

## Research Article

# Adaptive Hybrid Soft-Sensor Model of Grinding Process Based on Regularized Extreme Learning Machine and Least Squares Support Vector Machine Optimized by Golden Sine Harris Hawk Optimization Algorithm

Wei Xie, Jie-sheng Wang , Cheng Xing, Sha-Sha Guo, Meng-wei Guo, and Ling-feng Zhu

*School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China*

Correspondence should be addressed to Jie-sheng Wang; wang\_jiesheng@126.com

Received 20 March 2020; Accepted 12 May 2020; Published 28 May 2020

Academic Editor: Carlos Aguilar-Ibanez

Copyright © 2020 Wei Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Soft-sensor technology plays a vital role in tracking and monitoring the key production indicators of the grinding and classifying process. Least squares support vector machine (LSSVM), as a soft-sensor model with strong generalization ability, can be used to predict key production indicators in complex grinding processes. The traditional crossvalidation method cannot obtain the ideal structure parameters of LSSVM. In order to improve the prediction accuracy of LSSVM, a golden sine Harris Hawk optimization (GSHHO) algorithm was proposed to optimize the structure parameters of LSSVM models with linear kernel, sigmoid kernel, polynomial kernel, and radial basis kernel, and the influences of GSHHO algorithm on the prediction accuracy under these LSSVM models were studied. In order to deal with the problem that the prediction accuracy of the model decreases due to changes of industrial status, this paper adopts moving window (MW) strategy to adaptively revise the LSSVM (MW-LSSVM), which greatly improves the prediction accuracy of the LSSVM. The prediction accuracy of the regularized extreme learning machine with MW strategy (MW-RELM) is higher than that of MW-LSSVM at some moments. Based on the training errors of LSSVM and RELM within the window, this paper proposes an adaptive hybrid soft-sensing model that switches between LSSVM and RELM. Compared with the previous MW-LSSVM, MW-neural network trained with extended Kalman filter (MW-KNN), and MW-RELM, the prediction accuracy of the hybrid model is further improved. Simulation results show that the proposed hybrid adaptive soft-sensor model has good generalization ability and prediction accuracy.

## 1. Introduction

As one of the most important operating procedures in a beneficiation plant, the grinding and classifying process is to grind larger-sized metal ore to a reasonable size, expose useful metal components in the ore, and prepare for the next stage of the flotation process [1]. Some production indicators (such as granularity, impurity content, and iron content) become decisive indicators of whether the grinding process can be performed normally and will have an important impact on subsequent ore processing processes (especially flotation operations) [2]. Therefore, it is essential to monitor key production indicators in real time. Due to technical and

economic conditions and the harsh industrial environment, there is no effective sensor to monitor the variables related to product quality in real time, which may reduce industrial control performance, increase production costs, and cause obstacles to green and healthy production [3]. With the development of data acquisition technologies and data processing algorithms, data-driven models have established it as an effective tool to solve this problem by estimating online variables that are difficult to measure directly [4]. These data-driven models are often referred to as “data-driven soft sensor,” which is an important application of data analysis techniques designed to estimate difficult-to-measure variables by using easy-to-measure variables [5]. In

the past ten years, soft-sensor technology has been widely used in industrial data modeling due to its low cost, high implementation efficiency, and easy operation [6–9].

In the grinding and classifying process, many experts and scholars have made great contributions to the development of soft-sensor models for the grinding process. In the initial stage of soft-sensor modeling of the grinding process, due to the limitations of theoretical knowledge and calculation tools, some researchers (such as Casali) tried to establish a mechanism model to predict the granularity of the grinding process [10–12]. When establishing the mechanism model, the entire grinding process needs to be carefully analyzed, and a large number of algebraic and differential equations are used to describe the entire grinding process. The mechanism model needs to assume that all external conditions need to be in an ideal state. However, the actual industrial environment is complex and changeable, which will cause a large deviation in the mechanism model and the prediction accuracy will be greatly reduced. Therefore, in recent years, few researchers are willing to use mechanism soft-sensor models for grinding and classifying process. Some scholars use the idea of case-based reasoning to build soft-sensing models [13–15]. The principle of case-based reasoning is to compare the similarity between the test samples and the existing samples in the database and infer the output value of the test samples from the most similar training samples in the database. This method is simple in theory, high in implementation efficiency (less time required to calculate the output value of the test sample), strong in antiinterference, and has certain practicability. This method requires accurate classification of training samples based on their numerical characteristics. When there are many samples in the database, it is difficult to find a suitable clustering algorithm to accurately classify the samples. This inaccurate classification will cause case-based reasoning to have a large error when inferring the output of the test sample, which will reduce the prediction accuracy of the model. Similar to the case-based reasoning model, Yan et al. built a soft-sensing model of uncertainty reasoning relationships based on the cloud model [16], which is more robust, especially when part of the training data is missing and still guarantees that its prediction accuracy will not decrease significantly. However, this method still faces the problem of fuzzy membership function partitioning of the sample. When the membership of the samples is not accurate, the prediction accuracy of the model will still decrease. With the continuous progress of statistical theory, some scholars have established soft-sensor models of the grinding process using a nonlinear multiple regression model. Partial least squares (PLS) model is used by some experts in soft-sensor modeling of grinding process due to its nonlinear fitting ability [17–19]. A common disadvantage of multiple regression models is that they are particularly sensitive to changes in the external environment. When the external environment changes, the existing model parameters will not accurately reflect the current state of industrial production, which will result in a decrease in the prediction accuracy of the model and a problem of “model degradation” [20]. Tang et al. selected multiple PLS models as the soft-

sensor model on the grinding process to predict the load parameters of the ball mill [21, 22], which determines the number of PLS submodels based on multiple features of the sample set, uses an adaptive weighted fusion algorithm to integrate multiple PLS results, and obtains output so as to improve the adaptability of the soft-sensing model and the adjustment ability when the industrial status changes suddenly. However, the model parameters that need to be determined by this method will increase geometrically, which increases the computational burden of the model, and the training time of the model is also greatly increased.

Compared with other types of soft-sensor models, a neural network soft-sensor shows greater advantages in which the neural network model only needs to focus on the input and output of the model and does not need to pay too much attention to the specific industrial production process. Most neural network models have strong generalization ability. Once the number of neurons is determined, only the appropriate model parameters need to be found to track key industrial production indicators [23]. Therefore, more experts and scholars are willing to study neural network models to set up soft-sensing models for grinding and classifying process. Common neural network soft-sensing models include BP neural network [15, 24, 25], RBF neural network [26–30], support vector machine (SVM) [31, 32], and extreme learning machine (ELM) [33]. The traditional training methods of these neural networks often cannot find suitable model parameters, which makes the prediction accuracy of the model unable to be further improved. Natural heuristic algorithms provide better ideas for solving this problem. The neural network is treated as a nonlinear function, the natural heuristic algorithm optimizes the error function formed by the nonlinear function, and the model parameters corresponding to the optimal value of the error function are reasonable. Some scholars applied this kind of algorithms to find the parameters of neural network models and obtained better experimental results [34–36]. Xie et al. proposed an improved black hole algorithm (GSLBH) with strong optimization ability [37] and used GSLBH to find suitable weights and biases for ELM models with different kernel functions [23]. The experimental results show that GSLBH-ELM has better prediction ability under the condition of stable industrial status. In many industrial environments, the characteristics of the equipment and other processing behaviors frequently change, such as equipment aging, and the change of industrial raw materials [6]. This working condition change will make the optimization algorithm no longer able to find suitable structural parameters for the neural network model and the accuracy of the model is difficult to improve by this method. Wang et al. introduced the idea of model migration to wavelet neural network, which can revise structural parameters as the environment changes [38]. This method requires frequent updating of model parameters, which increases the computational burden. Dai et al. [39] proposed a robust random vector functional link network, which can maintain its generalization ability without major changes in the face of changes of the industrial status. Some other scholars have proposed a hybrid soft-sensing model combining multiple neural

networks [2, 40], whose main advantage is that it can play the prediction capabilities of different neural networks in different states so that the prediction accuracy of the model is always maintained at a high level. However, the fatal disadvantage of this model are that there are too many model parameters, the training time of the model is too long, and the working efficiency of model is extremely low, which are contrary to the efficiency of the soft-sensor model. How to make the most of the advantages of the hybrid model while ensuring the implementation efficiency of the soft sensor has become a problem to be solved.

Compared with the traditional SVM model, the least squares SVM (LSSVM) is widely used to predict key production indicators in the steel and chemical industries due to its small training volume, low training difficulty, and strong generalization ability [41]. LSSVM transforms the quadratic programming problem of SVM parameters into linear equations by constructing the constraints of the equation, which reduces the difficulty of model solving [42]. However, the two types of parameters (the penalty factor and kernel parameter) which affect the prediction accuracy of the model in LSSVM need to be set manually [43]. In the LSSVM model, the reasonable selection of penalty factors and kernel function parameters becomes the key to whether the prediction accuracy of the model can meet the requirements. The most common parameter selection method is cross-validation [44], the model parameters obtained by this method have strong limitations, and the optimal parameter combination is likely to fall far short of prediction accuracy. In contrast, the natural heuristic algorithm provides a better solution for finding the optimal model parameters of LSSVM.

The Harris Hawk optimization (HHO) algorithm is a new type of natural heuristic algorithm proposed by Mirjalili et al. [45]. This algorithm has strong exploration and weak exploitation, and it has been proved in Xie [37] that the golden sine operator can expand the search range of agents, so this paper introduces the golden sine operator to the HHO (GSHHO) when the HHO is switched to the exploitation phase, which expands exploitation range of the algorithm and enhances the exploitation of the algorithm. This paper uses GSHHO algorithm to optimize the error function of LSSVM, and the model parameters corresponding to the minimum value of the obtained error function are used as the optimal parameter of LSSVM. In order to explore the optimization ability of GSHHO algorithm for LSSVM with different kernel functions, GSHHO algorithm is used to optimize LSSVM models with linear kernel, sigmoid kernel, polynomial kernel, and radial basis kernel, and the prediction results of different LSSVMs are compared. The model with the best prediction result is currently the most ideal LSSVM soft-sensor model. Through the observation and analysis of the dataset, the industrial status selected in this paper has been in dynamic changes. Therefore, local LSSVM models are established by using moving window (MW) strategy to predict the ore granularity at different periods. This MW-LSSVM model can respond well to changes of industrial status. At the same

time, it was found that the regularized extreme learning machine (RELM) based on MW strategy (MW-RELM) also obtained similar prediction results as MW-LSSVM. In order to take advantage of different neural network soft-sensor models in different situations, based on the MW strategy, this paper proposes a hybrid soft-sensor model with LSSVM and RELM continuously switching (MW-LSSVM-RELM). Because the RELM model has the characteristics of extremely short training time, the introduction of RELM into MW-LSSVM does not affect the implementation efficiency of the soft-sensing model, and at the same time improves the prediction accuracy of the soft-sensing model. Neural network trained with extended Kalman filter (KNN) has proven to be a predictive model with strong generalization ability [46, 47]. This paper compares the experimental results of KNN with LSSVM and RELM and also compares the experimental results of MW-KNN with MW-LSSVM-RELM. Finally, simulation experiments prove that this hybrid model based on moving window strategy is more practical. The rest of this paper is organized as follows. Section 2 introduces LSSVM and RELM, Section 3 introduces GSHHO algorithm, Section 4 introduces the working principle of the hybrid model proposed in this paper, Section 5 conducts simulation experiments, and Section 6 draws conclusions.

## 2. Neural Network Soft-Sensor Models

### 2.1. Least Squares Support Vector Machine Regression Model.

The core idea of the support vector machine is to derive an estimate  $f(x)$  of the unknown relationship  $y = f(x)$  between the input variable  $x$  and the output variable  $y$  in a given training set:

$$f(x) = w^T \varphi(x) + b, \quad (1)$$

where  $x \in R^{N \times p}$ ,  $N$  is the number of samples,  $p$  is the dimension of the variables,  $f(x) \in R^{N \times 1}$ , and  $\varphi(x)$  represents a high-dimensional nonlinear mapping of  $x$ ,  $w$ , and  $b$  are constants. Let the given training sample set  $x = \{(x_1, y_1), \dots, (x_N, y_N)\}$ ,  $x_i$  and  $y_i$  are the corresponding input and output of the  $i$ th sample. The optimization problem of SVR can be expressed as follows:

$$\min J(w, \xi, \xi^*) = \frac{1}{2} w^T w + C \sum_{i=1}^N (\xi_i + \xi_i^*). \quad (2)$$

The unequal constraints of this function can be expressed as follows:

$$\begin{cases} y_i - w^T \varphi(x_i) - b \leq \varepsilon + \xi_i, \\ w^T \varphi(x_i) + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, l, \end{cases} \quad (3)$$

where  $\xi_i$  and  $\xi_i^*$  are slack variables,  $\varepsilon > 0$  is the allowable error threshold, and  $C$  is the penalty factor.

Compared with the original SVR, LSSVM regression replaces the inequality constraint with the linear least squares criterion and applies it to the loss function, whose objective function can be expressed as follows:

$$\min J(w, e) = \frac{1}{2}w^T w + \frac{1}{2}C \sum_{i=1}^N e_i^2. \quad (4)$$

At this time, the constraints of the objective function become

$$y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, 2, \dots, N. \quad (5)$$

The following Lagrange function can be constructed from mathematical theoretical knowledge:

$$L(w, b, e, \alpha) = \psi(w, e) - \sum_{i=1}^N \alpha_i (w^T \varphi(x_i) + b + e_i - y_i), \quad (6)$$

where  $\alpha_i (i = 1, 2, \dots, N)$  is the Lagrangian multiplier. So, the optimal KKT condition of  $L(w, b, e, a)$  can be expressed as follows:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \implies w = \sum_{i=1}^N \alpha_i \varphi(x_i), \\ \frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^N \alpha_i = 0, \\ \frac{\partial L}{\partial e_i} = 0 \implies \alpha_i = C e_i, \quad i = 1, 2, \dots, N, \\ \frac{\partial L}{\partial \alpha_i} = 0 \implies w^T \varphi(x_i) + b + e_i - y_i = 0, \quad i = 1, 2, \dots, N. \end{cases} \quad (7)$$

By calculating the variables  $w$  and  $e_i$ , the optimization problem can be transformed into the following linear equations:

$$\begin{bmatrix} 0 & 1_N^T \\ 1_N & \Omega + C^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (8)$$

where  $1_N = [1, \dots, 1]^T$ ,  $\alpha = [\alpha_1, \dots, \alpha_N]^T$ , and  $y = [y_1, \dots, y_N]^T$ . The matrix  $\Omega = \{\Omega_{ij}\}_{N \times N}$  should satisfy the Mercer condition, that is to say,

$$\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j). \quad (9)$$

After solving the equations shown in equation (8), the final result of LSSVM regression can be expressed as follows:

$$f(x) = \sum_{i=1}^N \alpha_i K(x, x_i) + b, \quad (10)$$

where  $\alpha_i$  and  $b$  are obtained by solving the linear equations in equation (8). In equation (10),  $K(x, x_i)$  is a kernel function that satisfies the Mercer condition. The kernel functions commonly used in LSSVM are shown in Table 1 [48], which give the parameters that LSSVM needs to be manually adjusted in the case of different kernel functions.  $C$  in Table 1 is the penalty factor for LSSVM.

