A Dynamic Analytics Method Based on Multistage Modeling for a BOF Steelmaking Process

Chang Liu, Student Member, IEEE, Lixin Tang, Senior Member, IEEE, Jiyin Liu, Zhenhao Tang

Abstract—This paper proposes a dynamic analytics method based on a least squares support vector machine with a hybrid kernel to address real-time prediction problems in the converter steelmaking process. The hybrid kernel function is used to enhance the performance of the existing kernels. To improve the model’s accuracy, the internal parameters are optimized by a differential evolution algorithm. In light of the complex mechanisms of the converter steelmaking process, a multistage modeling strategy is designed instead of the traditional single-stage modeling method. Owing to the dynamic nature of the practical production process, great effort has been made to construct a dynamic model that uses the prediction error information based on the static model. The validity of the proposed method is verified through experiments on real-world data collected from a BOF steelmaking process. The results indicate that the proposed method can successfully solve dynamic prediction problems and outperforms other state-of-the-art methods in terms of prediction accuracy.

Index Terms—Dynamic analytics, multistage modeling, least squares support vector machine, hybrid kernel function, basic oxygen furnace.

I. INTRODUCTION

THE iron and steel industry is a major backbone of many fundamental production processes worldwide. It provides the basic raw materials for civil infrastructure, such as that for construction, machinery, automobiles, shipbuilding, railroads, and for other modern manufacturing industries [1-4]. Usually, iron and steel production processes consist of ironmaking, steelmaking-continuous casting, hot rolling, and cold rolling. Fig. 1 shows the overall process of iron and steel production, where the converter steel production accounts for over 80% of the total steel production. Thus, basic oxygen furnace (BOF) steelmaking is a critical bottleneck process in the entire production chain [5]. To guarantee safe production and improve product performance, many scientists and engineers pay a great deal of attention to quality predictions in the BOF steelmaking process [6-10].

For the BOF production process in a hazardous environment, it is a challenge to establish an accurate prediction model. The widely used modeling methods for the BOF process can be roughly divided into two categories: mechanism models, and data-driven models. The former use analytical models derived from the energy balance, mass balance, momentum balance, and physicochemical reactions occurring in the converter furnace [11-13]. However, the complexity of mechanism models makes it difficult to reduce them to a set of analytical equations. Considering the lack of exact mathematical models, some data-driven models can be constructed by machine learning methods [14], because sufficient sampling data can be obtained from the industrial production process using a vice-gun probe, the spectral analysis method, and the flue gas analysis method [15, 16]. At present, many prevalent data analytics methods have been used in the prediction problem, including support vector machines (SVMs) [17], least squares support vector machines (LSSVMs) [18], extreme learning machines (ELMs) [19], and neural networks (NNs) [20]. Additionally, some deep learning (DL) methods have emerged in recent years [21]. Mahmud et al. [22] surveyed DL, reinforcement learning, and deep reinforcement learning techniques in biological applications. Wang et al. [23] provided a comprehensive survey of advanced DL methods and applications for smart manufacturing. Liu et al. [24] proposed an ensemble deep kernel learning (EDKL) model to predict the melt index in industrial polymerization processes.
These DL methods are also promising to be applied to the BOF steelmaking process. Most of the literature on quality predictions in the BOF steelmaking process focuses on static models [6]. Wang et al. [7] presented an input-weighted SVM for endpoint prediction in the BOF steelmaking process. Xu et al. [8] designed a static prediction model based on an SVM and spectrum information. Han and Liu [9] proposed an ELM method to address end-point prediction problems using model parameters that were adaptively adjusted by the evolutionary membrane algorithm. In [10], an improved case-based reasoning (CBR) method with fuzzy c-means clustering (FCM) was applied to end-point prediction for the BOF. Some hybrid methods have also been applied to the steelmaking process [25]. Tian and Mao [26] proposed an ELM with a modified AdaBoost.RT algorithm. This hybrid method can accurately predict the temperature of molten steel in a ladle furnace. Das et al. [27] developed an artificial neural network (ANN) with Bayesian regularization and an adaptive neuro-fuzzy inference system (ANFIS) for prediction problems in an electric arc furnace. Liu et al. [28] utilized computer vision and a general regression neural network (GRNN) to solve the end-point BOF prediction problem.

Compared to extensive research on end-point BOF prediction problems, static models utilize only the difference between the initial state and endpoint state, relatively little attention has been paid to internal changes during the reaction process. Nevertheless, real-world problems are generally dynamic, and involve various unexpected real-time events. There are few studies concerning dynamic monitoring, prediction, or control problems [29]. Han and Zhao [30] presented an adaptive-network-based fuzzy inference system (ANFIS) with a robust relevance vector machine (RRVM) method to solve dynamic control problems in the BOF steelmaking process. For the ladle furnace steelmaking process, Wang et al. [31] proposed a multiple dynamic principal component analysis method to monitor complex batch processes.

However, due to the high temperature and the corrosive environment in the BOF steelmaking process, some challenging problems are difficult to overcome in practice:

1) The temperature and quality of molten steel must be continuously estimated and predicted over a short time period to avoid production delays.
2) To replace experienced judgments, an effective dynamic prediction model, and internal operation parameters need to be adaptively optimized.
3) Due to the fierce physicochemical reaction in the BOF steelmaking process, the dynamic prediction model must be adjusted in real time.
4) The performance of a single-stage model is unstable under different smelting states and different blowing stages.

To solve these problems, a dynamic analytics method based on multistage modeling is developed. The proposed method can predict the temperature and multi-element quality content of molten steel in real time. The contributions of our proposed method are as follows:

1) A Gaussian kernel and a positive semi-definite kernel are adopted as a hybrid kernel function, which improves the performance of the LSSVM with a single kernel function.
2) The model parameters are adaptively optimized by a differential evolution (DE) algorithm.
3) An idea for multistage modeling is proposed instead of the traditional single modeling method.
4) A dynamic model is developed to realize the real-time prediction in the converter steelmaking.

