sensors

Article

A Novel Online Sequential Extreme Learning Machine for Gas Utilization Ratio Prediction in Blast Furnaces

Yanjiao Li [1,2], **Sen Zhang** [1,2,*], **Yixin Yin** [1,2], **Wendong Xiao** [1,2] **and Jie Zhang** [1,2]

[1] School of Automation & Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China; liyanjiao2009622@163.com (Y.L.); yyx@ies.ustb.edu.cn (Y.Y.); wdxiao@ustb.edu.cn (W.X.); zhangjie2009622@163.com (J.Z.)

[2] Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, Beijing 100083, China

* Correspondence: zhangsen@ustb.edu.cn; Tel.: +86-10-6233-2019

Received: 1 July 2017; Accepted: 7 August 2017; Published: 10 August 2017

Abstract: Gas utilization ratio (GUR) is an important indicator used to measure the operating status and energy consumption of blast furnaces (BFs). In this paper, we present a soft-sensor approach, i.e., a novel online sequential extreme learning machine (OS-ELM) named DU-OS-ELM, to establish a data-driven model for GUR prediction. In DU-OS-ELM, firstly, the old collected data are discarded gradually and the newly acquired data are given more attention through a novel dynamic forgetting factor (DFF), depending on the estimation errors to enhance the dynamic tracking ability. Furthermore, we develop an updated selection strategy (USS) to judge whether the model needs to be updated with the newly coming data, so that the proposed approach is more in line with the actual production situation. Then, the convergence analysis of the proposed DU-OS-ELM is presented to ensure the estimation of output weight converge to the true value with the new data arriving. Meanwhile, the proposed DU-OS-ELM is applied to build a soft-sensor model to predict GUR. Experimental results demonstrate that the proposed DU-OS-ELM obtains better generalization performance and higher prediction accuracy compared with a number of existing related approaches using the real production data from a BF and the created GUR prediction model can provide an effective guidance for further optimization operation.

Keywords: soft-sensor approach; data-driven model; machine learning; gas utilization ratio; blast furnace; online sequential extreme learning machine

1. Introduction

Single-hidden layer feedforward neural networks (SLFNs) have been widely used in many fields because they can approximate any function and form decision boundaries with arbitrary shape if the activation function is properly chosen [1–3]. However, traditional machine learning algorithms for training SLFNs such as the back-propagation (BP) algorithm [4,5] can easily get trapped in a local minimum and are time-consuming. To address the aforementioned problems, Huang et al. proposed a new kind of machine learning algorithm named extreme learning machine (ELM) for training SLFNs [6]. The salient features of ELM are that its hidden layer parameters do not require manual intervention and can be assigned randomly before training, and the output weight is determined analytically via the least squares estimation method, making it easy to implement with better generalization performance and faster learning speed [7–9]. Nowadays, ELM has been widely used in many fields, such as landmark recognition [10], industrial production [11], electronic nose [12], localization [13,14], etc.