The method of adjusting these parameters is generally crossvalidation. This method has a simple idea and is easy to operate. However, the limitation of this method is large, and the obtained structural parameters are probably not optimal solutions. Natural heuristic algorithm provides a good solution for finding the optimal parameters of the model. It works by optimizing the error function of the model, and the parameter corresponding to the minimum value of the obtained error function is the optimal parameter. Commonly used algorithms include genetic algorithm (GA) and particle swarm optimization (PSO) algorithm, but a large number of experiments have proved that many algorithms have better optimization capabilities than GA and PSO [37, 45, 49]. This paper proposes a golden sine Harris Hawk optimization algorithm to select reasonable structural parameters for the LSSVM soft-sensor model. This paper will use the golden sine Harris Hawk algorithm to optimize the LSSVM with the four kernel functions in Table 1 and compare the experimental results of these four LSSVMs. The specific implementation steps are described in the following sections.

## 2.2. Regularized Extreme Learning Machine Regression Model.

Extreme learning machine (ELM) is a single hidden layer feed-forward network (SLFN) proposed by Huang et al. [50]. This neural network has a simple structure, extremely short training time, efficient execution, and good generalization ability [4, 23, 51, 52]. Principles of ELM training and prediction have been explained in details in [23]. The key work of training ELM is to solve the weight  $\beta$  between the hidden layer and the output layer, and the calculation of  $\beta$  is realized by

$$\beta = H^+ T', \quad (11)$$

where  $H^+$  is the Moore–Penrose generalized inverse of the hidden layer output matrix  $H$ ,  $H^+$  can also be expressed as  $(H^T \cdot H)^{-1} \cdot H^T$ , and  $T$  represents the output matrix. After solving the structural parameters, the trained model can be used to predict the output variables. The main disadvantage of ELM is its instability, it is too sensitive to the interference of data noise, and it is prone to overfitting. To solve this difficulty, Deng et al. [53] introduced regularization parameters when calculating the weight  $\beta$ , which greatly enhanced the generalization ability of the ELM model and made the ELM model more practical.

According to statistical learning theory, the actual predicted risk of learning is composed of empirical risk and structural risk. A model with good generalization capabilities should make the best trade-off between the two risks. Therefore, the actual risk can be expressed by the weighted sum of these two risks. By introducing a weighting factor  $\gamma$  for empirical risks, their ratio can be adjusted [53]. Empirical risk can be expressed by squared error  $\|e\|^2$ . Structural risk is represented by variables  $\|\beta\|^2$  that maximize the distance to the interface [54–56]. The mathematical model of the regularized ELM model can be expressed as follows:

TABLE 1: Kernel functions used by LSSVM.

Kernel function	Expression	Parameters to be adjusted
Linear	$K(x, x_i) = x_k^T x$	$C$
Polynomial	$K(x, x_i) = (x_k^T x + \theta)^2$	$C, \theta$
Sigmoid	$K(x, x_i) = \tanh(kx_k^T x)$	$C, k$
RBF	$K(x, x_i) = \exp(-\ x - x_k\ _2^2 / \sigma^2)$	$C, \sigma^2$

$$\begin{cases} \min & \left( \frac{1}{2} \|\beta\|^2 + \frac{1}{2} \gamma \|\varepsilon\|^2 \right) \\ \text{s.t.} & \sum_{i=1}^N \beta_i g(w_i x_i + b_i) - t_j = \varepsilon_j, \\ & j = 1, 2, \dots, N, \end{cases} \quad (12)$$

where  $\tilde{N}$  is the number of samples,  $N$  is the number of neurons, and  $\varepsilon = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N]$ . To solve the optimization problem shown in equation (12), the following Lagrangian equation can be constructed:

$$\begin{aligned} L(\beta, \varepsilon, \alpha) &= \frac{\gamma}{2} \|\varepsilon\|^2 + \frac{1}{2} \|\beta\|^2 - \sum_{j=1}^N \alpha_j \left( \sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) - t_j - \varepsilon_j \right) \\ &= \frac{\gamma}{2} \|\varepsilon\|^2 + \frac{1}{2} \|\beta\|^2 - \alpha (H\beta - T - \varepsilon), \end{aligned} \quad (13)$$

where  $\alpha_j$  is the Lagrangian multiplier. The optimal KKT condition of  $L(\beta, \varepsilon, \alpha)$  can be expressed as follows:

$$\begin{cases} \frac{\partial L}{\partial \beta} = 0 \implies \beta^T = \alpha H, \\ \frac{\partial L}{\partial \varepsilon} = 0 \implies \gamma \varepsilon^T + \alpha = 0, \\ \frac{\partial L}{\partial \alpha} = 0 \implies H\beta - T - \varepsilon = 0. \end{cases} \quad (14)$$

Three sets of equations in equation (14) can be combined to calculate the final expression of  $\beta$ :

$$\beta = \left( \frac{I}{\gamma} + H^T H \right)^{-1} H^T T. \quad (15)$$

The ELM that obtains the weight  $\beta$  between the hidden layer and the output layer through equation (15) is called a regularized ELM (RELM). Adjustment of  $\gamma$  can adjust the ratio between empirical risk and structural risk. The model will obtain the best generalization ability when these two risks reach the optimal compromise [53]. Therefore, compared with traditional ELM, RELM has better anti-interference ability, stronger generalization ability of neural network, and higher prediction accuracy. The implementation steps of RELM are described as follows:

Step 1: determine the number of hidden layer neurons  $N$  of the RELM model according to the size of the samples, and determine the activation function  $g(x)$ .

Step 2: given a training set of samples, randomly initialize weights  $w_i (i = 1, 2, \dots, \tilde{N})$  and biases values  $b_i (i = 1, 2, \dots, \tilde{N})$  between the input layer and the hidden layer and calculate the output matrix  $H$  of the hidden layer.

Step 3: calculate the weight  $\beta$  between the hidden layer and the output layer according to equation (15).

Step 4: input the query set into the RELM model to obtain the output.

### 3. Improved Harris Hawk Optimization Algorithm

**3.1. Harris Hawk Optimization Algorithm.** The Harris Hawk optimization (HHO) algorithm is a new natural heuristic algorithm proposed by Mirjalili et al. inspired by the hunting behavior of Harris Hawk [23]. It has the characteristics of strong exploration and extremely high efficiency. Therefore, once HHO algorithm was proposed, it attracted the attention of many scholars [57, 58]. According to the current physical energy consumption of the prey, Harris Hawk will adopt different hunting strategies to capture the prey. Let energy of the current prey be  $E$ , and it can be calculated by

$$E = 2E_0 \left( 1 - \frac{t}{T} \right), \quad (16)$$

where  $E_0$  is a random number between  $(-1, 1)$ ,  $t$  is the current number of iterations, and  $T$  is the specified total number of iterations. When  $E_0$  is between  $-1$  and  $0$ , prey is physically flagging. When  $E_0$  is between  $0$  and  $1$ , prey is strengthening. It can be seen from equation (16) and Figure 1 that as  $t$  increases, the prey energy  $E$  decreases. Harris Hawk will judge whether it should keep exploration phase or switch to the exploitation phase based on the  $E$ .

When  $|E| \geq 1$ , the prey has more energy, the Harris Hawk is in the exploration phase, and the position update strategy in the exploration phase can be realized by

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2 X(t)|, & q \geq 0.5, \\ \left( X_{\text{pery}}(t) - X_m(t) \right) - r_3 (\text{LB} + r_4 (\text{UB} - \text{LB})), & q < 0.5, \end{cases} \quad (17)$$

where LB and UB represent upper and lower bounds,  $r_1, r_2, r_3, r_4$ , and  $q$  are random numbers between  $0$  and  $1$ ,  $X(t)$  and  $X(t+1)$  represent the current location of the Harris Hawk and the location of the next iteration, respectively,  $X_{\text{rand}}(t)$  represents the current position of an agent in the Harris Hawk population,  $X_{\text{pery}}(t)$  represents the current position of the prey (the current optimal location),

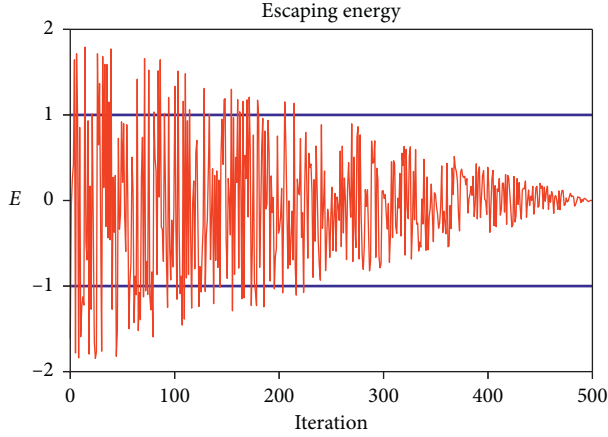


FIGURE 1: Energy change curve of the prey.

and  $X_m(t)$  represents the coordinate average of all agents in the Harris Hawk population, whose calculation formula is described as follows:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t), \quad (18)$$

where  $N$  is the population size.

When  $|E| < 1$ , Harris Hawk transforms into the exploitation phase and began to capture the prey. Scientific research shows that Harris Hawk has seven ways to attack the prey [59]. In HHO, only four methods of attacking prey were selected: “Soft besiege,” “Hard besiege,” “Soft besiege with progressive rapid dives,” and “Hard besiege with progressive rapid dives.” A random number  $r \in (0, 1)$  is also defined in the process of the eagle attacking the prey, which indicates whether the prey successfully escaped.

In the exploitation phase, when  $r \geq 0.5$  and  $|E| \geq 0.5$ , the prey has enough energy, but still fails to escape. At this time, the Harris Hawk attacked the prey with a “soft besiege” method. The movement of Harris Hawk is expressed as follows:

$$\begin{cases} X(t+1) = \Delta X(t) - E|JX_{\text{prey}}(t) - X(t)|, \\ J = 2(1 - r_5), \\ \Delta X(t) = X_{\text{prey}}(t) - X(t), \end{cases} \quad (19)$$

where  $r_5$  is a random number between 0 and 1. The other parameters have the same meaning as those in equation (17).

When  $r \geq 0.5$  and  $|E| < 0.5$ , the prey has no enough energy to escape and the Harris Hawk uses a “hard besiege” method to attack the prey. The movement of Harris Hawk can be expressed as follows:

$$X(t+1) = X_{\text{prey}}(t) - E|\Delta X(t)|, \quad (20)$$

where  $\Delta X(t)$  has the same meaning as the parameter in equation (19).

When  $r < 0.5$  and  $|E| \geq 0.5$ , the prey is difficult to be caught by the Harris Hawk. Harris Hawk uses a more clever method “Soft besiege with progressive rapid dives” to capture the prey. The movement pattern of this attack method is described as follows:

$$\begin{cases} Y = X_{\text{prey}}(t) - E|JX_{\text{prey}}(t) - X(t)|, \\ Z = Y + S \times \text{Levy}(D), \end{cases} \quad (21)$$

where  $J$ ,  $X_{\text{prey}}$ , and  $X(t)$  have the same meaning as in equation (19),  $D$  represents the dimension of the agents, and  $S$  represents a  $1 \times D$  random matrix, and Levy means Levy flight function. The laws of movement of most animals in nature conform to the Levy flight function [60], and the expression of the Levy flight function is expressed as follows:

$$\text{LF}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}, \quad (22)$$

$$\sigma = \left( \frac{\Gamma(1 + \beta) \times \sin(\pi\beta/2)}{\Gamma(1 + \beta/2) \times \beta \times 2(\beta - 1/2)} \right)^{1/\beta},$$

where  $u$  and  $v$  are a random number between 0 and 1 and the value of  $\beta$  defaults to 1.5. In the end, the Harris Hawk determined the movement method of the next iteration according to the fitness value of the objective function as follows:

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)), \\ Z, & \text{if } F(Z) < F(X(t)), \end{cases} \quad (23)$$

where  $F(x)$  is the calculated fitness function value, and other parameters have the same meanings as the parameters in equation (21).

When  $r < 0.5$  and  $|E| < 0.5$ , although the prey is exhausted, it still tries to escape the capture of Harris Hawk. At this time, Harris Hawk uses “Hard besiege with progressive rapid dives”. The movement pattern of this attack method is expressed as follows:

$$\begin{cases} Y = X_{\text{prey}}(t) - E|JX_{\text{prey}}(t) - X_m(t)|, \\ Z = Y + S \times \text{Levy}(D), \end{cases} \quad (24)$$

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)), \\ Z, & \text{if } F(Z) < F(X(t)), \end{cases} \quad (25)$$

where  $X_m(t)$  has the same meaning as in equation (17), and the remaining parameters have the same meaning as in equation (21). Similar to “Soft besiege with progressive rapid dives”, in the end, the Harris Hawk determines the movement pattern of the next iteration according to the fitness value of the objective function shown in equation (25). The implementation flowchart of Harris Hawk optimization algorithm is shown in Figure 2.

### 3.2. Golden Sine Harris Hawk Optimization Algorithm

**3.2.1. Golden Sine Operator.** The golden sine operator is a search strategy derived from the golden sine algorithm (Gold-SA) [60]. This operation can expand the search range of agents. From the geometric meaning of the trigonometric function, it can be known that the process of continuously moving a point on a sine function is equivalent to the process of continuously scanning this point on a unit circle, but in Gold-SA, agents do not move according to a standard sine

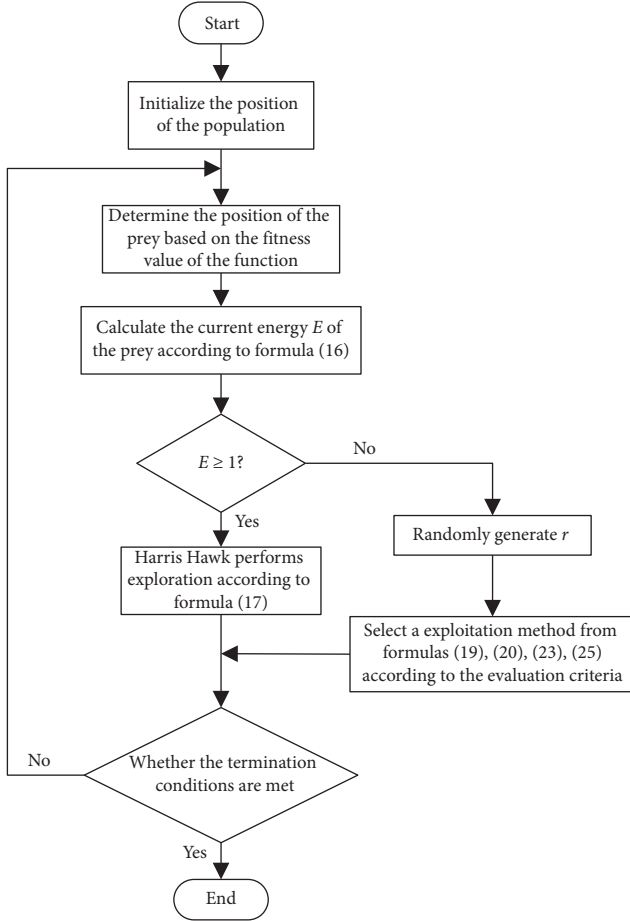


FIGURE 2: Procedure of HHO algorithm.

curve, but move according to a gold sine curve [37]. During each iteration, the ability that agents search within a certain local improved is expanded, when the agents move according to this golden sine curve. This golden sine curve path shape in two-dimensional space is shown by the red curves in Figure 3 [37].