The rest of this paper is organized as follows: Section II provides a brief description of the BOF steelmaking process. Section III presents our proposed method in detail. In Section IV, the computational results on a real-world industrial application are reported and analyzed. Finally, Section V draws the conclusion and discusses the future work.

II. PROBLEM DESCRIPTION

The first top-blown oxygen converter steelmaking process, developed in 1948, was also known as the Linz–Donawitz (LD) process, or the BOF steelmaking process. Converter steelmaking is a batch process that converts carbon-rich molten pig iron into molten steel by using a lance to blow oxygen (O\textsubscript{2}) into a converter furnace. Generally, in an oxygen jet impact area, the carbon (C) in molten iron has a direct oxidation reaction with the blown O\textsubscript{2}, generating carbon monoxide (CO). When C and carbon dioxide (CO\textsubscript{2}) are mixed, CO is generated as the chemical reactant. All decarburization reaction kinetics
The BOF steelmaking process is illustrated in Fig. 2. First, molten iron in a ladle and some steel scraps are added directly to the converter furnace. High-purity O\textsubscript{2} is blown toward the iron bath through a water-cooled lance, and maintained at certain distance above the surface of the molten steel. The pure O\textsubscript{2} blowing onto the molten steel causes the dissolved carbon in the liquid steel to undergo a chemical reaction, forming CO and CO\textsubscript{2}, accompanied by a high temperature between approximately 1200°C and 1700°C. Meanwhile, auxiliary materials (burnt lime, dolomite, composite silicon carbide, etc.) are thrown into the furnace to become slag, and absorb unwanted chemical elements during the reaction process. An inert gas (such as argon (Ar) or nitrogen (N\textsubscript{2})) are blown into the bottom of the furnace to stir the molten steel. After approximately 20 min, the temperature is measured, and tiny part of molten steel are taken with a sublance. Finally, the steel tapping process is executed in which the molten steel is poured out through a tap hole, and the slag is poured into a slag pot.

Fig. 3 shows a simulation diagram of decarburization speed and blowing time. According to the physicochemical reactions at different time intervals in the BOF, the steelmaking production process is divided into three stages, as shown in Fig. 4. The detailed reaction mechanism can be described as follows:

1) In the early blowing stage, due to the low temperature of the molten iron, Si and manganese (Mn) undergo a more rapid oxidation than C and are thoroughly oxidized after approximately 2 to 4 min. Meanwhile, the iron oxide dissolves into slag, and the dissolved limestone also turns phosphorus (P) oxide into slag. Heat is released from Si, Mn, P, and molten iron, leading to the high temperature of the molten pool. The purpose of this stage is to generate good slag quickly, and introduce a positive effect on dephosphorization and desulfurization.

2) In the intermediate blowing stage, Si and Mn complete the oxidation process in the molten iron, and the rate of carbon oxidation increases due to the high temperature of the molten iron. Because of the fierce oxidation reaction occurring between C and O\textsubscript{2}, the rate of decarburization reaches a maximum, and the phenomena of Mn reversion and P reversion will occur in the converter furnace, accompanied by a high desulfurization reaction level. Meanwhile, a generous amount of scraps melt under the high temperature. The purpose of this stage is to decarburize, remove sulfur (S), and control the stirring rate between the O\textsubscript{2} and the inert gas blowing at the bottom to prevent slopping.

3) In the final blowing stage, the rate of decarburization decreases due to the low remaining C content in the molten steel. The iron is then partially oxidized by blowing O\textsubscript{2} into the molten pool. The operations of adding auxiliary materials and blowing O\textsubscript{2} continue until the end-point control demand is satisfied. Finally, the temperature and the quality of the molten steel satisfy the requirement for steel tapping.

III. The Proposed Method

In this section, a machine learning method based on LSSVM with a hybrid kernel function is proposed to construct a static
model, in which a Gaussian kernel and a positive semi-definite kernel are added into the hybrid kernel function. The internal model parameters are adaptively optimized by a DE algorithm. Next, a multistage modeling strategy is designed for the complex steelmaking reaction process. To predict the temperature and quality of molten steel in real time, a dynamic model based on prediction error is established. Finally, the overall structure of the proposed method is presented.

A. Static modeling method

The static model is mainly applied to the end-point prediction problem in the BOF steelmaking process. In this paper, the static model is the basis of the dynamic model, which is based on LSSVM with a hybrid kernel and DE.

1) LSSVM with a hybrid kernel: SVM is an effective machine learning method for regression problems. LSSVM was proposed by Suykens and Vandewalle [35] as an alternative form of SVM. Because the quadratic optimization problem of SVM is converted into a system of linear equations to be solved, LSSVM has an advantage in computation speed. However, the choice of kernel functions is crucial to LSSVM. Thus, this paper proposes an LSSVM with a hybrid kernel (HKLSSVM) method. The principle of HKLSSVM is described below:

Given a training set of \( N \) data points \( \{x_i, y_i\}, i = 1, 2, \ldots, N \), \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \) are the input data and the corresponding output data, respectively, and \( d \) denotes the feature dimension of the input data. The linear equation with high-dimensional feature space is expressed as follows:

\[
f(x_i) = w^T \cdot \phi(x_i) + b, \quad i = 1, 2, \ldots, N
\]

where the nonlinear mapping function \( \phi(x) \) transforms the input data \( x \) into a high-dimensional feature space. Here, \( w^T \) is the regression coefficient, and \( b \) is the bias.

