
- H.X. Yang, J.H. Liu, H.W. Sun, and H.G. Zhang: *IEEE Access*, 2020, vol. 8, pp. 112805–13.
- L.S. Shapley: *A value for n-person games*, Princeton University Press, Princeton, NJ, 1953, pp. 307–17.
- P. Giudici and E. Raffinetti: *Qual. Reliab. Eng. Int.*, 2022, vol. 38, pp. 1318–26.
- J. Bae, Y.R. Li, N. Stahl, G. Mathiason, and N. Kojola: *Metall. Mater. Trans. B*, 2020, vol. 51B, pp. 1632–45.
- J. Nickolls and W.J. Dally: *IEEE Micro*, 2010, vol. 30, pp. 56–69.
- J. Prakash, U. Agarwal, and P.K. Yalavarthy: *Sci. Rep.*, 2021, vol. 11, pp. 18536–45.
- J. Schmidhuber: *Neural Netw.*, 2015, vol. 61, pp. 85–117.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.