In Gold-SA, the agents to search according to the golden sine curve can be expressed as follows:

$$X(t+1) = X(t) * |\sin(r_1)| - r_2 * \sin(r_1) * |m_1 * X_{\text{best}}(t) - m_2 * X(t)|, \quad (26)$$

where  $r_1$  is a random number between 0 and  $2\pi$ ,  $r_2$  is a random number between 0 and  $\pi$ ,  $X_{\text{best}}(t)$  is the current optimal location, and  $m_1$  and  $m_2$  is the coefficient obtained by the gold selection method. This method can reduce the search space of the agents, improve the search efficiency of the agents, and make the agents move to the target location faster. In order to simplify the operation and improve the execution efficiency of the algorithm, Xie et al. [37] assign  $m_1$  and  $m_2$  in equation (26) with suitable constants. This approach guarantees that the search ability of the agents will not weaken, while increasing the stability of the

algorithm. The constant values given to  $m_1$  and  $m_2$  are set as follows:

$$\begin{aligned} m_1 &= -\pi + (1 - \tau), \\ m_2 &= -\pi + \tau * \pi, \\ \tau &= \frac{1 - \sqrt{5}}{2}, \end{aligned} \quad (27)$$

when  $m_1$  and  $m_2$  in equation (25) becomes constants in equation (27), the movement mode shown in equation (26) becomes the golden sine operator.

**3.2.2. Golden Sine Harris Hawk Optimization (GSHHO) Algorithm.** HHO is an algorithm with very strong exploration. Its global search mechanism can effectively reduce the possibility of the algorithm falling into a local optimum. However, when HHO enters the exploitation phase, its exploitation is not particularly strong. In order to expand the exploitation range of HHO, this paper introduces the golden sine operator into the exploitation phase of HHO to enhance its exploitation. During each iteration, when  $|E| < 1$ , all agents first perform the golden sine operation in equation (26) and then select one of four exploitation methods for the next operation according to the size of  $r$  and  $E$ . Compared with the implementation steps of the original HHO in Figure 2, GSHHO only adds a golden sine search operation to the exploitation phase, but the exploitation of the algorithm has been greatly improved. The pseudocode of GSHHO algorithm is described as follows:

Determine the number of populations as  $N$  and the number of iterations as  $T$ ;

Initialize population  $X_i (i = 1, 2, \dots, N)$ ;

while  $t < T$

Determine the optimal position according to the fitness value of all agents, and specify this optimal position as the position of the current prey;

Calculate the value of  $E$  according to formula (16);

if  $|E| \geq 1$

The agents perform exploration according to formula (17);

if  $|E| < 1$

The agents perform golden sine search according to formula (26);

if  $r \geq 0.5, |E| \geq 0.5$

The agents perform exploitation according to formula (19);

else if  $r \geq 0.5, |E| < 0.5$

The agents perform exploitation according to formula (20);

else if  $r < 0.5, |E| \geq 0.5$

The agents perform exploitation according to formula (23);

else if  $r < 0.5, |E| < 0.5$

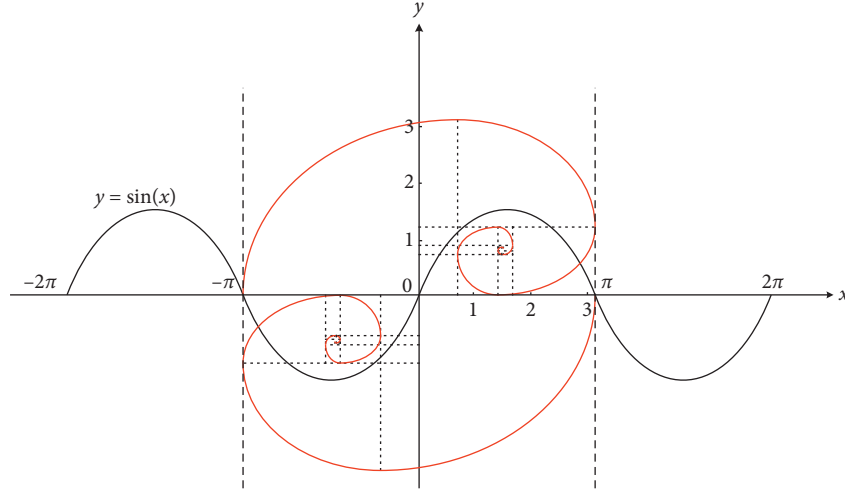


FIGURE 3: Golden sine curve in two-dimensional space.

The agents perform exploitation according to formula (25)

$$t = t + 1$$

Output the optimal value and optimal location found by HHO

**3.2.3. Golden Sine Algorithm to Optimize LSSVM Model.** As mentioned at the end of Section 2.1, using natural heuristics algorithm to select the structural parameters of LSSVM is a good idea. This paper adopts GSHHO algorithm to optimize the structural parameters of LSSVM with 4 kernel functions listed in Table 1. It can be seen from Table 1 that when the kernel function of LSSVM is different, the parameters to be adjusted will also be different, and the different parameters are the internal parameters of the kernel function. For each different kernel function, the parameter set to be adjusted is  $\Theta$ . For example, when the kernel function is *rbf* function,  $\Theta = \{C, \sigma^2\}$ . Let the error function formed by LSSVM be

$$\text{error} = \frac{1}{n} \sum_{i=1}^n (f(X_i, \Theta) - Y_i), \quad (28)$$

where  $n$  represents the number of training samples,  $X_i$  represents the input of the training samples,  $Y_i$  represents the actual output corresponding to the training sample  $X_i$ , and  $f$  represents the unknown nonlinear relationship derived from input  $X_i$  and output  $Y_i$  in the LSSVM model.

When using GSHHO algorithm, the size of  $\Theta$  determines the dimensions of the solution space, and error function is the objective function that GSHHO algorithm needs to be optimized. The smaller the objective function value found in the solution space, the smaller the training error of the LSSVM model and the higher the prediction accuracy of model for the query samples. When the iteration is completed, the optimal position obtained is optimal  $\Theta$  found by GSHHO algorithm. When the different kernel functions listed in Table 1 are optimized, the results that GSHHO optimize LSSVM models will definitely be different. In the subsequent simulation experiments, this paper will compare

the prediction results of GSHHO-LSSVM models with different kernel functions and find the optimal GSHHO-LSSVM model and the corresponding optimal parameters. The whole process of GSHHO-LSSVM model predicting query sample is divided into the following steps:

Step 1: construct the error function using the training sample set  $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$  and any given  $\Theta$  according to the function construction rule of  $f$  in Section 2.1

Step 2: find the minimum value of the error function and the coordinate  $\Theta_{\text{best}}$  corresponding to the minimum value according to the GSHHO algorithm implementation rule shown in Section 3.2.2

Step 3: after the iteration, output the optimal position  $\Theta_{\text{best}}$  and use  $\Theta_{\text{best}}$  as the optimal model parameter of the LSSVM

Step 4: train the LSSVM model again using the optimal parameter  $\Theta_{\text{best}}$  and the training sample set  $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$

Step 5: input the test samples to the trained LSSVM model to obtain prediction results

## 4. Adaptive Hybrid Soft-Sensor Model Based on LSSVM and RELM

**4.1. Moving Window Strategy.** In the actual grinding and classifying process, changes in various external conditions will cause the industrial process to change at any time (such as different types of ore, aging of processing equipment, and frequent switching of equipment working modes), and historical databases cannot contain all possible states and conditions in the future industrial process [61]. When the above problems occur, the existing soft-sensor models can no longer accurately describe the current status of industrial production, which will cause the prediction accuracy of the model to decrease. In order to cope with the frequent changes of industrial status, this paper introduces moving window (MW) strategy to adaptively revise the parameters of the soft-sensor model. The idea of MW strategy was



proposed by Michalski et al. [62], and its main work is to select a specific set of data to adjust the parameters of the soft-sensor model. In the vast majority of cases, a fixed amount of the latest data is considered the most relevant to the current state of the industry [20]. Therefore, when predicting the current production indicators, some training samples that are most relevant to the current moment can be selected from the database to build a local soft-sensor model, and the prediction value of such a local model for the current production indicator is likely to be closest to the real value. The implementation process of the MW strategy is shown in Figure 4.

Suppose the size of the window is  $L$  and the moving step is  $S$ , the training set in the window at the initial moment is

$$D_{\text{ori}} = \{(x_1, y_1), \dots, (x_L, y_L)\}. \quad (29)$$

Model  $y = f_{\text{ori}}(x)$  is trained according to  $D_{\text{ori}}$ , and the predicted value of a test sample  $t_1$  that is closest to  $D_{\text{ori}}$  in time is

$$\hat{p}_1 = f_{\text{ori}}(t_1). \quad (30)$$

When new samples are added, the window will slide forward according to the step size  $S$ . At this time, the training set in the window can be expressed as follows:

$$D_1 = \{(x_{1+s}, y_{1+s}), \dots, (x_{L+s}, y_{L+s})\}. \quad (31)$$

Model  $y = f_1(x)$  is trained according to  $D_1$ , the predicted value of a test sample  $t_2$  that is closest to  $D_1$  in time is

$$\hat{p}_2 = f_{\text{ori}}(t_2). \quad (32)$$

The model training phase of the MW strategy can be repeatedly applied to the online phase without the need for other online algorithms, which means that it can be combined with all existing soft-sensor models [20] and these advantages also make the MW strategy become a very popular adaptive strategy.

**4.2. Hybrid Soft-Sensor Model Based on Moving Window Strategy.** Section 3.2.3 details the process of GSHHO algorithm to optimize LSSVM. Compared with the traditional crossvalidation method, GSHHO algorithm can find more suitable parameters for LSSVM, which further improves the prediction accuracy of the LSSVM soft-sensor model. However, in the face of changes in the working conditions, the ability of algorithm to optimize LSSVM has a certain limit. Because a hyperplane formed by single LSSVM model cannot characterize all industrial status in the training sample, this will cause a bottleneck in improving the accuracy of the soft-sensor model. MW strategy can effectively solve this problem. Based on the optimal LSSVM structural parameters discovered by GSHHO algorithm, the introduction of MW strategy into LSSVM will further improve the model prediction accuracy. When a current query sample is needed to be predicted, LSSVM selects the window data closest to the query sample to build a local model and uses this local model to predict the query sample. When the window slides  $m$  times,  $m$  local models of LSSVM will be

established, which also means that  $m$  hyperplanes are used to characterize the industrial status at different time periods, and the hyperplanes of these local models can more realistically reflect the industrial conditions of certain periods and improve the prediction accuracy of the models. The specific implementation process of MW-LSSVM is as described in formulas (29)–(23) in Section 4.1. However, the MW strategy has a major drawback: the computational burden increases when training the model. If the window slides  $m$  times, the soft-sensing model needs to be retrained  $m$  times, so it is necessary to maintain a balance between improving the prediction accuracy of the model and reducing the calculation burden. For the regression model of SVM, one of its biggest advantages is that the training time of the model is very short after the structural parameters are determined. LSSVM is more simplified and training time is shorter than traditional SVM training process. Therefore, although the calculation load of LSSVM is increased after the introduction of MW strategy, it is far less than the acceptable training time. This advantage of LSSVM further enhances the utility of MW-LSSVM. It is found that the RELM model can obtain good prediction results similar to the MW-LSSVM model after the MW strategy is introduced. In order to further improve the prediction accuracy of MW-LSSVM, this paper proposes a hybrid model of LSSVM and RELM based on MW (MW-LSSVM-RELM), which can take advantage of LSSVM and RLEM under different conditions. The implementation mechanism of this hybrid model is shown in Figure 5.

When using this hybrid model, the computational burden of the model will further increase, but RELM is also a model with extremely high execution efficiency [53]. Although the data in each window requires training of the two models, the time required to train two models is still very short, much less than the acceptable time. The computational efficiency of the hybrid model will be explained later in the simulation experiment section. The operation of this hybrid soft-sensor model is described as follows:

Step 1: establish local LSSVM model1 and RELM model2 according to the training samples in the window

Step 2: compare the training errors of model1 and model2, and select a model with a smaller training error from these two local models as the current soft-sensor model

Step 3: predict query samples with the model identified in Step 2

Step 4: when a new sample is added, the window slides according to the specified step size, and then repeat the operations of Steps 1–3

## 5. Simulation Experiments and Result Analysis

**5.1. Technique of Grinding and Classifying Process.** Based on the processing of iron ore in a beneficiation plant, this paper studies the grinding and classifying process of iron ore. Xie et al. have introduced the grinding classification process in

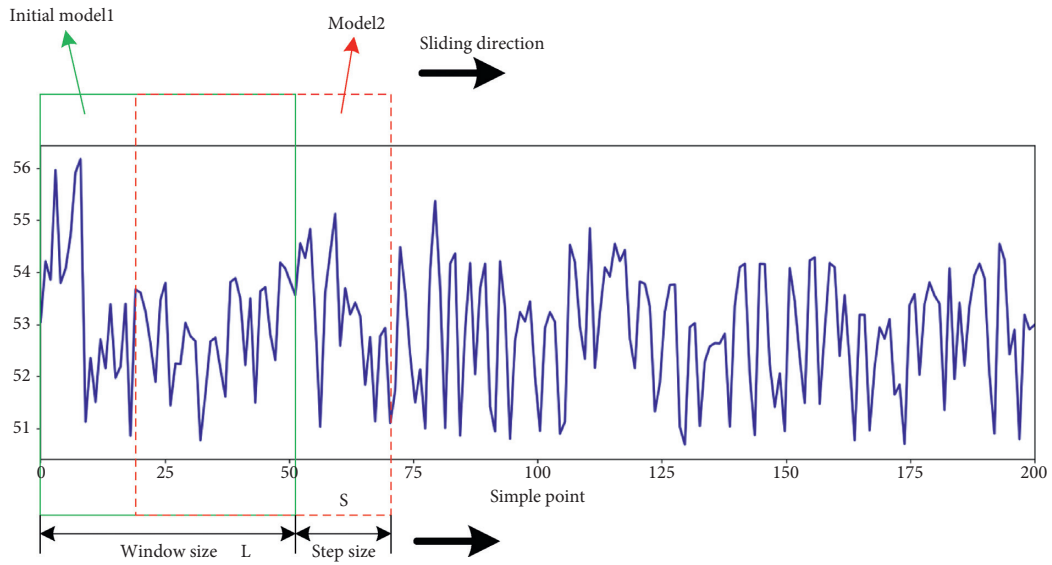


FIGURE 4: Schematic diagram of moving window.