Based on the principle of structural risk minimization, the regression problem is converted to an optimization problem with constraints:

\[
\min_{w, b, e} J(w, e) = \frac{1}{2} \|w\|^2 + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2
\]

\[
s.t. \quad y_i = w^T \cdot \phi(x_i) + b + e_i,
\]

\[
\gamma \geq 0.
\]

where \( \gamma \) is a penalty coefficient, and \( e_i \) is a slack variable. According to Eq. (2), the Lagrangian function can be written as

\[
L(w, b, e, \alpha) = J(w, e) - \sum_{i=1}^{N} \alpha_i [w^T \cdot \phi(x_i) + b + e_i - y_i]
\]

where \( \alpha_i \) is the Lagrange multiplier. By applying the Karush-Kuhn-Tucker (KKT) condition, the following linear equations can be obtained using transformation:

\[
\begin{bmatrix}
0 \\
1
\end{bmatrix} \Omega + \gamma^{-1} \begin{bmatrix}
b \\
y
\end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

where \( I = (1, \ldots, 1)^T \), \( \alpha = (\alpha_1, \ldots, \alpha_N)^T \), \( y = (y_1, \ldots, y_N)^T \), and \( \Omega \) is a symmetric matrix of the kernel function such that \( \Omega = \{\Omega_{ij} | i, j = 1, \ldots, N\} \), and \( \Omega = \phi(x_i)^T \phi(x_j) \). According to Mercer's condition, the kernel function is expressed as follows:

\[
K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j), \quad i, j = 1, 2, \ldots, N.
\]

The LSSVM model can then be defined as

\[
f(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x) + b.
\]

The Gaussian radial basis function (RBF) has good mapping ability [36], and is a widely used kernel function in LSSVM. This Gaussian kernel function is defined as

\[
K_i(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}), \quad i, j = 1, 2, \ldots, N
\]

where \( \sigma \) is the width parameter of the kernel function.

To compensate for the weakness of a single kernel function, some composite kernel functions [37] have been effectively applied to regression problems. Thus, this paper takes full advantage of both the Gaussian kernel and a positive semi-definite kernel as the hybrid kernel function.

As stated in [38], the inner product of the kernel function is a mapping with a unit matrix, which is described as follows:

\[
K(x_i, x_j) = (Ax_i)^T \cdot Ax_j = x_i^T A^T Ax_j.
\]

Based on the above description, we directly construct \( B \) for the kernel function. Let \( B = A^T A \), where \( B \) is regarded as a positive semi-definite matrix. Thus, the positive semi-definite kernel function is defined in Eq. (10):

\[
K_{\gamma}(x_i, x_j) = x_i^T B x_j, \quad i, j = 1, 2, \ldots, N.
\]

It is clear that the key idea of the kernel function is to construct \( B \). To construct the positive semi-definite matrix, we optimize \( B \) using DE. Because the dimension of \( B \) has a great impact on the computation for the kernel function, to reduce its complexity, we simplify \( B \) to a diagonal matrix:

\[
B = \begin{pmatrix}
b_1 & 0 & \cdots & 0 \\
0 & b_2 & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & b_d
\end{pmatrix}
\]

The optimized parameter in the kernel function is \( b_k (k = 1, 2, \ldots, d) \). When \( b_k \geq 0 \), the diagonal matrix \( B \) is considered a positive semi-definite matrix. Given that condition, the computation time decreases substantially.

In addition, the validity of the matrix has been proven in Proposition 1 [39]. Based on that proposition, we can infer that \( K_{\gamma}(x_i, x_i) \) is a valid kernel function.

**Proposition 1:** If \( x_i \in \mathbb{R}^d \) and \( x_j \in \mathbb{R}^d \) are input vectors, and \( B^{bid} \) is a symmetric positive semi-definite matrix, then \( K_{\gamma}(x_i, x_j) = x_i^T B x_j \) is a kernel function.

Thus, the hybrid kernel function can be expressed as follows:

\[
K(x_i, x_j) = (1 - \lambda)K_i(x_i, x_j) + \lambda K_{\gamma}(x_i, x_j), \quad i = 1, 2, \ldots, N
\]

where \( \lambda \) is a scale coefficient of the hybrid kernel function, and \( \lambda \geq 0 \).
The validity of the hybrid kernel function has been proven in [39], and thus the hybrid kernel function $K(x, x_i)$ is a valid kernel function.

2) Parameter optimization algorithm: In this paper, to obtain an accurate static model, due to the fast and effective search capability of the DE algorithm [40], the main model parameters ($\gamma$, $\sigma$, $B$, $\lambda$) are adaptively optimized by DE. The main steps of the algorithm are described as follows:

Step 1. Initialization: Set the maximum number of generations $g_{\text{max}}$. In this implementation, $z^g_p = (z^{g,p}_1, z^{g,p}_2, \ldots, z^{g,p}_{NP})$ represents the $p$th individual with $D$ dimensions in each generation $g$, where $p = 1, 2, \ldots, NP$. The components of each individual are subject to the boundary constraints $L_q \leq z^{g,p}_q \leq U_q$, $q = 1, 2, \ldots, D$, and $L_q$ and $U_q$ are the lower and upper boundaries of the $q$th dimension, respectively. $Z^0 = (z^0_1, z^0_2, \ldots, z^0_{NP})$ is regarded as the initial population, and the initial individuals are randomly generated within the boundary scope. $\hat{f}(z^g_p)$ denotes the fitness value of the $p$th individual in the $g$th generation, and $\hat{f}(\cdot)$ is defined by minimizing the root-mean-square error between the predicted values and the actual values.

Step 2. Mutation: In generation $g$, each individual is updated by the proposed mutation strategy as follows:

$$v^g_{p,q} = z^g_{p,q} + F_1(z^g_{i,q} - z^g_{s,q}) + F_2(g/g_{\text{max}})(z^g_{r,q} - z^g_{i,q} + z^g_{s,q})/2 \quad (13)$$

where $v^g_{p,q}$ represents the individual to be mutated, $p = 1, 2, \ldots, NP$, $q = 1, 2, \ldots, D$, $F_1$ and $F_2$ are the mutation factors, and $r_1$, $r_2$ and $r_3$ are different integers randomly selected from $[1, NP]$.

Step 3. Crossover: A trial individual $o^g_p = (o^{g,1}_p, \ldots, o^{g,D}_p)$ is obtained through the following crossover operation:

$$o^{g}_{p,q} = \begin{cases} v^g_{p,q} & \text{if } (\text{rand}_{p,q} \leq CR) \\ z^g_{p,q} & \text{otherwise} \end{cases} \quad (14)$$

where $CR$ is a crossover rate, and $\text{rand}_{p,q}$ is a random number from $[0, 1]$.