ELM was initially proposed for batch learning, which considers all the available data at once. However, in a great number of real world applications, data may be received sequentially. In such cases, sequential learning should be preferable to batch learning. For this reason, the online learning type of ELM was proposed by Liang et al. and named online sequential ELM (OS-ELM) [15]. OS-ELM can deal with data of different sizes arriving one-by-one or chunk-by-chunk to update the training model in each time step. Recently, many improved learning algorithms based on OS-ELM were proposed by Lan et al. [16], Deng et al. [17], Huang et al. [18], Matias et al. [19]. However, these variants do not consider the data saturation problem, which is caused by some data generated in time-varying environments, thus there are also many improved OS-ELM-based algorithms for dealing with time-varying data. In [20], OS-ELM with kernels (OS-ELMK) was proposed by Wang et al., in which a limited memory strategy was designed to remove the data that have been trained when the number of trained data exceeds a predetermined threshold. Zhao et al. [21] proposed an OS-ELM with forgetting mechanism (FOS-ELM) to learn the data generated in time-varying environments, they implemented fixed sliding window to discard the old data when the new ones arrived. With the arrival of new observations, the sliding window and limited memory strategy ignore the contribution of the old data to the model, thus they cannot be adapted well to time-varying environments. Considering the problem from another point of view, the concepts of forgetting factor and weight strategy were introduced. Gu et al. [22] developed timeliness OS-ELM (TOS-ELM), which brought an adaptive weight strategy and iteration scheme to OS-ELM so that the newly arriving data could be given a reasonable attention to represent the current situation. However, TOS-ELM is time-consuming due to its iteration scheme. Lim et al. [23] presented an adaptive forgetting factor for OS-ELM and illustrated a complex mechanism to calculate the forgetting factor based on a gradient descent method, but it has a relatively high additional computational complexity and is also time-consuming. In addition, all these listed methods update the training model in each time step during the learning process, which would increase the computational complexity. Actually, when the system is relatively stable or the current training model can meet the prediction requirement, it may not need to update in each time step. In view of this fact, an appropriate model updated strategy should be considered to reduce the computational complexity and time consumption.

In order to tackle these issues, we propose a novel OS-ELM referred to as DU-OS-ELM. In DU-OS-ELM, a novel dynamic forgetting factor (DFF) and updated selection strategy (USS) are proposed to tackle the time-varying sequential data. The motivation for the DFF is that larger estimation errors should employ smaller forgetting factor to adapt to the time-varying environments and smaller estimation errors should employ larger forgetting factor to increase memory length at stable situations. In this way, the old collected data are forgotten gradually and the newly acquired data are given more considerable attention. Taking advantage of the novel DFF can effectively accelerate the convergence speed of DU-OS-ELM with low computational complexity. In addition, we also develop an USS to judge whether the model should be updated with the newly coming data in each time step. Thus, the proposed approach is more in line with the actual production situation. In addition, the convergence analysis of the proposed DU-OS-ELM is presented to ensure the estimation of the output weight converge to the true value with the new data arriving.

The blast furnace (BF) ironmaking process is the primary link in iron and steel production and one of its main energy-consuming processes [24–26]. Due to its high energy consumption, energy-saving and consumption-reduction are currently becoming important issues in BF ironmaking [27]. In actual production, the gas utilization ratio (GUR) is an important indicator for determining the energy consumption and operating status [28]. The improvement of GUR is an important embodiment of technical progress, therefore, the reasonable prediction and control of GUR is significant for efficient and low consumption BF iron production. However, due to the high complexity of the furnace interior, GUR is difficult to understand and evaluate through only the rich experience of specialized operators, therefore, how to establish an accurate GUR prediction model has become a hot research topic. Recently, several researchers are engaged in addressing this issue, mainly including mechanism-based models and data-driven models. For mechanism-based models, taking into account the influence of BF on GUR, a comprehensive improvement strategy based on production mechanism was proposed to enhance the GUR [29]. Due to the fast development of measurement instruments and sensor techniques, one can obtain a large amount of data from the BF ironmaking process. Therefore, data-driven models using machine learning approaches are an efficient way for predicting GUR. Based on expert theory and practical experience, a linear regression model between operating parameters and GUR was established [30]. Xiao et al. proposed a GUR forecasting model based on an improved BP algorithm [31]. An improved support vector regression (SVR) model based on the correlation analysis of some influence factors and GUR [32], and a chaos radial basis function (RBF) model [33] for GUR prediction were proposed, respectively. Xiao et al. presented a chaotic prediction method to analyze the characteristics of GUR [34]. Besides, many methods have been employed to establish soft-sensor models to predict the hot metal silicon content in the BF. A pruning algorithm is implemented to find both relevant inputs with their corresponding time delay and appropriate network connectivity based on a feedforward neural network to improve the prediction accuracy of hot metal silicon content in [35]. Support vector machine (SVM) and least square support vector machine (LSSVM) are not only applied to the value prediction of hot metal silicon content, but also for the prediction of change trends [36–38]. Although the aforementioned studies have obtained relatively satisfactory achievement, most of these models are constructed based on mechanistic research and fixed learning samples, which ignore the fact that BF is a time-varying system, and they are not adequate for online use. To tackle this issue, the proposed DU-OS-ELM is employed to establish an online GUR prediction model.