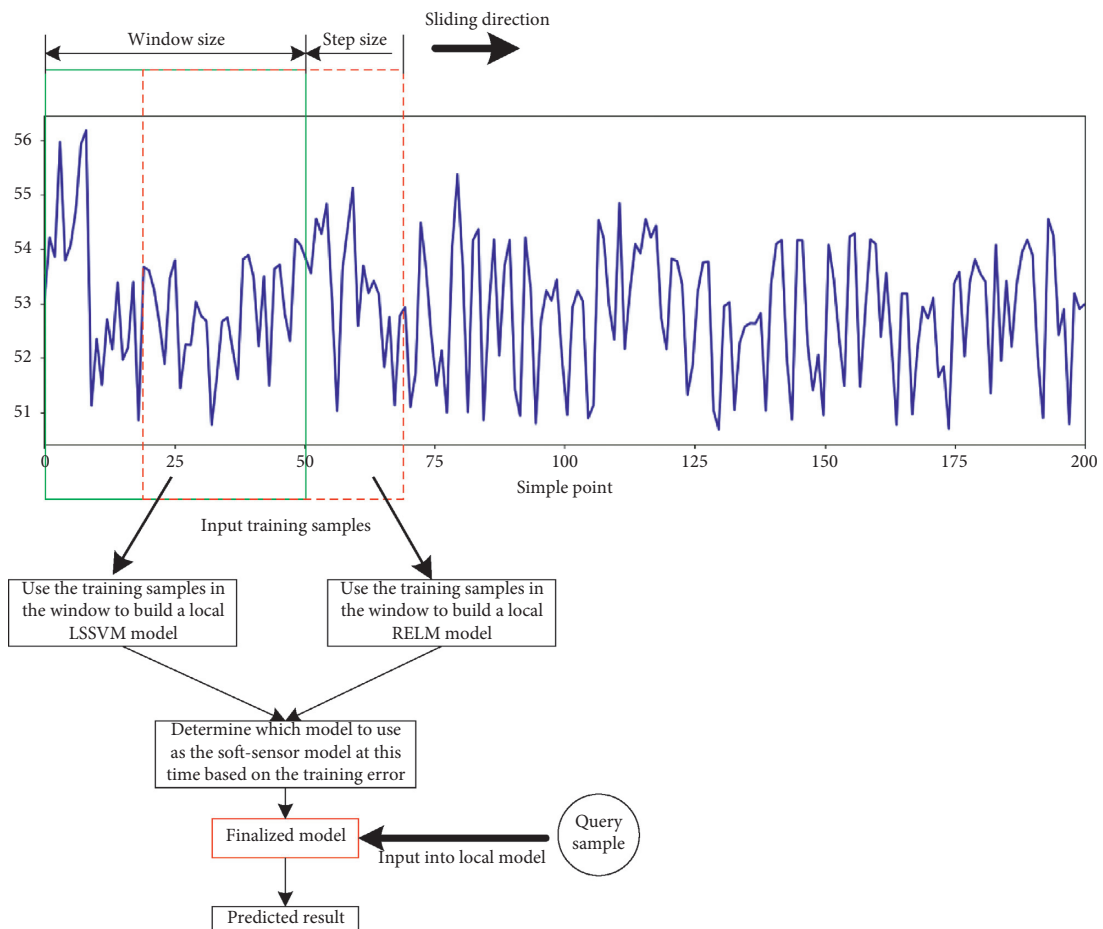


FIGURE 5: Schematic diagram of the implementation of the hybrid soft-sensor model.

details [23], and this article only briefly describes the grinding and classifying process, whose schematic diagram of the grinding and classifying process is shown in Figure 6 [23].

The grinding and classifying process is mainly composed of two parts: the first part is the first stage of closed-circuit grinding process, and the second part is the second stage of closed-circuit grinding process. The iron ore is first sent to

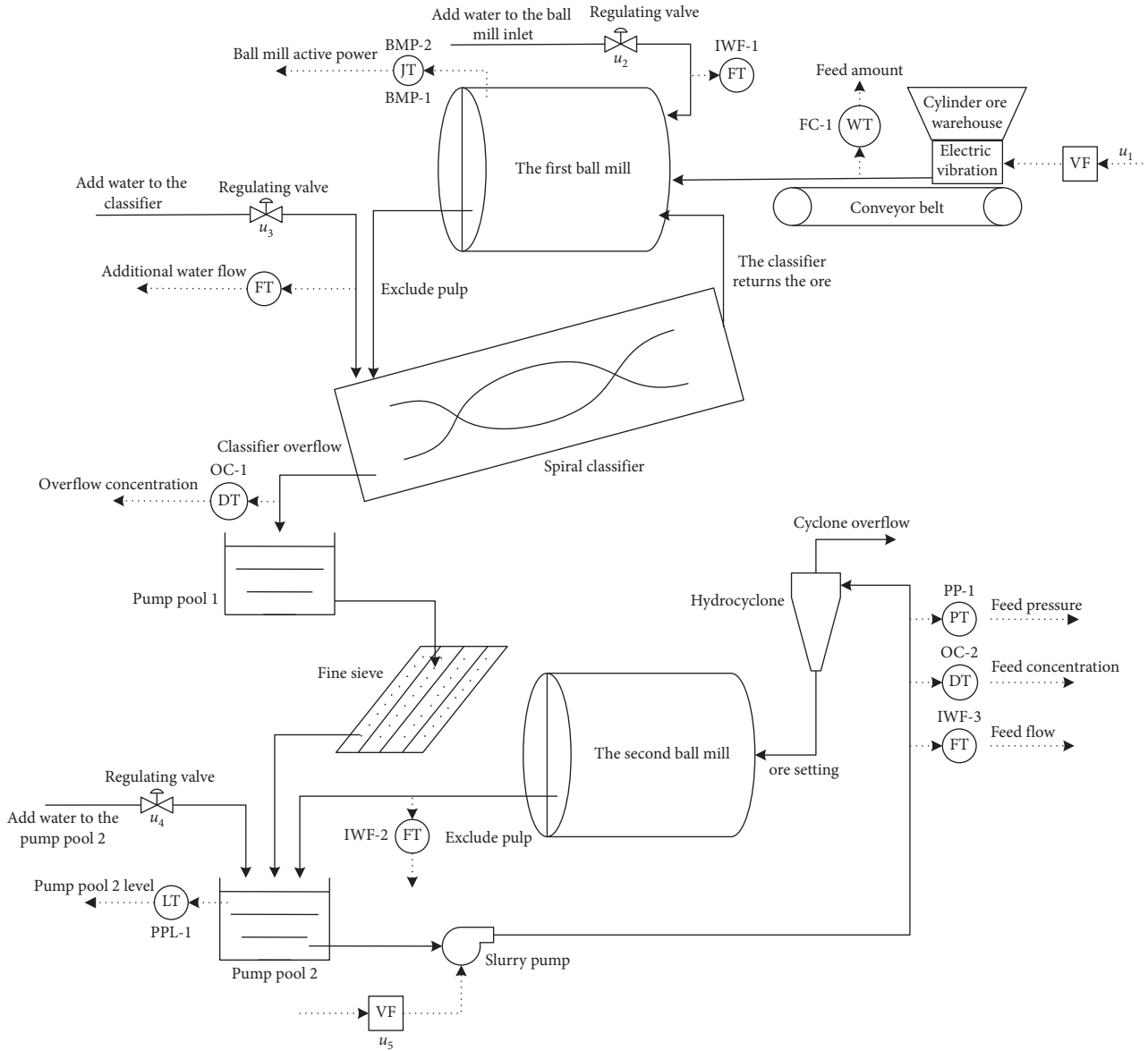


FIGURE 6: Schematic diagram of grinding and classification process.

the first ball mill for grinding through a belt. After a period of grinding, the pulp produced by the first ball mill is sent to the spiral classifier to classify the ore. The ore with larger granularity (size of the ore) is sent to the first ball mill for grinding again; the ore with smaller granularity flows out with the overflow product and then enters the second stage ball mill for fine grinding after fine sieve work. The hydrocyclone will separate the ore particles whose granularity meets the production standards. For the particles that do not meet the requirements, they will enter the second stage ball mill to continue grinding until the size of the ore meets the standards. Similar to Ref. [23], this paper still selects 10 easy-to-measure variables as secondary variables for the soft-sensor model and adopts these variables to predict the ore granularity during the grinding process. Information of secondary variables and objective variables are shown in Tables 2 and 3 [23]. Table 3 shows some data of

the grinding process soft-sensor modeling. The collection positions of these secondary variables have been indicated in Figure 6.

**5.2. Data Dimension Reduction.** For the neural network soft-sensor model, too large dimension of the input variables makes the network topology very large and the training process becomes complicated. At the same time, there may be redundancy between the variables, which will interfere with the prediction process of the neural network, so high-dimensional data information needs to be reduced [23, 63]. In the case of unstable industrial status, this paper also uses the KPCA dimensionality reduction method used in Ref. [23] to reduce the data in Table 3, and the results are listed in Table 4. The contribution percentage of the first 5 variables in Table 4 reached 85%. The input data after KPCA

TABLE 2: Information on various variables of the grinding process.

Variable type	Variable	Variable symbol	Unit
Secondary variable	Feeding capacity	FC-1	kg
	Inlet water flow	IWF-1	m <sup>3</sup> /s
	Outlet water flow	IWF-2	m <sup>3</sup> /s
	Pump pool level	PPL-1	m
	Pump pressure	PP-1	kpa
	Current	BMP-1	A
	Feed flow of cyclone	IWF-3	m <sup>3</sup> /s
	Overflow concentration	OC-1	%
	Ball mill power	BMP-2	kw
	Feed concentration	OC-2	%
Objective variable	Granularity		%

TABLE 3: Soft-sensor modeling data for the grinding process.

	Secondary variable										Objective variable
	Feeding capacity	Inlet water flow	Outlet water flow	Pump pool level	Pump pressure	Current	Feed flow of cyclone	Overflow concentration	Ball mill power	Feed concentration	Granularity
1	134.796	81.35	62.253	1.584	0.042	63.000	128.156	74.326	1119.311	70.364	58.621
2	134.578	82.547	63.424	1.609	0.043	58.000	127.717	73.325	1118.638	69.314	58.992
3	133.633	82.465	63.718	1.595	0.042	66.000	127.911	74.364	1121.663	69.415	57.694
4	135.702	82.475	64.687	1.592	0.041	63.000	127.561	74.631	1118.561	67.314	58.323
5	135.412	82.457	62.413	1.617	0.041	60.000	127.454	75.254	1119.878	68.141	59.922
6	134.794	83.347	63.081	1.606	0.043	59.000	128.932	73.344	1118.749	70.564	54.773
...	...	...	...	...	...	...	...	...	...	...	...
1700	136.364	82.145	84.341	1.534	0.043	60.000	126.347	72.384	1120.379	70.364	56.314

TABLE 4: Contribution percentage of the principal components.

$d$	Variance percentage (%)	Cumulative variance percentage (%)
1	0.3335	0.3335
2	0.2132	0.5467
3	0.1832	0.7299
4	0.0821	0.812
5	0.0501	0.8621
6	0.0482	0.9103
7	0.0413	0.9516
8	0.0322	0.9838
9	0.0091	0.9929
10	0.0071	1

dimension reduction can improve the calculation efficiency of the soft-sensor model and reduce the complexity of the model.

**5.3. Simulation Experiments.** In this paper, five input variables after KPCA processing are used to predict the content of ore granularity to 200 mesh during the grinding and classifying process. This paper collected 1,700 samples from an existing database, of which 1500 were used as training samples and 200 were used as test samples. In order to make it clear that the prediction accuracy of the soft-sensor models proposed in this paper is higher than the prediction accuracy

of other models, three quantitative indicators were selected to describe the prediction accuracy of the soft-sensor model, whose calculation methods of these three quantitative indicators are listed in Table 5. Many scholars use these three quantitative indicators to evaluate the predictive performance of the model [24, 25, 29, 30]. In Table 5,  $n$  represents the number of test samples,  $\hat{y}_i$  represents the predicted value of the objective variables,  $y_i$  represents the true value of the objective variables, and  $\bar{y}$  represents the average of all objective variables. The smaller MAE and RMSE, the higher the prediction accuracy of the model. The larger  $R^2$ , the stronger the fitting ability of the soft-sensor model.

**5.3.1. GSHHO Algorithm to Optimize LSSVM Model.** In order to study the optimization capability of GSHHO algorithm for LSSVM with different kernel functions, this paper uses GSHHO algorithm to optimize the four different LSSVMs listed in Table 1. The traditional three-layer crossvalidation method (CV) and HHO-optimized LSSVM were used as comparative experimental results to prove the superiority of GSHHO algorithm in this paper. The population of GSHHO algorithm and HHO algorithm is set to 30 and the number of iterations is set to 100. Figures 7–10 show the results that these three methods optimize the LSSVM parameters, where (a) of each figure represents the prediction results of the soft-sensor model, and (b) represents the prediction error of the soft-sensor model. In order

TABLE 5: Quantitative indicators of soft-sensor models.

Index	Function
Mean absolute error	$MAE = (1/n) \sum_{i=1}^n  \hat{y}_i - y_i $
Root mean square error	$RMSE = [(1/n) \sum_{i=1}^n (\hat{y}_i - y_i)^2]^{1/2}$
$R^2$	$R^2 = (\sum_{i=1}^n (\hat{y}_i - \bar{y})^2) / (\sum_{i=1}^n (y_i - \bar{y})^2)$

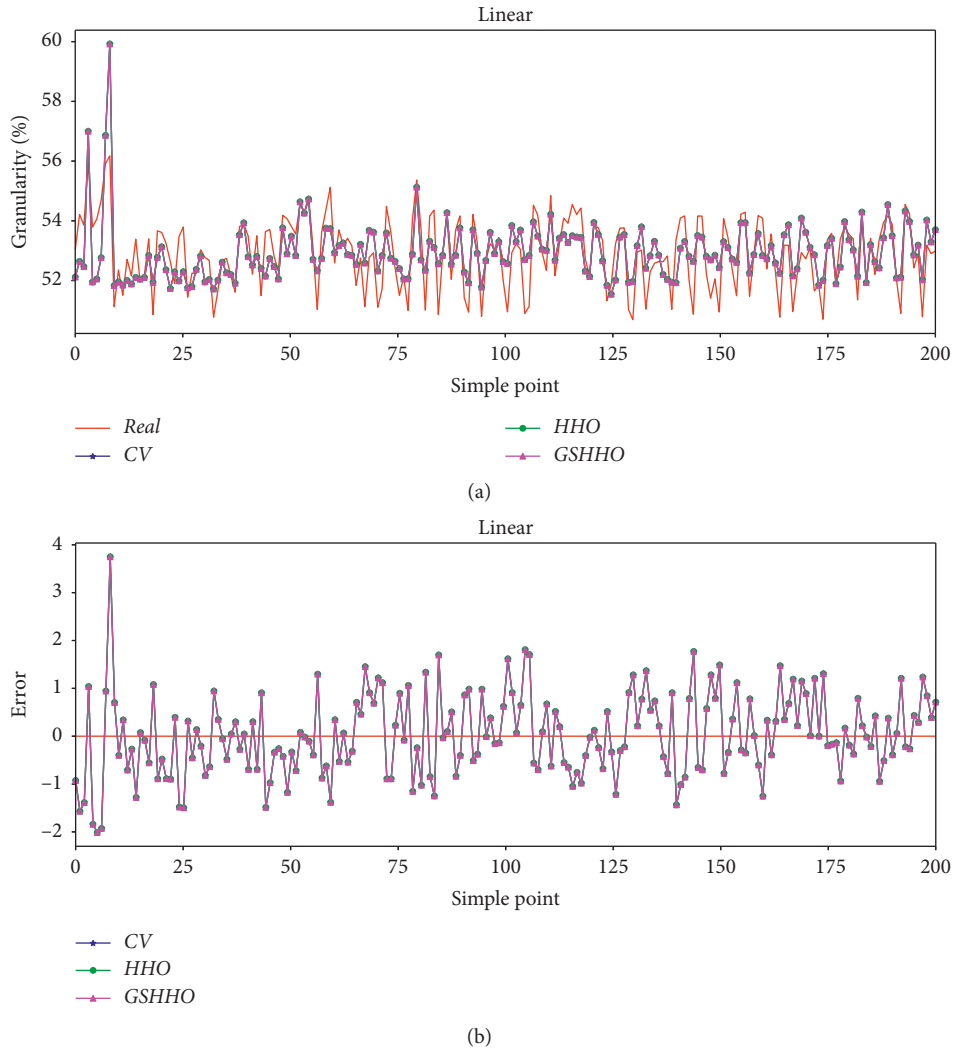


FIGURE 7: Prediction results under LSSVM model with *linear* function. (a) Prediction results under different LSSVM models. (b) Prediction errors under different LSSVM models.

to reflect the optimization capabilities of GSHHO algorithm and HHO algorithm, the convergence curves that GSHHO algorithm and HHO algorithm optimize different error functions are shown in Figure 11. The quantitative indicator results of ore granularity predicted by CV-LSSVM, GSHHO-LSSVM, and HHO-LSSVM are listed in Table 6 to further illustrate the prediction accuracy of different soft-sensor models.