Step 4. Selection: $z^{g+1}_p$ is determined by comparing the fitness values between $\hat{f}(o^g_p)$ and $\hat{f}(z^g_p)$; thus, the selection strategy is described as follows:

$$z^{g+1}_p = \begin{cases} o^g_p & \text{if } (\hat{f}(o^g_p) \leq \hat{f}(z^g_p)) \\ z^g_p & \text{otherwise} \end{cases} \quad (15)$$

Step 5. Stopping condition: If the algorithm satisfies the maximum generation $g_{\text{max}}$, the parameter optimization operation will terminate, and the optimal parameters will be acquired based on the minimum fitness value. Otherwise, next generation is performed, and the algorithm returns to Step 2.

B. Multistage modeling strategy

A previous study was conducted to introduce the static state modeling method, which is used to establish the single-stage prediction model. However, some model parameters in the reaction process are variable due to uncertainty factors. Thus, the static model will not satisfy the actual production requirements, and cannot precisely reflect the actual physicochemical reaction mechanism. In recent years, the use of multistage modeling strategies has enabled a promising future in the field of multistage manufacturing processes [41, 42]. In this paper, the idea refers to a multistage model consisting of the different reaction stages required to complete the real-time product quality prediction.

The principle of multistage modeling in the BOF steelmaking process is illustrated in Fig. 5. In terms of multistage temperature prediction models, the initial temperature enters the model at stage 1, denoted as $Y_0$, and the current operation parameters and state parameters are denoted as $X_1$. Thus, the current input data consists of $Y_0$ and $X_1$, which are transmitted to the model at stage 1, and the predicted value ($Y_1$) is obtained by the current model. Because the reaction process is continuous in real time, the current output value is regarded as the partial input data of the next stage. In other words, $Y_1$ and $X_2$ are transferred to the next stage. However, determining which stage should be performed is a problem. We consider the predicted result at the current stage as the evaluation criterion, which sets scopes for different stages. If $Y_1$ exceeds the boundary of the current stage, stage 2 will be selected as the next stage; otherwise, the model will continue to use stage 1. Subsequent operations proceed according to the above rules: $R$ and $M$ denote the total number of multistage models and the total amount of real-time data, respectively. Thus, the divided stages are key to the multistage modeling strategy.

According to the reaction mechanism in the BOF steelmaking process, the first reaction stage will terminate when the temperature reaches approximately 1420°C. However, the terminal condition of the intermediate stage is mainly judged by whether the carbon content reaches 0.45%.
Therefore, we divide the entire process into many single-stage processes due to the mixed data types arising from multiple processes at each stage. Moreover, considering the smelting temperature (for convenience, temperature is denoted as T) and different molten steel quality (such as C, Mn, Si, S, and P) in the converter furnace, each multistage model is independent. The modeling strategy is driven only by events; it has no relationship with reaction time. To represent different states more clearly, the divided results of multistage models are shown in Table I.

C. Dynamic real-time prediction model

The static model lacks feedback concerning real-time variations. One goal of this paper is to establish a dynamic model to solve the problem. In the sense that the dynamic model as defined in this paper is equivalent to dynamic compensation, the feedback is based on model prediction error. The dynamic prediction error is used to modify the static model to attain the real trajectory.

The static model predicts the single-stage process, whereas the dynamic model provides a prediction error that compensates for the static model. The static models at different stages are fixed and have been established by historical data. However, among the propagation processes of prediction errors, the dynamic model changes with the new input data at different stages. The most challenging aspect of dynamic multistage prediction is accumulating and propagating the prediction error. To interpret the relationship between the static state and dynamic state, this paper provides an example showing dynamic temperature prediction of the BOF steelmaking process in Fig. 6, where the dynamic profile is acquired by the generated points using the static model. We can see that it is difficult to deal with in-process prediction in BOF steelmaking, and the variation in the static model is propagated, and accumulates throughout a series of stages. However, the feedback of prediction errors can help improve the prediction model because the current outputs of one stage are the inputs of the next stage. Therefore, a successful dynamic multistage model should compensate for the prediction errors at each stage while aiming to obtain an accurate prediction across the entire process, rather than an end-point prediction at the final stage.

Based on the variation propagation derived from the feedback of the prediction error, we establish a dynamic prediction model according to Eq. (6) as follows:

$$ f(x_i) = \sum_{j=1}^{N} \alpha_j K(x_i, x_j) + b + S(x_i), \ i = 1, 2, ..., M $$

(16)

where $S(x_i)$ is a dynamic error term used to update the dynamic model. In Eq. (16), $x_i, i = 1, 2, ..., M$ represents the real-time data, and $x_j, j = 1, 2, ..., N$ represents the training data. The detailed formulations are expressed as follows:

$$ S(x_i) = \text{sgn}^\theta(\hat{e}_{i+1}) \cdot E_i \cdot \theta \cdot \sum_{j=1}^{N} p(x_i, x_j) $$

(17)

where $\theta$ is a learning rate parameter, and $\hat{e}_{i+1}$ is the prediction error at the previous moment with an initial value $\hat{e}_0 = 0$. The kernel scale coefficient $\eta(x_i, x_j)$ is defined as

$$ \eta(x_i, x_j) = \frac{K_i(x_i, x_j) K_j(x_i, x_j)}{\sqrt{K_i(x_i, x_i) K_i(x_i, x_j) K_j(x_i, x_i) K_j(x_i, x_j)}} $$

(18)

where $K_i(x_i, x_j)$ and $K_j(x_i, x_j)$ are the kernel functions of the static model, respectively, $i = 1, 2, ..., M$, and $j = 1, 2, ..., N$.

The modified dynamic prediction error $E_i$ is calculated by

---

### Table I

**DIVIDED RESULTS OF MULTISTAGE MODELS.**

<table>
<thead>
<tr>
<th>$R$</th>
<th>T (°C)</th>
<th>C(%)</th>
<th>Mn(%)</th>
<th>Si(%)</th>
<th>S(%)</th>
<th>P(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt; 1230</td>
<td>&gt; 4.050</td>
<td>&gt; 0.140</td>
<td>&gt; 1.000</td>
<td>&gt; 0.020</td>
<td>&gt; 0.080</td>
</tr>
<tr>
<td>2</td>
<td>1230-1250</td>
<td>4.050-3.750</td>
<td>0.140-0.100</td>
<td>1.000-0.700</td>
<td>0.020-0.010</td>
<td>0.080-0.050</td>
</tr>
<tr>
<td>3</td>
<td>1250-1280</td>
<td>3.750-3.450</td>
<td>&lt; 0.100</td>
<td>0.700-0.350</td>
<td>0.010-0.008</td>
<td>&lt; 0.050</td>
</tr>
<tr>
<td>4</td>
<td>1280-1300</td>
<td>3.450-3.150</td>
<td>\</td>
<td>0.350-0.200</td>
<td>0.008-0.005</td>
<td>\</td>
</tr>
<tr>
<td>5</td>
<td>1300-1320</td>
<td>3.150-2.950</td>
<td>\</td>
<td>0.200-0.020</td>
<td>0.005-0.003</td>
<td>\</td>
</tr>
<tr>
<td>6</td>
<td>1320-1350</td>
<td>2.950-2.650</td>
<td>\</td>
<td>&lt; 0.020</td>
<td>0.003-0.001</td>
<td>\</td>
</tr>
<tr>
<td>7</td>
<td>1350-1380</td>
<td>2.650-2.450</td>
<td>\</td>
<td>&lt; 0.001</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>8</td>
<td>1380-1400</td>
<td>2.450-2.150</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>9</td>
<td>1400-1420</td>
<td>2.150-1.950</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>10</td>
<td>1420-1450</td>
<td>1.950-1.650</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>11</td>
<td>1450-1480</td>
<td>1.650-1.450</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>12</td>
<td>1480-1500</td>
<td>1.450-1.150</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>13</td>
<td>1500-1520</td>
<td>1.150-0.950</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>14</td>
<td>1520-1550</td>
<td>0.950-0.650</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>15</td>
<td>1550-1580</td>
<td>0.650-0.450</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>16</td>
<td>1580-1600</td>
<td>0.450-0.250</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>17</td>
<td>1600-1630</td>
<td>0.250-0.150</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>18</td>
<td>1630-1650</td>
<td>0.150-0.045</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>19</td>
<td>&gt; 1650</td>
<td>&lt; 0.045</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
</tbody>
</table>

Note: “>” , “<” and “=” stand for “More than”, “Less than” and “Nothing”, respectively.
Algorithm 1 Pseudo-code for DM-HKLSSVM

1 Construct the static model
   1) Set the basic model parameters.
   2) Construct the hybrid model function.
   3) Establish the HKLSSVM model.
   4) Input the training data.
   5) Optimize the main model parameters by DE.
      - Step 1: Initialize the parameters of DE, and create an initial population, and evaluate the fitness values.
      - While stopping criterion is not satisfied do
        - Step 2: Conduct mutation;
        - Step 3: Conduct crossover;
        - Step 4: Perform selection.
      - End While
   6) Output the static training model, and obtain the optimal parameters;

2 Multistage modeling strategy
   7) Divide the different stages for T, C, Mn, Si, S, and P.
   8) Establish the corresponding models based on the divided stages.

3 Dynamic real-time prediction model
   9) Input the real-time data $Y_i$ ($i = 1, 2, \ldots, M$) for the instance of $T$ (the instances of C, Mn, Si, S, and P are similar to $T$).
10) Set divided values of the stages as $BV_j$, $j = 1, 2, \ldots, R-1$, and select the proper “Model” for each stage.
   - If $Y_i < BV_1$
     - Model = Model1;
   - Else If $Y_i \geq BV_1$ and $Y_i < BV_2$
     - Model = Model2;
   - ....
   - Else If $Y_i \geq BV_{R-2}$ and $Y_i \leq BV_{R-1}$
     - Model = Model_{R-1};
   - Else If $Y_i > BV_{R-1}$
     - Model = Model_{R};
   - End If
11) Test the real-time data by M-HKLSSVM.
12) Establish dynamic multistage models.
13) Output the dynamic real-time prediction results.

$$E_i = \sqrt{\frac{e_{i-1}^2 + \delta^2 \cdot (N - 1)}{N}}$$  \hspace{1cm} (19)

where $\delta$ is the root-mean-square error of the static training model, which is obtained by Eq. (20):

$$\delta = \sqrt{\frac{\sum_{i=1}^{N} u_i^2}{N}}$$  \hspace{1cm} (20)

where $u_i$ is the prediction error of each training data item in the static model.

D. The overall framework of the proposed method

In this paper, the HKLSSVM with a dynamic multistage (DM-HKLSSVM) modeling method can solve the real-time prediction problems in the BOF steelmaking process. The HKLSSVM is used to establish the static model, and its parameters are adaptively optimized by the DE algorithm. The different reaction stages in the BOF steelmaking process are established by the multistage modeling strategy. Furthermore, to dynamically predict the molten steel quality in the production process, a dynamic prediction method with error compensation is applied to the model. Fig. 7 shows the schematic diagram of the dynamic analytics method based on multistage modeling. Additionally, to interpret the proposed method more clearly, this paper provides the pseudo-code for the DM-HKLSSVM in Algorithm 1.

IV. EXPERIMENTS

In this section, to show the practicability of the DM-HKLSSVM, we test it in a practical application. Based on real data from the industry, we solve the dynamic prediction problems in the BOF steelmaking process. Then, through a comparison analysis with other state-of-the-art methods, we discuss the performance of the proposed method.

A. Experimental Setting

1) Experimental platform: The experiment was performed on a computer with an Intel® Core™ i7-6700 3.40 GHz CPU and 16 GB RAM. The operating system was a 64-bit Windows 7 version. The software was developed using Microsoft Visual Studio 2008, and the programming language was C++.