It can be seen from the experimental results in these figures and tables that LSSVM with different kernel functions show different prediction capabilities. From Figures 8

and 9 and Table 6, it can be seen that compared with traditional crossvalidation, the optimization capabilities of GSHHO algorithm and HHO algorithm for LSSVM with *linear* functions and *poly* function do not show great advantages, the prediction results of these three methods are not ideal, and it is impossible to perform real-time monitoring of the particle size of the ore. From Figure 10 and Table 6, for LSSVM with *sigmoid* function, the optimization capabilities of GSHHO algorithm are significantly better than HHO and CV. Compared with traditional HHO algorithm, GSHHO algorithm has both strong exploration

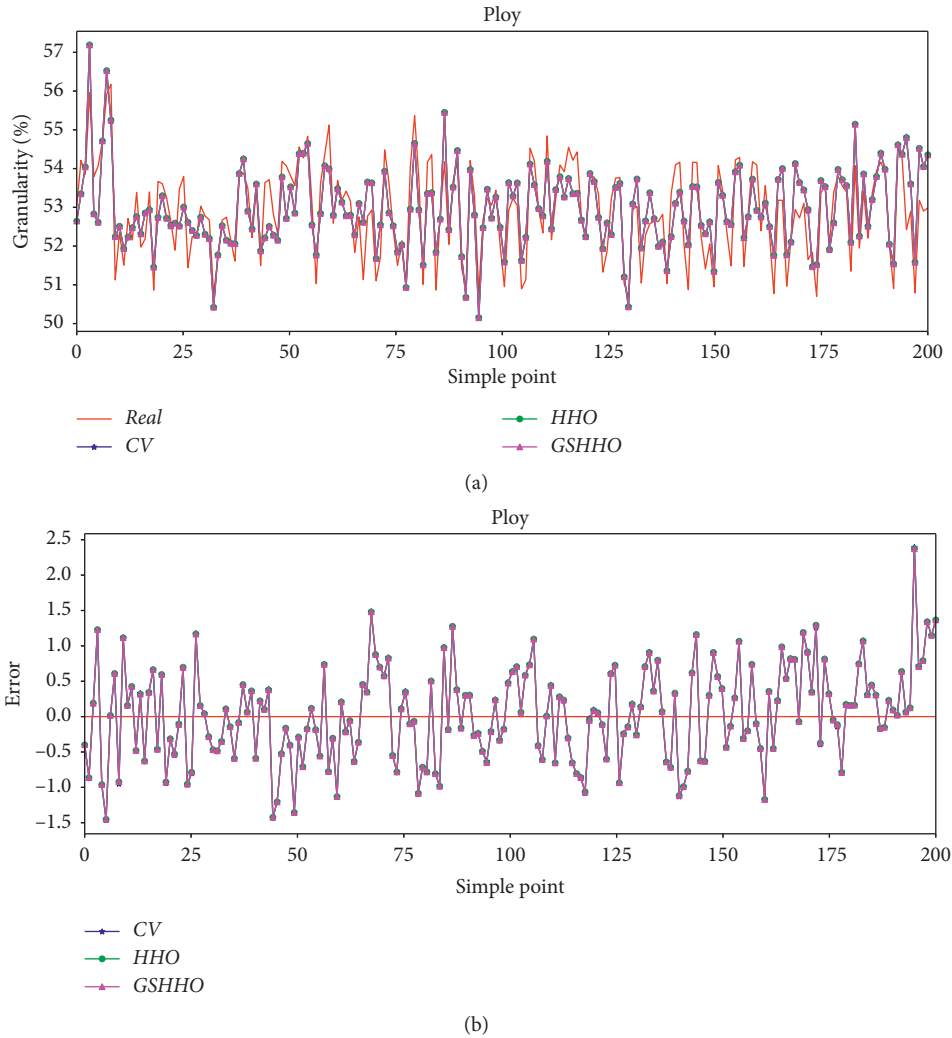


FIGURE 8: Prediction results under LSSVM model with *ploy* function. (a) Prediction results under different LSSVM models. (b) Prediction errors under different LSSVM models.

and exploitation. It can be seen from the convergence curve of Figure 11(d) that GSHHO can find more suitable minimum value for the error function in equation (28), which makes prediction accuracy of GSHHO-LSSVM higher. However, compared with the LSSVM with *sigmoid* function is still not optimal. Combining the quantitative indicators in Table 5 and the results in Figures 9 and 10, it can be seen that the prediction result of LSSVM with *rbf* optimized by CV is better than with *sigmoid* optimized by GSHHO algorithm. Looking further at the results in Figure 10 and Table 6, it can be seen that compared with HHO-LSSVM and CV-LSSVM, GSHHO-LSSVM with *rbf* function shows better performance. By the way, in the convergence curve of Figure 11, when GSHHO algorithm optimizes the error function, the convergence speed is faster and the convergence accuracy is higher. Table 7 shows the parameter values found by GSHHO algorithm and HHO algorithm when optimizing different soft-sensor models. Combining all the above conclusions, it can be concluded that GSHHO algorithm has a great advantage in optimizing the LSSVM model, and the

optimization result of the LSSVM model with *rbf* function is the best.

**5.3.2. Simulation Experiments of Regularized Extreme Learning Machine Model.** As explained in Section 2.2, RELM is a neural network model with strong generalization ability and high implementation efficiency. This section studies the prediction accuracy of RELM for ore granularity. GSHHO-LSSVM with *rbf* function has been proved to be a prediction model with good prediction accuracy, so this GSHHO-LSSVM will be compared with the prediction result of the RELM model. The traditional ELM experimental results will be used as a comparison of the RELM model. The number of hidden layer neurons in the ELM model and RELM model is 45. *sigma* function is used as the kernel function of ELM and RELM, and regularized parameter  $1/\lambda$  of RELM is 0.001. In the case of stable industrial status, the GSLBH-ELM proposed by Xie et al. [23] has been proven to be an effective soft-sensor model. When predicting the ore granularity, GSLBH-ELM

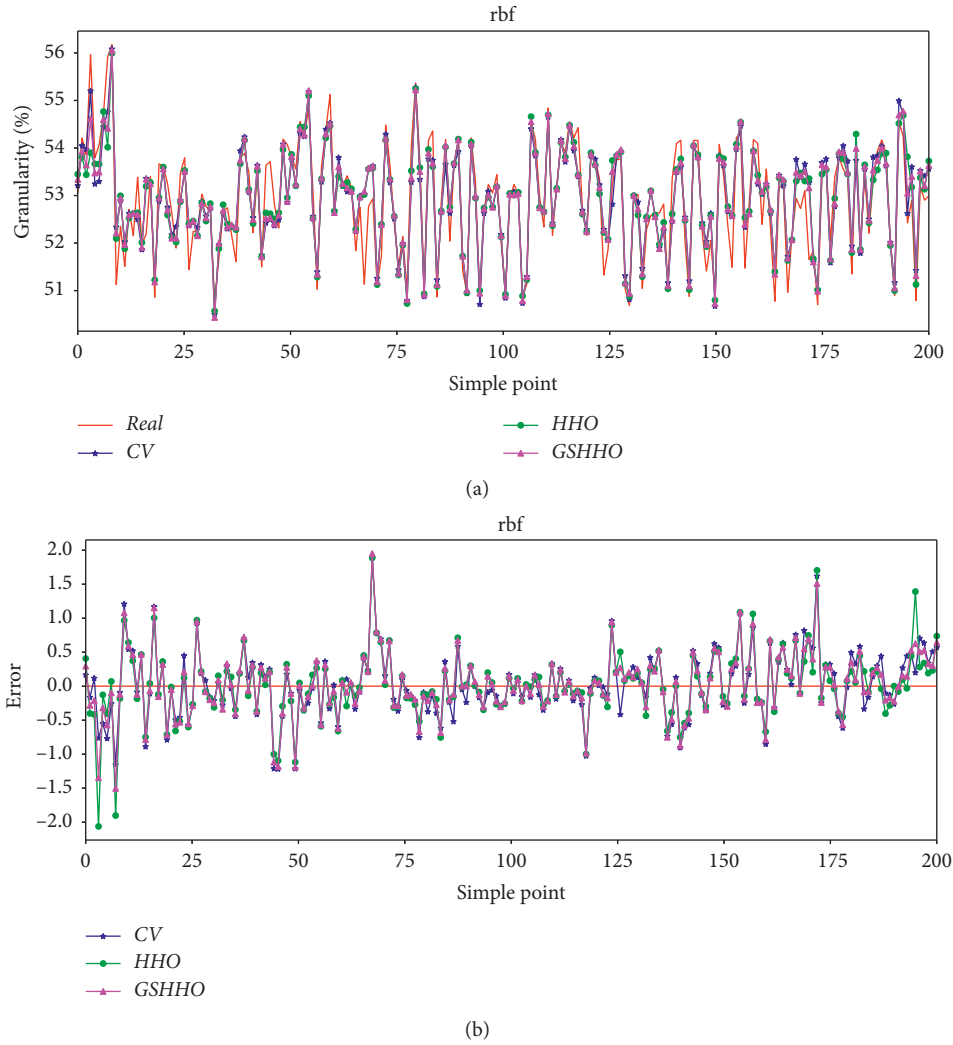


FIGURE 9: Prediction results under LSSVM model with *rbf* function. (a) Prediction results under different LSSVM models. (b) Prediction errors under different LSSVM models.

with arctan function has the highest prediction accuracy. This paper will also use GSLBH-ELM with arctan function to predict the ore granularity when this industrial status is constantly changing, and the results obtained will be compared with the experimental results of RELM, ELM, and LSSVM. The experimental results of these four models are shown in Figure 12 and Table 8.

It can be seen from the experimental results that the GSLBH-ELM model does not show good prediction ability. The main reason is that the number of hidden layer neurons selected by GSLBH-ELM is too small. The small number of neurons does not reflect the dynamic industrial characteristics of this experiment. So, GSLBH algorithm cannot find more reasonable weights and biases. If the number of neurons in the GSLBH-ELM is increased, the parameters that the algorithm needs to be optimized will increase geometrically and the time required to train the model will be too long, which is contrary to the efficiency of the soft-sensor model. Therefore, when the industrial process changes dynamically, GSLBH-ELM is not suitable for

predicting the ore granularity. Combining the results shown in Figure 12 and Table 8, the prediction performance of RELM is slightly better than ELM, especially in Figure 12(c), and it can be clearly seen that the prediction value of RELM is more accurate than ELM at some moments. However, the prediction accuracy of GSHHO-LSSVM is still higher than that of RELM and ELM.

In order to study the ability of the neural network model trained with the extended Kalman filter (KNN) to predict the granularity in the grinding and classifying process, this paper compares the prediction results of KNN with RELM and LSSVM. In this experiment, the structural parameters of the KNN model and the hyperparameters of the Kalman filter method are shown in Table 9, where  $I$  represents the identity matrix, the size of the matrix  $P$  is the square of the number of model parameters, and  $\text{std} = 0.01$ , the size of the matrix  $Q$  is the square of the number of output. The experimental results are shown in Figure 13, and the quantitative indicators of the prediction results are shown in Table 8. As can be seen from the results in Figure 13 and

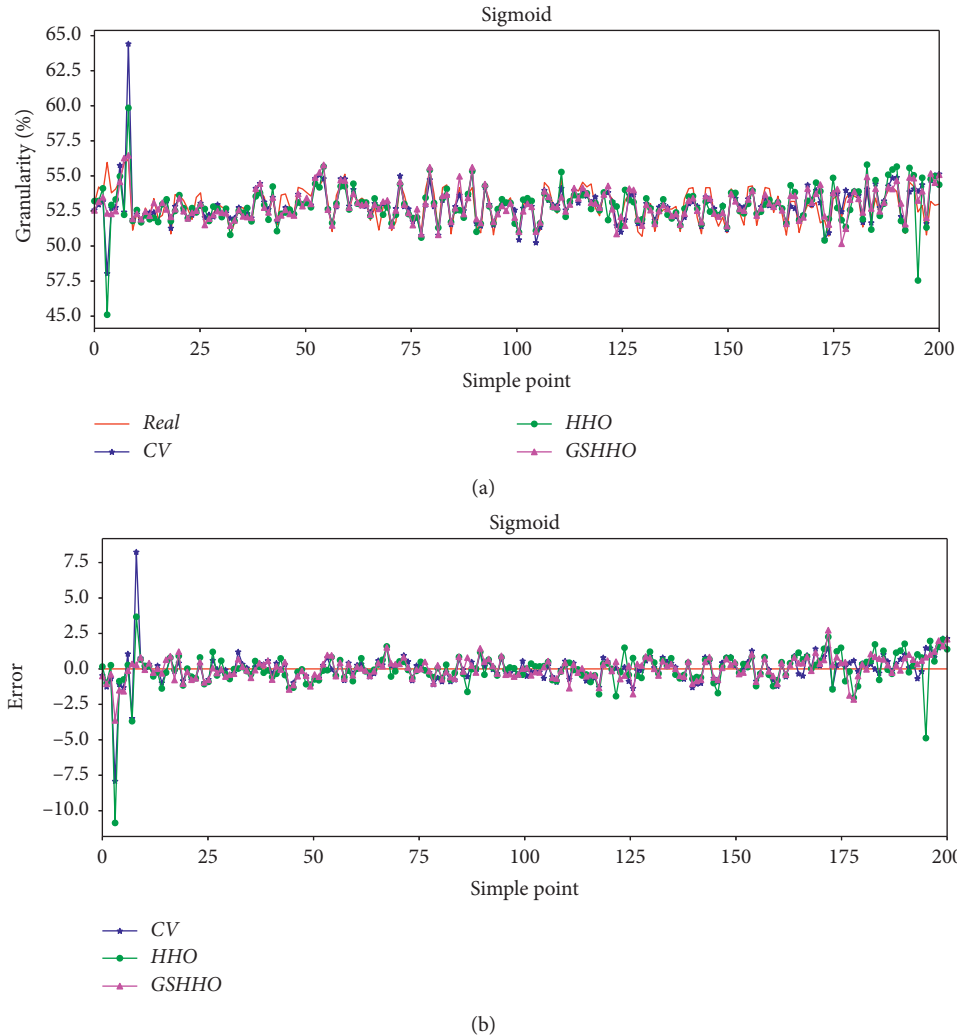


FIGURE 10: Prediction results under LSSVM model with *sigmoid* function. (a) Prediction results under different LSSVM models. (b) Prediction errors under different LSSVM models.

Table 8, compared with RELM and LSSVM, KNN does not show a large advantage, and its prediction accuracy is better than RELM, but still worse than LSSVM.