2) Parameter setup: To identify better model parameters, this paper sets different search scopes for the HKLSSVM, in which $\gamma$ and $\sigma$ are set to [1, 1000] and [1, 10], respectively. Meanwhile, to address the different prediction problems of molten steel quality, $\lambda$ is set to [0, 0.1] in terms of T, S, and P, and the corresponding parameter is set to [0.3, 0.4] with respect to C, Mn, and Si. The values of other specific parameters are shown in Table II.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig6.png}
\caption{Fig. 6. Relationship between static state and dynamic state.}
\end{figure}
B. Data processing

1) Experimental data: The data used in this experiment were collected from a real BOF steelpmaking production process. A large amount of data was obtained through multi-source sensors: a flame analyzer is used to measure the real-time temperature, and some off-gas data are acquired by the gas analyzer. The quality components in the molten steel are sampled by sublance and throwing probes. According to the operational experiences of an actual steelpmaking process, the input feature data for the static model include the initial T, the initial contents of C, Mn, Si, S, and P, the weight of steel scrap, the weight of molten iron, the height of oxygen lance, the flows of flue gas, O2, CO, CO2, N2, and Ar, and seven types of auxiliary materials (burnt lime, dolomite, composite silicon carbide, etc.). Hence, a total of 22 input features are considered for the training data sets. The output feature data consist of the process smelting temperature (T) and five process quality component contents (C, Mn, Si, S, and P).

2) Pre-processing: To guarantee the effectiveness of the experimental data, data sampled within one month are used in the experiment; the original data were processed and clustered. Some abnormal data were removed, and large amounts of missing data were complemented through interpolation and some statistical methods. In addition, due to the various features, normalization was implemented before modeling. Finally, 30 instances were selected as testing data sets, as shown in Table III. The sampling period of each data item is 6 sec. These processed data are used as the experimental data to verify the performance of the proposed method.

C. Experimental results

1) Comparisons with different strategies: In this subsection, experiments are performed to compare the proposed strategies to solve the dynamic prediction problems in the BOF steelpmaking process. Fig. 8 shows the dynamic prediction results for part of the instances. The proposed DM-HKLSSVM has a substantial advantage over LSSVM with the multistage (M-LSSVM) strategy, and over HKLSSVM with the multistage (M-HKLSSVM) strategy for T, C, Mn, Si, S, and P, and the trends of the dynamic curves are closer to the real curves.

The statistical comparison results of 30 instances are presented in Table IV. For convenience, RMSE, MAPE, and MAXE denote the root-mean-square error, mean absolute percentage error, and maximum error between the predicted values and the real values, respectively. It follows that DM-HKLSSVM has better performance than M-HKLSSVM and M-LSSVM on the different instances, and that M-HKLSSVM can obtain a smaller prediction error than M-LSSVM. From Table IV, we can see that the prediction error values of T are generally between 0°C and 15°C, and the RMSE of C is below 0.04. Due to the instability of Mn and S caused by a high temperature, the RMSE of Mn and S can be limited to below 0.01 and 0.001, respectively. For Si and P, we also achieve a relatively high precision prediction, and the RMSE can be kept below 0.03 and 0.005, respectively. These results illustrate that DM-HKLSSVM can dynamically predict the temperature and other quality component contents during the BOF steelpmaking process, and the prediction error rate is acceptable relative to actual production demands.
Fig. 8. Dynamic prediction results of 3 instances using M-LSSVM, M-HKLSSVM, and DM-HKLSSVM: (a) Temperature, (b) Carbon, (c) Manganese, (d) Silicon, (e) Sulfur, (f) Phosphorus.
TABLE IV

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Indexes</th>
<th>M-LSVM</th>
<th>M-HKLSSVM</th>
<th>DM-HKLSSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>RMSE</td>
<td>1.16E+01</td>
<td>9.42E+00</td>
<td>4.00E+00</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>6.75E-01</td>
<td>5.28E-01</td>
<td>2.12E-01</td>
</tr>
<tr>
<td></td>
<td>MAXE (C)</td>
<td>2.60E+01</td>
<td>2.51E+01</td>
<td>1.56E+01</td>
</tr>
<tr>
<td>C</td>
<td>RMSE</td>
<td>1.91E-01</td>
<td>6.83E-02</td>
<td>3.94E-02</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>9.26E+00</td>
<td>6.66E+00</td>
<td>3.45E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>7.77E-01</td>
<td>1.79E-01</td>
<td>1.28E-01</td>
</tr>
<tr>
<td>Mn</td>
<td>RMSE</td>
<td>2.47E-02</td>
<td>2.10E-02</td>
<td>9.73E-03</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>1.78E+01</td>
<td>1.50E+01</td>
<td>6.47E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>5.22E-02</td>
<td>3.92E-02</td>
<td>2.22E-02</td>
</tr>
<tr>
<td>Si</td>
<td>RMSE</td>
<td>7.93E-02</td>
<td>5.92E-02</td>
<td>2.72E-02</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>1.51E+02</td>
<td>4.90E+01</td>
<td>1.77E+01</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>2.53E-01</td>
<td>1.87E-01</td>
<td>1.25E-01</td>
</tr>
<tr>
<td>S</td>
<td>RMSE</td>
<td>1.23E-03</td>
<td>1.05E-03</td>
<td>3.83E-04</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>3.78E+01</td>
<td>3.02E+01</td>
<td>9.53E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>2.44E-03</td>
<td>2.29E-03</td>
<td>1.21E-03</td>
</tr>
<tr>
<td>P</td>
<td>RMSE</td>
<td>8.39E-03</td>
<td>7.60E-03</td>
<td>4.67E-03</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>2.07E+01</td>
<td>1.92E+01</td>
<td>8.21E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>2.14E-02</td>
<td>1.92E-02</td>
<td>1.93E-02</td>
</tr>
</tbody>
</table>