**5.3.3. Simulation Experiment of Hybrid Soft-Sensor Model Based on Moving Window Strategy.** Under the circumstances that the industrial status is changing at any time, this paper introduces moving window strategy into the hybrid soft-sensor model to improve the prediction accuracy of the model. The operation of this hybrid model has been described in details in Section 4.2. The window size in this experiment is 90, and the step size is 15. The LSSVM in the hybrid model uses the best performing model (the kernel function is *rbf* kernel, and the structural parameters are those found by GSHHO in Table 6) described in 5.3.2, the parameters of the RELM model are those selected in Section 5.3.3. At the same time, the moving window strategy is introduced into the LSSVM model (MW-LSSVM), the RELM model (MW-RELM), and the ELM model (MW-

ELM) as comparative experiments. The experimental results are all shown in Figures 14 and 15 and Tables 10 and 11.

From the experimental results in Figure 14 and Table 10, it can be seen that when building a local soft-sensing model, the MW-ELM model is very prone to overfitting due to the small number of training samples, there are some abnormal points in the test results of MW-ELM, and the prediction accuracy of the model is greatly reduced, even worse than the ELM prediction results in Section 5.3.3. MW-RELM overcomes the shortcomings of MW-ELM, and the introduction of regularization parameters  $1/\lambda$  makes the model avoid the interference of noise and greatly improves the generalization ability of the model. Because MW-RELM can adaptively adjust the structural parameters of the model according to changes in the working conditions, compared with the single model in Section 5.3.3, the prediction accuracy of MW-RELM has been greatly improved. From the results of Figure 15 and Table 11, it can be seen that the prediction results of MW-LSSVM are slightly better than MW-RELM. The hybrid model



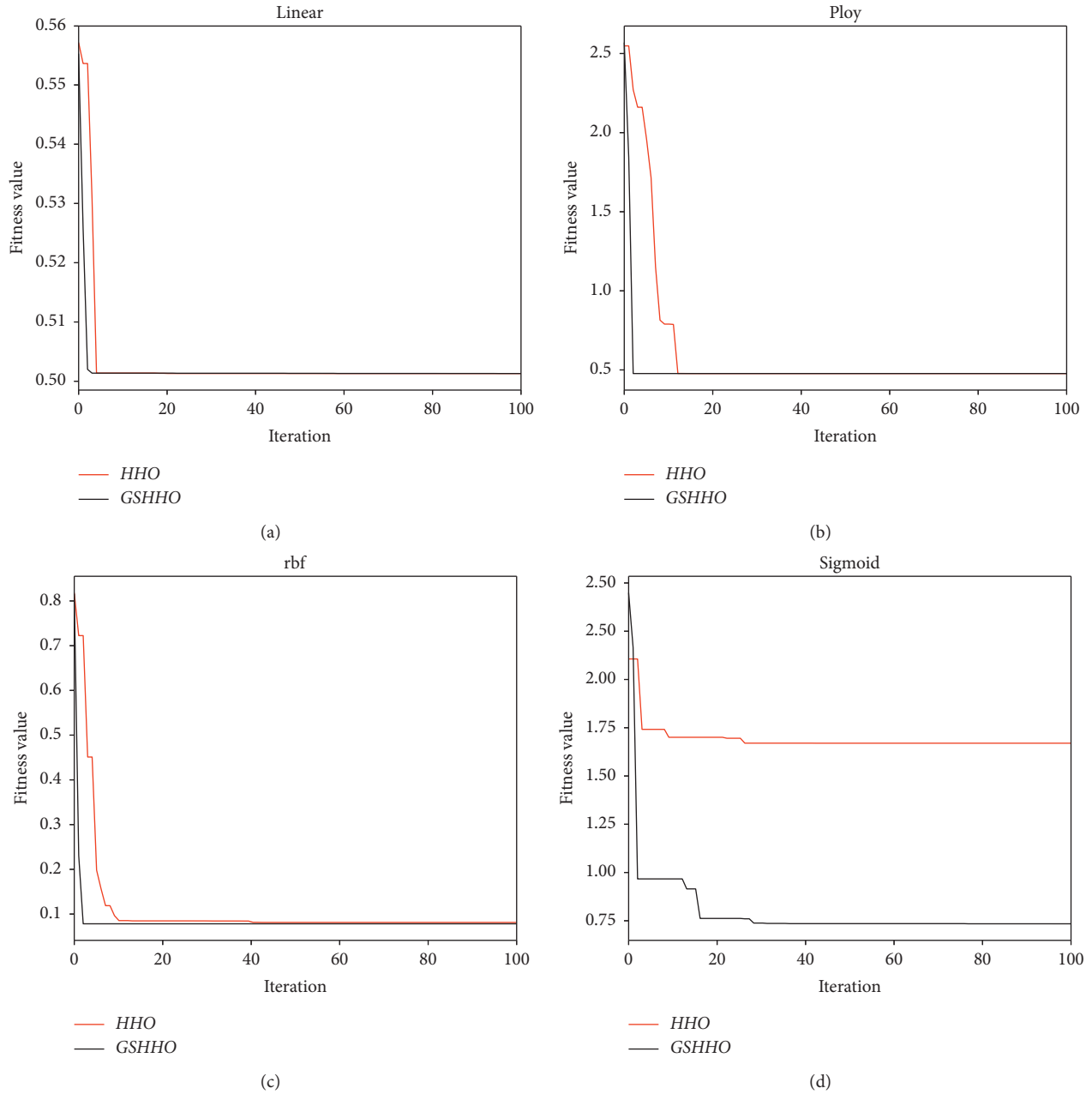


FIGURE 11: Convergence curves of error functions under four different LSSVM models.

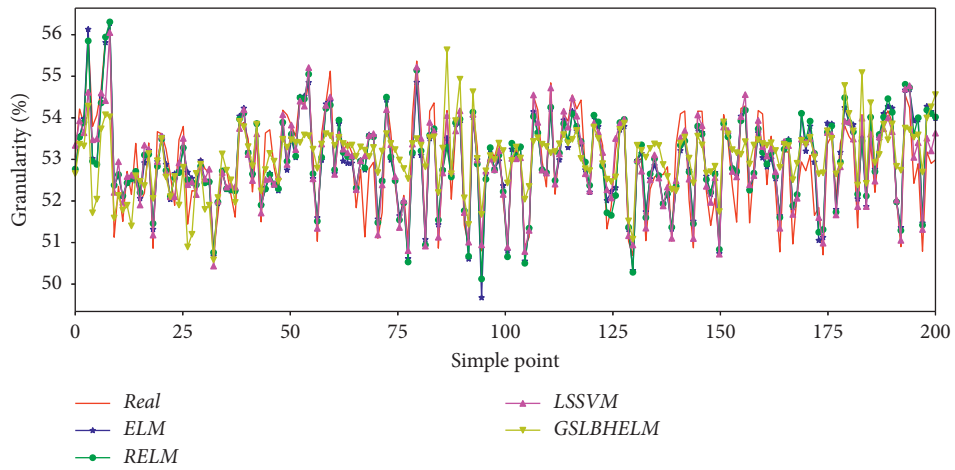
TABLE 6: Quantitative indicators under different soft-sensor models.

Kernel function	Algorithm	MAE	RMSE	$R^2$
Linear	CV	0.686	0.7327	0.202
	HHO	0.6861	0.7327	0.2024
	GSHHO	0.6861	0.7327	0.2024
	CV	0.5368	0.436	0.5763
Ploy	HHO	0.5363	0.4348	0.5786
	GSHHO	0.5363	0.4347	0.5787
	CV	0.3642	0.2296	0.7812
rbf	HHO	0.3412	0.2351	0.7731
	GSHHO	0.3453	0.2191	0.79
	CV	0.6469	1.1839	0.3368
Sigmoid	HHO	0.7233	1.4488	0.2213
	GSHHO	0.5841	0.5839	0.4942

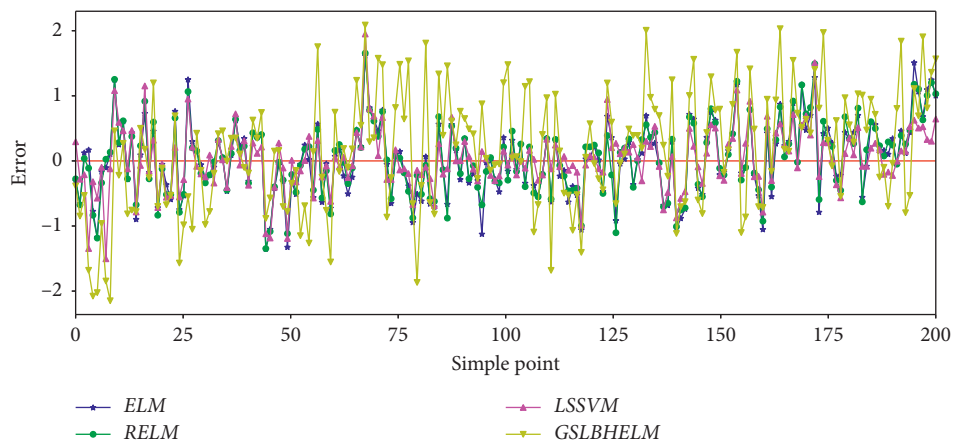
TABLE 7: Parameter values found by the algorithms.

Kernel function	Algorithm	Parameter value
Linear	HHO	$C = 124.36546$
	GSHHO	$C = 148.61254$
Ploy	HHO	$C = 85.31564, \theta = 3.2146$
	GSHHO	$C = 98.63345, \theta = 1.36543$
rbf	HHO	$C = 53.24514, \sigma^2 = 1.12283$
	GSHHO	$C = 61.34664, \sigma^2 = 0.71341$
Sigmoid	HHO	$C = 253.54376, k = 0.86543$
	GSHHO	$C = 257.77324, k = 0.36424$

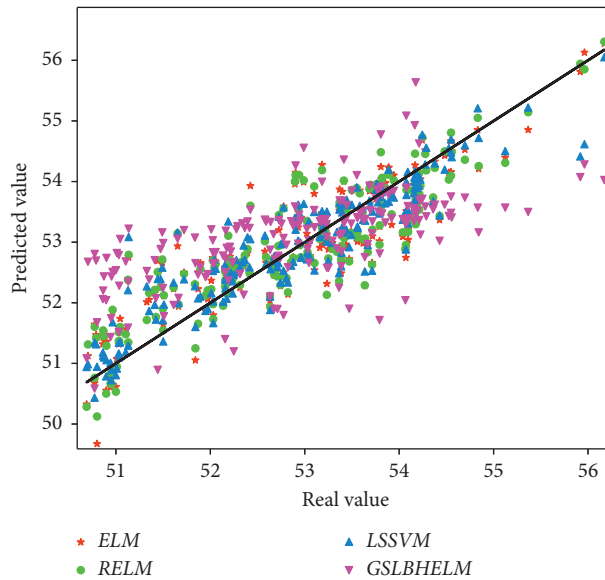
MW-LSSVM-RELM combines the advantages of LSSVM and RELM under different industrial status which can further improve the prediction accuracy of the model and



(a)



(b)



(c)

FIGURE 12: Experimental results under different ELM and LSSVM models. (a) Prediction results under different soft-sensor models. (b) Prediction errors under different soft-sensor models. (c) Predicted scatter plots under different soft-sensor models.

TABLE 8: Quantitative indicators under different soft-sensor models.

Model	MAE	RMSE	$R^2$
ELM	0.4525	0.3183	0.7101
RELM	0.4488	0.3067	0.7209
LSSVM	0.3453	0.2191	0.7901
GSLBH-ELM	0.7403	0.0267	-0.5167
KNN	0.3532	0.2307	0.7904

TABLE 9: The structural parameters of the KNN model and the hyperparameters of the Kalman filter method.

<i>(a) Neural network model structure parameters</i>				
Parameter	Number of model inputs	Number of hidden neurons	Number of model output	Number of iterations
Value	5	10	1	200
<i>(b) Kalman filter parameters</i>				
Parameter	Step of update weight	Initial weight covariance $P$	Data covariance $R$	Process covariance $Q$
Value	1	$P = 0.5 * I$	$R = (0.1 + \text{std}^2) * I$	$Q = 0$

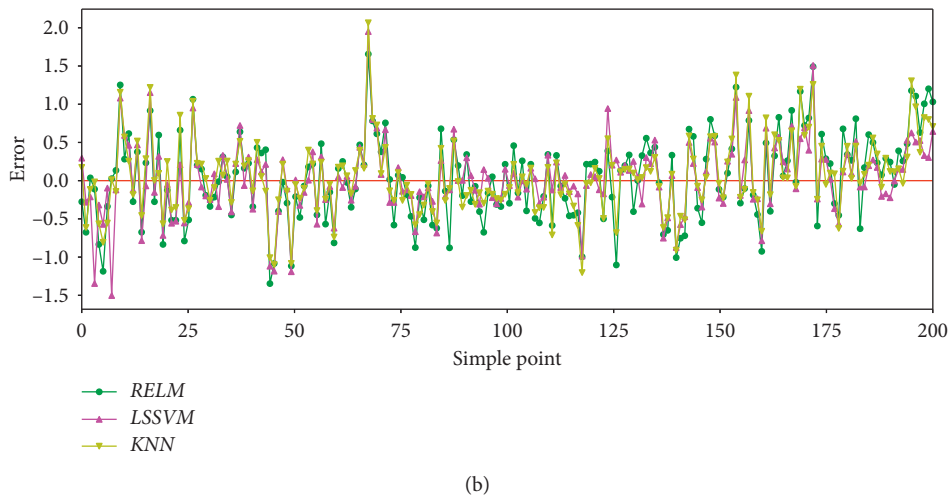
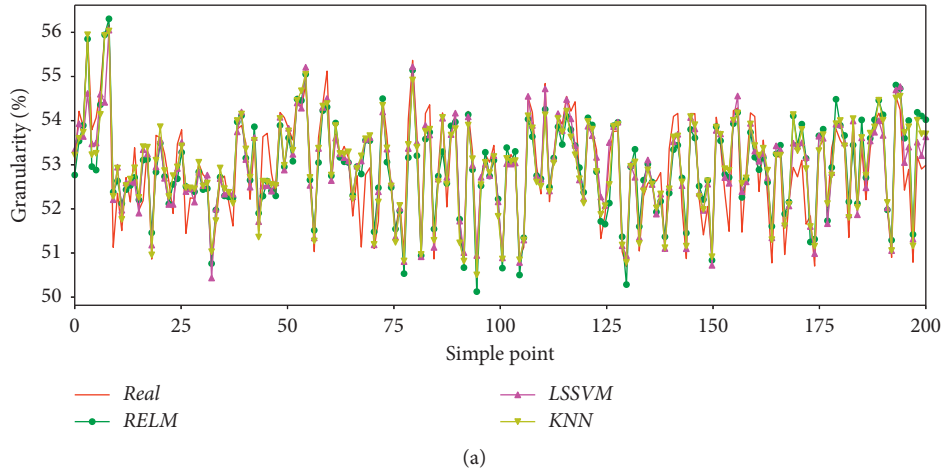


FIGURE 13: Continued.

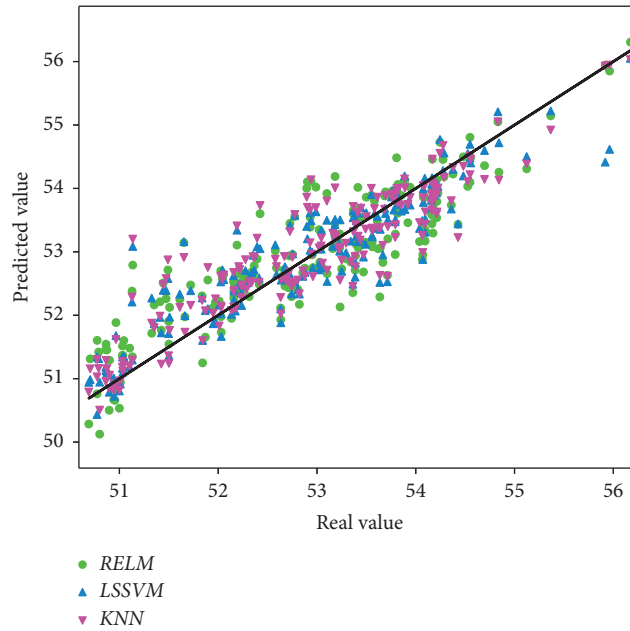


FIGURE 13: Experimental results under KNN and other models. (a) Prediction results under different soft-sensor models. (b) Prediction errors under different soft-sensor models. (c) Predicted scatter plots under different soft-sensor models.

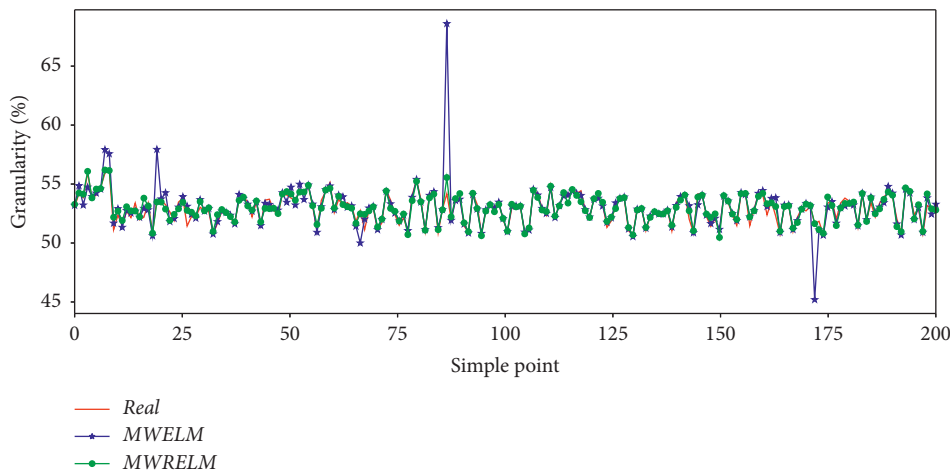


FIGURE 14: Prediction results under MW-ELM model and MW-RELM model. (a) Prediction results under different soft-sensor models. (b) Prediction errors under different soft-sensor models. (c) Predicted scatter plots under different soft-sensor models.

achieve the purpose of online prediction of the ore granularity in a dynamic industrial environment.

In order to find out which model is selected every time the window slides, this article shows the model switching process in Figure 16, where “1” indicates that the LSSVM model is used as the local soft-sensor model and “-1” indicates that the RELM model is used as the local soft-sensing model. It can be seen from the results in Figure 16 that the LSSVM model is used as the local soft-sensing model is the majority, and the role of the RELM model is to “revise” when LSSVM performs poorly. In order to explore the working efficiency of the MW-LSSVM-RELM model,

the training time required for the MW-LSSVM-RELM model to predict a new query sample is listed in Table 12. The results in Table 12 show that although the training time of MW-LSSVM-RELM is longer than that of MW-RELM and MW-LSSVM, its training time is still very short and the model is extremely efficient, much less than the acceptable training time.

Based on the above experimental results, it can be seen that the MW-LSSVM-RELM model has better prediction ability than other single soft-sensor models and models that introduce the MW strategy into a single soft-sensor model. The MW-LSSVM-RELM model has better prediction

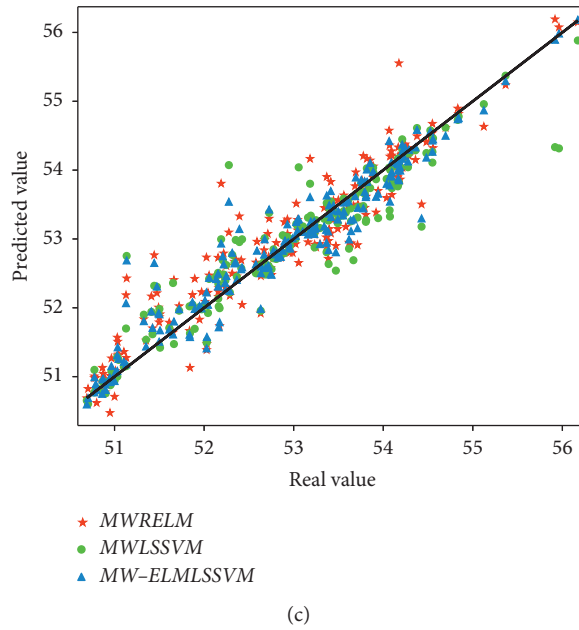
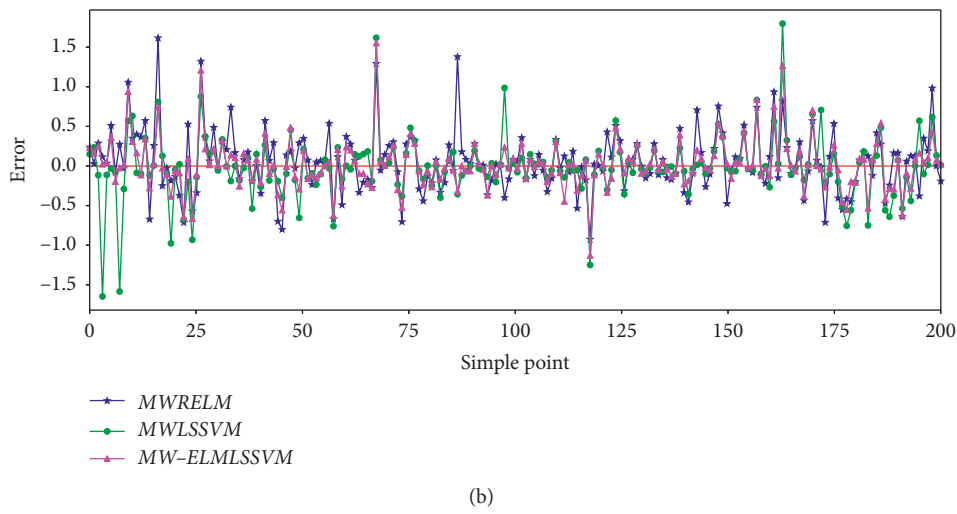
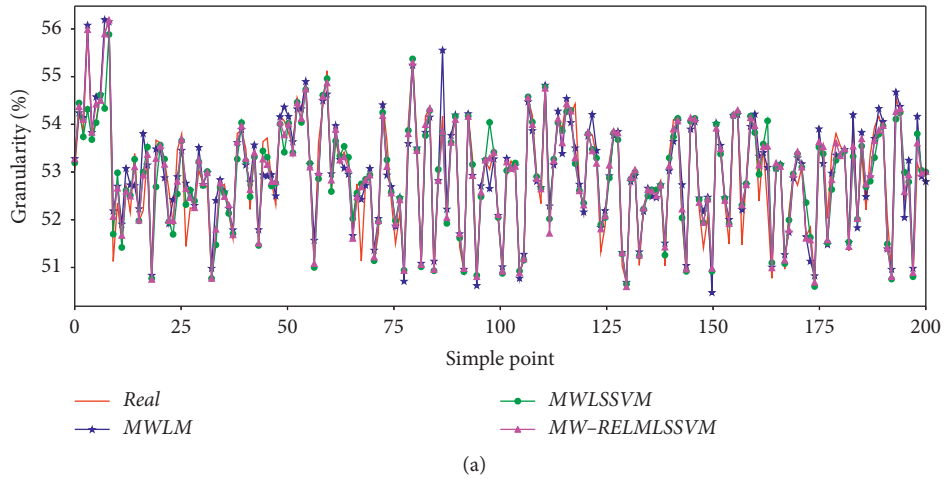


FIGURE 15: Experimental results under different hybrid models. (a) Prediction results under different soft-sensor models. (b) Prediction errors under different soft-sensor models. (c) Predicted scatter plots under different soft-sensor models.

TABLE 10: Quantitative indicators under different soft-sensor models.

Model	MAE	RMSE	$R^2$
MW-ELM	0.4104	1.5481	0.5021
MW-RELM	0.2859	0.1552	0.8765

TABLE 11: Quantitative indicators under different models.

Model	MAE	RMSE	$R^2$
MW-RELM	0.2859	0.1552	0.8765
MW-LSSVM	0.2439	0.1534	0.8651
MW-LSSVM-RELM	0.2094	0.1003	0.9175
MW-KNN	0.2386	0.1373	0.8886

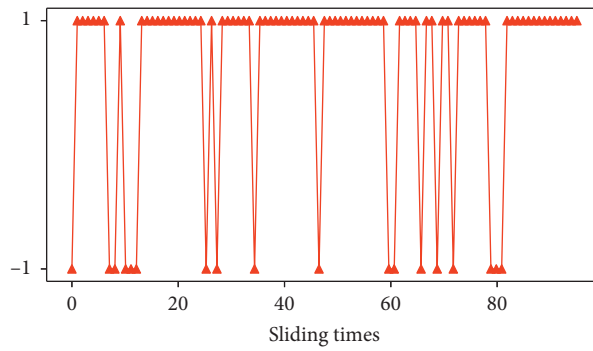
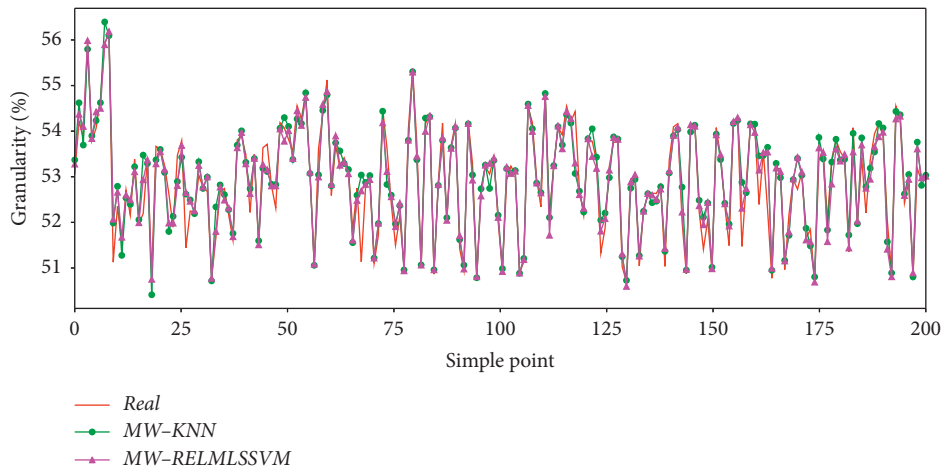


FIGURE 16: Switching process of different models.

TABLE 12: Training time under deterrent soft-sensor models.

Model	Training time(s)
MW-RELM	0.0043
MW-LSSVM	0.0086
MW-LSSVM-RELM	0.0111
MW-KNN	2.0076



(a)

FIGURE 17: Continued.

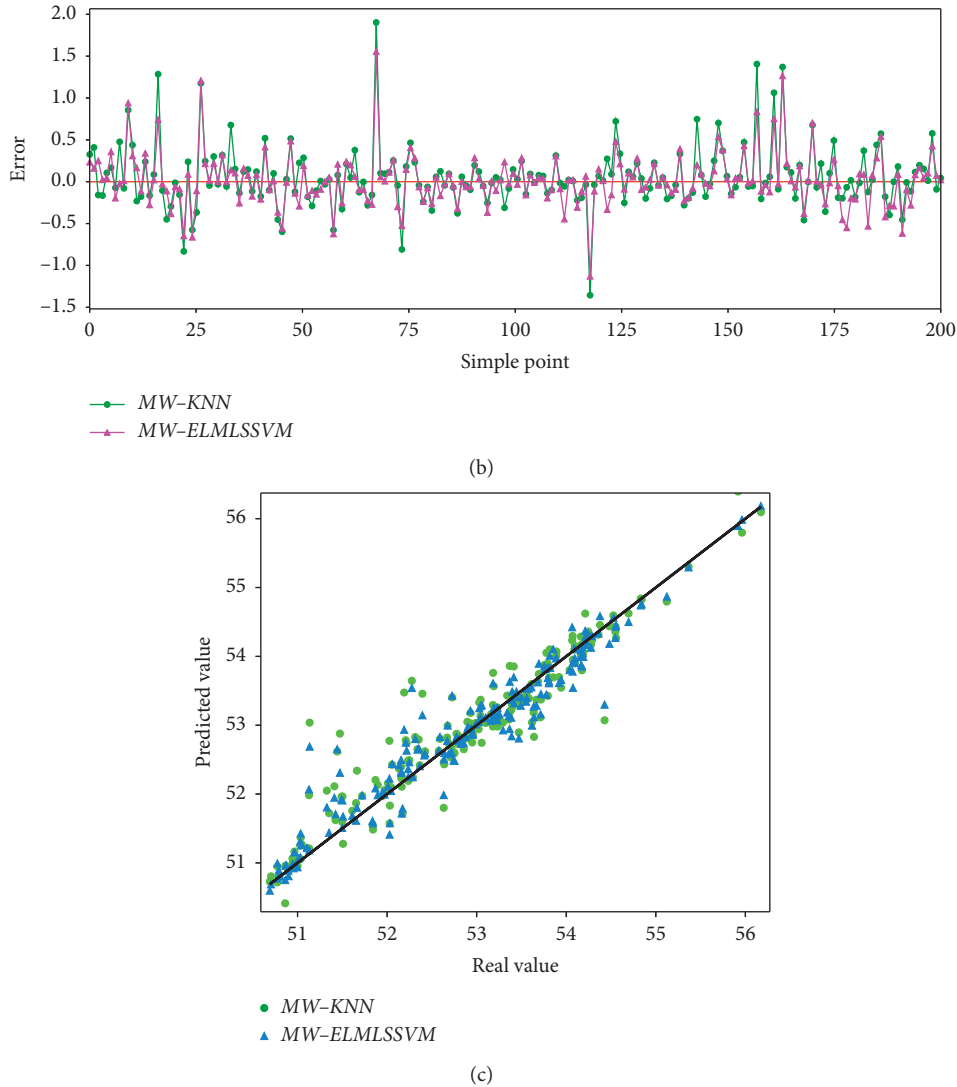


FIGURE 17: Experimental results under different hybrid models.

accuracy and higher work efficiency and can achieve online prediction of ore granularity.

It can be seen from the experimental results in Figure 13 and Table 9 that after the extended Kalman filtering method is used to adjust the parameters of the neural network, the neural network has good generalization ability, and its prediction accuracy is even better than RELM. In this section, in order to study the influence of moving window strategy on KNN prediction accuracy, this paper introduces moving window strategy into KNN. The window size and sliding steps are the same as MW-LSSVM-ELM. In order to improve the working efficiency of the model, the number of training iterations is set to 100, and other parameters remain unchanged. The experimental results of MW-KNN are shown in Figure 17 and Table 11. It can be seen from the results that the prediction accuracy of KNN has been greatly improved after the introduction of the sliding window strategy. Prediction accuracy of KNN is better than MW-RELM and MW-LSSVM, but still not as good as MW-LSSVM-ELM. Compared with MW-LSSVM-ELM, MW-KNN has a

major disadvantage, that is, it is less efficient. From the results in Table 12, it can be seen that although the number of iterations of KNN changes from 200 to 100, the time for MW-KNN to predict a test sample is still much longer than MW-LSSVM-ELM. From two aspects of work efficiency and prediction accuracy, the performance of the MW-LSSVM-ELM model is better than that of MW-KNN.