TABLE V

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Indexes</th>
<th>SVR</th>
<th>RVM</th>
<th>KELM</th>
<th>DSAE</th>
<th>DM-HKLSSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>RMSE</td>
<td>2.27E+01</td>
<td>1.26E+01</td>
<td>1.55E+01</td>
<td>1.13E+01</td>
<td>4.00E+00</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>1.35E+00</td>
<td>1.47E-01</td>
<td>8.29E-01</td>
<td>5.96E-01</td>
<td>2.12E-01</td>
</tr>
<tr>
<td></td>
<td>MAXE (C)</td>
<td>5.08E+01</td>
<td>1.60E-02</td>
<td>6.34E+01</td>
<td>4.62E+01</td>
<td>1.56E+01</td>
</tr>
<tr>
<td>C</td>
<td>RMSE</td>
<td>9.99E-02</td>
<td>4.03E-01</td>
<td>1.45E-01</td>
<td>8.51E-02</td>
<td>3.94E-02</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>1.53E+01</td>
<td>2.37E+01</td>
<td>2.29E+01</td>
<td>1.26E+01</td>
<td>3.45E-00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>2.00E+01</td>
<td>3.05E+00</td>
<td>7.73E-01</td>
<td>4.20E+01</td>
<td>1.28E+01</td>
</tr>
<tr>
<td>Mn</td>
<td>RMSE</td>
<td>3.00E-02</td>
<td>5.88E-02</td>
<td>1.25E-02</td>
<td>9.09E-03</td>
<td>9.73E-03</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>2.05E+01</td>
<td>3.71E+01</td>
<td>7.82E+00</td>
<td>5.30E+00</td>
<td>6.47E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>5.06E-02</td>
<td>1.27E-01</td>
<td>2.64E-02</td>
<td>2.51E-02</td>
<td>2.22E-02</td>
</tr>
<tr>
<td>Si</td>
<td>RMSE</td>
<td>1.11E-01</td>
<td>1.38E-01</td>
<td>6.71E-02</td>
<td>9.78E-02</td>
<td>2.72E-02</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>5.21E+02</td>
<td>2.87E+01</td>
<td>6.87E-02</td>
<td>3.61E+02</td>
<td>1.77E+01</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>4.47E-01</td>
<td>6.78E-01</td>
<td>2.87E-01</td>
<td>3.23E-01</td>
<td>1.25E-01</td>
</tr>
<tr>
<td>S</td>
<td>RMSE</td>
<td>1.31E-03</td>
<td>3.46E-03</td>
<td>6.22E-04</td>
<td>6.94E-04</td>
<td>3.83E-04</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>3.17E+01</td>
<td>9.14E+01</td>
<td>1.72E+01</td>
<td>2.26E+01</td>
<td>9.53E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>2.62E-03</td>
<td>7.85E-03</td>
<td>1.77E-03</td>
<td>2.01E-03</td>
<td>1.21E-03</td>
</tr>
<tr>
<td>P</td>
<td>RMSE</td>
<td>7.98E-03</td>
<td>1.28E-02</td>
<td>7.77E-03</td>
<td>9.73E-03</td>
<td>4.67E-03</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>1.08E+01</td>
<td>4.00E+01</td>
<td>1.25E+01</td>
<td>1.37E+01</td>
<td>8.21E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>1.66E-02</td>
<td>5.59E-02</td>
<td>2.34E-02</td>
<td>2.38E-02</td>
<td>1.93E-02</td>
</tr>
</tbody>
</table>

2) Comparisons with state-of-the-art methods: To illustrate the competitiveness of the proposed method, we compared DM-HKLSSVM with several state-of-the-art methods, including support vector regression (SVR) [43], relevance vector machine (RVM) [44], kernel extreme learning machine (KELM) [45] and deep sparse autoencoders (DSAE) [46]. The comparison results are presented in Table V, and the number of best results are recorded for both DM-HKLSSVM and the state-of-the-art methods. From these results, we find that DM-HKLSSVM is strongly superior to other state-of-the-art methods and obtains 14 of the best results for all evaluation indexes. Consequently, these statistics demonstrate that the proposed method has powerful generalization ability for different prediction problems.

D. Performance analysis of DM-HKLSSVM

To show the stability of DM-HKLSSVM in terms of different dynamic prediction problems, Fig. 9 shows a box-plot of the RMSE for 30 instances using different strategies. Each box contains information for the lower quartile, the median, and the upper quartile values. We can see that DM-HKLSSVM obtains a smaller average RMSE and performs more consistently than other strategies. On the whole, DM-HKLSSVM has good stability in dealing with different dynamic prediction problems for the BOF steelmaking production process.
TABLE VI

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Indexes</th>
<th>M-HKLSSVM</th>
<th>S-HKLSSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>RMSE</td>
<td>9.42E+00</td>
<td>3.59E+01</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>5.28E-01</td>
<td>2.05E+00</td>
</tr>
<tr>
<td></td>
<td>MAXE (°C)</td>
<td>2.51E+01</td>
<td>9.58E+01</td>
</tr>
<tr>
<td>C</td>
<td>RMSE</td>
<td>6.83E-02</td>
<td>2.81E-01</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>6.66E+00</td>
<td>2.55E+01</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>1.79E-01</td>
<td>6.63E-01</td>
</tr>
<tr>
<td>Mn</td>
<td>RMSE</td>
<td>2.10E-02</td>
<td>5.03E-02</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>1.50E+01</td>
<td>3.10E+01</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>3.92E-02</td>
<td>7.42E-02</td>
</tr>
<tr>
<td>Si</td>
<td>RMSE</td>
<td>5.92E-02</td>
<td>9.17E-02</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>4.90E+01</td>
<td>2.53E+02</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>1.87E-01</td>
<td>3.21E-01</td>
</tr>
<tr>
<td>S</td>
<td>RMSE</td>
<td>1.05E-03</td>
<td>5.03E-03</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>3.02E+01</td>
<td>7.18E+01</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>2.29E-03</td>
<td>6.84E-03</td>
</tr>
<tr>
<td>P</td>
<td>RMSE</td>
<td>7.60E-03</td>
<td>1.25E-02</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>1.92E+01</td>
<td>2.06E+01</td>
</tr>
<tr>
<td></td>
<td>MAXE (%)</td>
<td>1.92E-02</td>
<td>3.16E-02</td>
</tr>
</tbody>
</table>