Combining all the above results, it can be concluded that the predictive ability of the MW-LSSVM-ELM model is superior to ELM, RELM, LSSVM, and KNN, even if these four models introduce moving window strategy, the prediction accuracy of MW-LSSVM-ELM is still better than them, and MW-LSSVM-ELM is more efficient than other models.

## 6. Conclusions

This paper first introduces the structure and implementation procedure of LSSVM model and RELM model in details, then introduces the HHO algorithm and proposes the

GSHHO algorithm, and also introduces the optimization process of LSHVM by GSHHO. In order to improve the adaptive ability of soft-sensor models, this paper adopts MW strategy into the LSSVM model and the RELM model to revise the structural parameters of the models. Based on MW technology, this paper proposes the MW-LSSVM-RELM model to further improve the prediction accuracy of the models. The following conclusions are drawn through the final simulation experiments results:

- (1) Compared with the traditional CV method and HHO algorithm, GSHHO algorithm can find more suitable structural parameters for LSSVM, which makes the prediction accuracy of LSSVM higher
- (2) The introduction of the MW strategy into the LSSVM model, the RELM model, and KNN model can greatly improve the ability of models to respond to changes of the industrial status and make these models maintain good generalization ability in different states
- (3) The MW-LSSVM-RELM model proposed in this paper combines the advantages of the LSSVM model and the RELM model in different industrial status, which can make this hybrid model have better prediction capabilities than a single soft-sensor model

In the database of the grinding and classifying process, there are many samples where data are missing for various reasons. Conventional soft-sensor models cannot take advantage of this data-missing sample, which contains a lot of key information that can characterize the state of the industry. Abandoning these samples means losing a lot of useful information and wasting data resources. Based on the existing research work, this team will focus on deep neural network models because deep neural networks can better describe the strong nonlinear and dynamic changes of industrial processes, and the team will study the semi-supervised model of the existing soft-sensor model so that these data-missing samples can be used for high-quality online prediction of key variables in the grinding and classification process.

### Data Availability

There are no data available for this paper.

### Conflicts of Interest

The authors declare that there are no conflicts of interests regarding the publication of this article.

### Authors' Contributions

Wei Xie participated in the algorithm simulation and draft writing. Jie-Sheng Wang participated in the concept, design, interpretation, and commented on the manuscript and the critical revision of this paper. Cheng Xing, Sha-sha Guo, Meng-wei Guo, and Ling-feng Zhu participated in the data collection and analysis of the paper.

### Acknowledgments

This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant no. 2017FWDF10) and Project by Liaoning Provincial Natural Science Foundation of China (Grant no. 20180550700).

### References

- [1] D. Sbárbaro and R. Del Villar, *Advanced Control and Supervision of Mineral Processing Plants*, Springer Science & Business Media, Berlin, Germany, 2010.
- [2] H. G. Wen and M. Y. Ni, "Multiple neural networks-based soft sensing model for particle size of grinding product," *Metal Mine*, vol. 2, no. 344, pp. 47–49, 2005.
- [3] W. Shao, Z. Ge, and Z. Song, "Semi-supervised mixture of latent factor analysis models with application to online key variable estimation," *Control Engineering Practice*, vol. 84, pp. 32–47, 2019.
- [4] W. Shao, Z. Ge, Z. Song, and K. Wang, "Nonlinear industrial soft sensor development based on semi-supervised probabilistic mixture of extreme learning machines," *Control Engineering Practice*, vol. 91, p. 104098, 2019.
- [5] L. Yao and Z. Ge, "Moving window adaptive soft sensor for state shifting process based on weighted supervised latent factor analysis," *Control Engineering Practice*, vol. 61, pp. 72–80, 2017.
- [6] Z. Ge and Z. Song, "A comparative study of just-in-time-learning based methods for online soft sensor modeling," *Chemometrics and Intelligent Laboratory Systems*, vol. 104, no. 2, pp. 306–317, 2010.
- [7] W. Shao, Z. Ge, and Z. Song, "Soft-sensor development for processes with multiple operating modes based on semi-supervised Gaussian mixture regression," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 2169–2181, 2018.
- [8] Z. Ge, "Supervised latent factor analysis for process data regression modeling and soft sensor application," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1004–1011, 2015.
- [9] X. Yuan, Z. Ge, Z. Song, Y. Wang, C. Yang, and H. Zhang, "Soft sensor modeling of nonlinear industrial processes based on weighted probabilistic projection regression," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 4, pp. 837–845, 2017.
- [10] A. Casali, G. Gonzalez, G. Vallebuona, C. Perez, and R. Vargas, "Grindability soft-sensors based on lithological composition and on-line measurements," *Minerals Engineering*, vol. 14, no. 7, pp. 689–700, 2001.
- [11] A. Casali, G. Vallebuona, M. Bustos, G. Gonzalez, and P. Gimenez, "A soft-sensor for solid concentration in hydrocyclone overflow," *Minerals Engineering*, vol. 11, no. 4, pp. 375–383, 1998.
- [12] A. Casali, G. Gonzalez, F. Torres, G. Vallebuona, L. Castelli, and P. Gimenez, "Particle size distribution soft-sensor for a grinding circuit," *Powder Technology*, vol. 99, no. 1, pp. 15–21, 1998.
- [13] P. Zhou, S.-W. Lu, and T. Chai, "Data-driven soft-sensor modeling for product quality estimation using case-based reasoning and fuzzy-similarity rough sets," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 992–1003, 2014.



- [14] P. Zhou et al., "Soft-sensor approach with case-based reasoning and its application in grinding process," *Control and Decision*, vol. 21, no. 6, p. 646, 2006.
- [15] P. ZHOU and C. H. A. I. Tian-you, "Comparative study on soft-sensor of grinding particle size," *Journal of Northeastern University*, vol. 12, p. 2, 2010.
- [16] G. Yan, S. Ji, and G. Xie, "Soft sensor for ball mill fill level based on uncertainty reasoning of cloud model," *Journal of Intelligent & Fuzzy Systems*, vol. 30, no. 3, pp. 1675–1689, 2016.
- [17] Q. WANG et al., "Real-time online measurement of the overflow particle size for grinding process based on PLS," in *Nonferrous Metals (Mineral Processing Section)*, p. S1, 2011.
- [18] D. Sbarbaro, P. Ascencio, P. Espinoza, F. Mujica, and G. Cortes, "Adaptive soft-sensors for on-line particle size estimation in wet grinding circuits," *Control Engineering Practice*, vol. 16, no. 2, pp. 171–178, 2008.
- [19] J. Tang, Y. Tian, and Li J. Zhao, "On-line KPLS algorithm with application to ensemble modeling parameters of mill load," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 39, no. 5, pp. 471–486, 2013.
- [20] P. Kadlec, R. Grbić, and B. Gabrys, "Review of adaptation mechanisms for data-driven soft sensors," *Computers & Chemical Engineering*, vol. 35, no. 1, pp. 1–24, 2011.
- [21] J. Tang et al., "Modeling load parameters of ball mill in grinding process based on selective ensemble multisensor information," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 726–740, 2012.
- [22] J. Tang, T. Chai, L. Zhao, W. Yu, and H. Yue, "Soft sensor for parameters of mill load based on multi-spectral segments PLS sub-models and on-line adaptive weighted fusion algorithm," *Neurocomputing*, vol. 78, no. 1, pp. 38–47, 2012.
- [23] W. Xie, J. S. Wang, C. Xing, S. S. Guo, M. W. Guo, and L. F. Zhu, "Extreme learning machine soft-sensor model with different activation functions on grinding process optimized by improved black hole algorithm," *IEEE Access*, vol. 8, pp. 25084–25110, 2020.
- [24] G. Aquino, J. D. J. Rubio, J. Pacheco et al., "Novel nonlinear hypothesis for the delta parallel robot modeling," *IEEE Access*, vol. 8, pp. 46324–46334, 2020.
- [25] I. Elias, J. d. J. Rubio, D. R. Cruz et al., "Hessian with mini-batches for electrical demand prediction," *Applied Sciences*, vol. 10, no. 6, p. 2036, 2020.
- [26] Y. Zhang and Z. Wang, "Soft sensor model based on RBF neural network for particle size of grinding circuit and implementation," *Computer Measurement and Control*, vol. 25, no. 1, pp. 38–39, 2017.
- [27] X. D. Zhang, W. Wang, and X. G. Wang, "Research of the particle size neural network soft sensor for concentration process," *Control Theory and Applications*, vol. 19, no. 1, pp. 85–88, 2002.
- [28] Y. Li and S. Cheng, "Application of grey relation analysis and RBF network on grinding concentration's soft sensing," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.
- [29] J. de Jesus Rubio, "SOFMLS: online self-organizing fuzzy modified least-squares network," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1296–1309, 2009.
- [30] J. A. Meda-Campana, "On the estimation and control of nonlinear systems with parametric uncertainties and noisy outputs," *IEEE Access*, vol. 6, pp. 31968–31973, 2018.
- [31] Y. Zhang and Y. Liang, "Research on soft-sensor based on support vector regression for particle size of grinding and classification process," in *Proceedings of the 2016 Chinese Control and Decision Conference (CCDC)*, May 2016.
- [32] J. Tang et al., "Soft sensor modeling of ball mill load via principal component analysis and support vector machines," *Advances in Neural Network Research and Applications*, pp. 803–810, Springer, Berlin, Heidelberg, 2010.
- [33] Y. Kang, M. C. Lu, and G. W. Yan, "Soft sensor for ball mill fill level based on DBN-ELM model," *Instrument Technique and Sensor*, vol. 4, pp. 73–75, 2015.
- [34] G. Yang and J. Wang, "Soft measurement of grinding size based on modified fruit fly algorithm to optimize BP neural network," *Instrument Technique and Sensor*, vol. 8, p. 36, 2018.
- [35] Y. Zhou et al., "Soft sensor of particle size of grinding process based on improved CSAPSO neural networks," *Journal of System Simulation*, vol. 28, no. 12, pp. 2946–2950, 2016.
- [36] Y. Zhou et al., "Application of BCOISOA-BP network in soft measurement of grinding particle size," *Chinese Journal of Engineering*, vol. 39, no. 10, pp. 1546–1551, 2017.
- [37] W. Xie, J. S. Wang, and Y. Tao, "Improved black hole algorithm based on golden sine operator and Levy flight operator," *IEEE Access*, vol. 7, pp. 161459–161486, 2019.
- [38] J.-S. Wang, Y. Yang, and S. Shi-Feng, "SFLA-WNN soft-sensor modeling and reconfiguration of grinding process based on model migration," *Journal of Shanghai Jiaotong University*, vol. 46, no. 12, pp. 1951–1955, 2012.
- [39] W. Dai, Q. Liu, and T. Chai, "Particle size estimate of grinding processes using random vector functional link networks with improved robustness," *Neurocomputing*, vol. 169, pp. 361–372, 2015.
- [40] J.-S. Wang and Na-Na Shen, "Hybrid multiple soft-sensor models of grinding granularity based on cuckoo searching algorithm and hysteresis switching strategy," *Scientific Programming*, vol. 2015, Article ID 146410, 11 pages, 2015.
- [41] B. Lin and X. Gu, "Soft sensor modeling based on DE-LSSVM," *Journal of Chemical Industry and Engineering (China)*, vol. 59, no. 7, pp. 1681–1685, 2008.
- [42] J. A. K. Suykens, J. Vandewalle, B. De Moor, and B. De Moor, "Optimal control by least squares support vector machines," *Neural Networks*, vol. 14, no. 1, pp. 23–35, 2001.
- [43] R. Liao, H. Zheng, S. Grzybowski, and L. Yang, "Particle swarm optimization-least squares support vector regression based forecasting model on dissolved gases in oil-filled power transformers," *Electric Power Systems Research*, vol. 81, no. 12, pp. 2074–2080, 2011.
- [44] Y. Liu and J. Chen, "Integrated soft sensor using just-in-time support vector regression and probabilistic analysis for quality prediction of multi-grade processes," *Journal of Process Control*, vol. 23, no. 6, pp. 793–804, 2013.
- [45] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [46] J. d. J. Rubio, E. Lughofer, A. Plamen, J. F. Novoa, and J. A. Meda-Campana, "A novel algorithm for the modeling of complex processes," *Kybernetika*, vol. 54, no. 1, pp. 79–95, 2018.
- [47] S. C. Stubberud, R. N. Lobbia, and M. Owen, "An adaptive extended Kalman filter using artificial neural networks," *IEEE in Proceedings of the 1995 34th IEEE Conference on Decision and Control*, vol. 2, pp. 1852–1856, New Orleans, LA, USA, December 1995.
- [48] Suykens, A. K. Johan, and J. Vandewalle, "Multiclass least squares support vector machines," in *Proceedings of the*

- IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, vol. 2, July 1999.
- [49] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
  - [50] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
  - [51] T. Guo, L. Zhang, and X. Tan, "Neuron pruning-based discriminative extreme learning machine for pattern classification," *Cognitive Computation*, vol. 9, no. 4, pp. 581–595, 2017.
  - [52] L. Zhang and D. Zhang, "Robust visual knowledge transfer via extreme learning machine-based domain adaptation," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4959–4973, 2016.
  - [53] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining*, IEEE, Nashville, TN, USA, March–April 2009.
  - [54] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer science & business media, Berlin, Germany, 2013.
  - [55] Y.-J. Lee, Yi-R. Yeh, and H.-K. Pao, *An Introduction to Support Vector Machines*, National Taiwan University of Science and Technology, Taipei, Taiwan, 2010.
  - [56] O. L. Mangasarian and E. W. Wild, "Proximal support vector machine classifiers," in *Proceedings of the KDD-2001: Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 2001.
  - [57] D. T. Bui, H Moayedi, B Kalantar et al., "Harris hawks optimization: a novel swarm intelligence technique for spatial assessment of landslide susceptibility," *Sensors*, vol. 19, no. 16, p. 3590, 2019.
  - [58] X. Bao, H. Jia, and C. Lang, "A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76529–76546, 2019.
  - [59] J. C. Bednarz, "Cooperative hunting Harris' hawks (*Parabuteo unicinctus*)," *Science*, vol. 239, pp. 1525–1527, 1988.
  - [60] X.-S. Yang, *Nature-inspired Metaheuristic Algorithms*, Luniver press, UK, 2010.
  - [61] Z. Yang, L. Yao, and Z. Ge, "Streaming parallel variational Bayesian supervised factor analysis for adaptive soft sensor modeling with big process data," *Journal of Process Control*, vol. 85, pp. 52–64, 2020.
  - [62] M. A. Maloof and R. S. Michalski, "Selecting examples for partial memory learning," *Machine Learning*, vol. 41, no. 1, pp. 27–52, 2000.
  - [63] W. Z. Sun and J. S. Wang, "Elman neural network soft-sensor model of conversion velocity in polymerization process optimized by chaos whale optimization algorithm," *Ieee Access*, vol. 5, pp. 13062–13076, 2017.