TABLE VII

<table>
<thead>
<tr>
<th>Different stages</th>
<th>T</th>
<th>C</th>
<th>Mn</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-HKLSSVM (s)</td>
<td>2.45E+04</td>
<td>1.83E+04</td>
<td>1.20E+03</td>
</tr>
<tr>
<td>S-HKLSSVM (s)</td>
<td>4.87E+02</td>
<td>3.75E+02</td>
<td>3.01E+02</td>
</tr>
<tr>
<td>Different stages</td>
<td>S-HKLSSVM (s)</td>
<td>S-HKLSSVM (s)</td>
<td>M-HKLSSVM (s)</td>
</tr>
<tr>
<td></td>
<td>Si</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>S-HKLSSVM (s)</td>
<td>5.85E+03</td>
<td>9.72E+02</td>
<td>1.34E+03</td>
</tr>
<tr>
<td>M-HKLSSVM (s)</td>
<td>1.32E+02</td>
<td>2.27E+02</td>
<td>3.32E+02</td>
</tr>
</tbody>
</table>

Fig. 9. Performance comparisons of 30 instances for different dynamic prediction problems: (a) Temperature, (b) Carbon, (c) Manganese, (d) Silicon, (e) Sulfur, (f) Phosphorus.

E. Efficiency analysis of multistage modeling

According to the specific characteristics of different quality components in the BOF steelmaking process, T, C, Mn, Si, S, and P are divided into different stages. Table VI shows the performance comparison results for the different types of models using the multistage modeling strategy (M-HKLSSVM) and the single-stage modeling strategy (S-HKLSSVM). Neither M-HKLSSVM nor S-HKLSSVM consider the dynamic nature of the process. The single-stage model contains only one model, but it includes all the multistage training data. From Table VI, we can see that the predicted results of the multistage models are much better than those of the single-stage model. Because a single-stage model based on LSSVM has an advantage with a small amount of sample data, if the dataset is very large, some testing data will not be predicted accurately. However, the proposed multistage modeling strategy can utilize training data with a different reaction range to establish multiple prediction models, which improves model generalization. Moreover, we can also infer that the predicted effect will be better if the stages are better divided. However, the workload and complexity will increase with the number of stages. Therefore, based on the reaction mechanism and the data variation range, our stages are reasonably divided in this study, and achieve a satisfying result.

F. Complexity analysis

This subsection of the paper reports the results of a computational complexity analysis of M-HKLSSVM. For the HKLSSVM, it should be noted that the hybrid kernel function in single-stage modeling requires $O(n^2d)$, and the hybrid kernel function in multistage modeling requires $O(n^2d/N)$, where $n < N$, and the value of $n$ is determined based on the number of stages. We can conclude that $N$ has a large effect on HKLSSVM. From a theoretical analysis viewpoint, M-HKLSSVM effectively reduces the computational complexity.

Table VII shows time comparisons of different stages between M-HKLSSVM and S-HKLSSVM. As Table VII shows, M-HKLSSVM has smaller running time than S-HKLSSVM for different instances of the BOF steelmaking process.

V. Conclusions

This study is mainly motivated by the challenging issue of solving real-time prediction problems in the BOF steelmaking process. To this end, we propose a dynamic analytics method based on LSSVM with a hybrid kernel whose model parameters are optimized by a DE algorithm. According to the reaction mechanism and sampled data in the converter steelmaking process, a multistage modeling strategy is
designed to improve the prediction accuracy, and to reduce the model complexity. In addition, a dynamic modeling strategy based on the static model is developed to predict the temperature and product quality in real time. To verify the practicability of the proposed method, we apply it to real data collected from a BOF steelmaking production process. The results show that DM-HKLSSVM is effective and can meet actual production demand across a wide range of different instances.

The process industry includes problems that involve controlling dynamic processes. These problems can be considered dynamic black-box optimization problems. In future research, we will attempt to apply advanced intelligent optimization techniques to optimize multistage dynamic operation analytics models.

REFERENCES


Chang Liu (S’18) is currently pursuing the Ph.D. degree in systems engineering from Institute of Industrial & Systems Engineering of Northeastern University, Shenyang, China.

His research interests include data analytics and optimization for smart industry, model predictive control, machine learning, computational intelligent optimization, and engineering applications in various industrial processes.

Lixin Tang (M’09–SM’14) received the B.E. degree in industrial automation, the M.E. degree in systems engineering and the Ph.D. degree in control theory and application from Northeastern University, Shenyang, China, in 1988, 1991, 1996, respectively.

He is currently a Cheung Kong Scholars Chair Professor, the Director of the Institute of Industrial and Systems Engineering, Northeastern University. His research interests include industrial big data science, data analytics and machine learning, reinforcement learning and dynamic optimization, computational intelligent optimization, plant-wide production and logistics planning, production and logistics batching and scheduling and engineering applications in manufacturing (steel, petroleum-chemical, and nonferrous), energy, resources industry, and logistics systems. He has published papers in journals such as Operations Research, IIE Transactions, Naval Research Logistics, European Journal of Operational Research, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON POWER SYSTEMS, and IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY.


Jiyin Liu received the B.E. degree in industrial automation and the M.E. degree in systems engineering from Northeastern University, Shenyang, China, in 1982 and 1985, respectively, and the Ph.D. degree in manufacturing engineering and operations management from the University of Nottingham, Nottingham, U.K., in 1993.

He is currently a Professor of Operations Management with the School of Business and Economics, Loughborough University, Leicestershire, U.K. He is also a Cheung Kong Scholars Visiting Chair Professor with the Institute of Industrial and Systems Engineering, Northeastern University. His research interests include operations planning and scheduling problems in production, logistics and supply chains, and mathematical modeling, optimization and heuristic methods. He has published papers in journals such as European Journal of Operational Research, IEEE TRANSACTIONS, IIE Transactions, International Journal of Production Research, Naval Research Logistics, Operations Research, and Transportation Research.

Zhenhao Tang received the M.E. degree in control theory and application and the Ph.D. degree in control theory and application from Northeastern University, Shenyang, China, in 2009, 2014, respectively.

He is currently an Associate Professor of Control Theory and Application with the School of Automation Engineering, Northeast Power Electric University, Jilin, China. His research interests include model predictive control, data analytics and machine learning, and computational intelligent optimization.