In this paper, a modified OS-ELM algorithm named DU-OS-ELM is firstly proposed based on DFF and USS. Compared with existing approaches, the proposed DU-OS-ELM is more suitable for dealing with data generated in time-varying environments and can meet the real-time requirements of real world applications. Then a soft-sensor model is established based on the proposed DU-OS-ELM to implement a fast and accurate GUR prediction. The experimental results indicate that DU-OS-ELM can achieve better performance than some similar algorithms using real industrial data. This research can help BF operators understand the energy consumption well, according to the changes of GUR, which provides effective guidance for further optimization operation.

The remaining parts of this paper are arranged as follows: Section 2 gives some preliminaries, including a brief introduction of ELM and OS-ELM. The proposed DU-OS-ELM is detailed in Section 3, including the DFF mechanism, USS and the convergence analysis of the proposed approach. The soft-sensor model for GUR is presented in Section 4. Section 5 presents the performance evaluation of DU-OS-ELM using the real data collected from a BF. Discussions and conclusions are given in Sections 6 and 7, respectively.

2. Preliminaries

In this section, ELM and OS-ELM are introduced to facilitate the understanding of the proposed DU-OS-ELM.

2.1. Extreme Learning Machine

As shown in Figure 1, ELM has a three-layer architecture. Considering N training samples $(x_i, t_i) \in R^n \times R^m$, here $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$ represents a n-dimensional attributes of the ith sample and $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T$ is a m-dimensional target vector [12]. The output of the network with L hidden nodes can be calculated by:

$$\sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, x_i) = t_j, j = 1, \ldots, N \tag{1}$$

where $\mathbf{a}_i = [a_{i1}, a_{i2}, \cdots, a_{in}]$ and b_i denote the learning parameters of the ith hidden node. $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]$ is weight vector connecting the ith hidden node and the output nodes, $G(\bullet)$ is the ith hidden node output function.

Equation (1) can be further compactly written in matrix form as:

$$\mathbf{H}\beta = \mathbf{T} \tag{2}$$

where **H** is the hidden layer output matrix:

$$\mathbf{H}(\mathbf{a}_1, \ldots, \mathbf{a}_L, b_1, \ldots b_L, x_1, \ldots, x_N) = \begin{bmatrix} G(\mathbf{a}_1, b_1, x_1) & \cdots & G(\mathbf{a}_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, x_N) & \cdots & G(\mathbf{a}_L, b_L, x_N) \end{bmatrix}_{N \times L} \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \text{ and } \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \tag{4}$$

The following optimization problem is formulated to identify β:

$$\begin{gathered} \min : J_{ELM} = \tfrac{1}{2}\|\beta\|^2 + \tfrac{\gamma}{2}\sum_{i=1}^{N}\|e_i\|^2 \\ s.t. : h(x_i)\beta = t_i^T - e_i^T \end{gathered} \tag{5}$$

where γ is the regularization parameter, e_i is the training error.

We can convert the above optimization problem into a dual optimization problem by employing the Karush-Kuhn-Tucker (KKT) theorem and further the output weight β can be obtained as:

$$\beta = (\frac{I}{\gamma} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{T} \tag{6}$$

$$\text{or } \beta = \mathbf{H}^T(\frac{I}{\gamma} + \mathbf{H}\mathbf{H}^T)\mathbf{T} \tag{7}$$

For computational convenience, when $N > L$, the solution of Equation (6) is preferred, and when $N < L$, we prefer to use the solution of Equation (7) to estimate β.

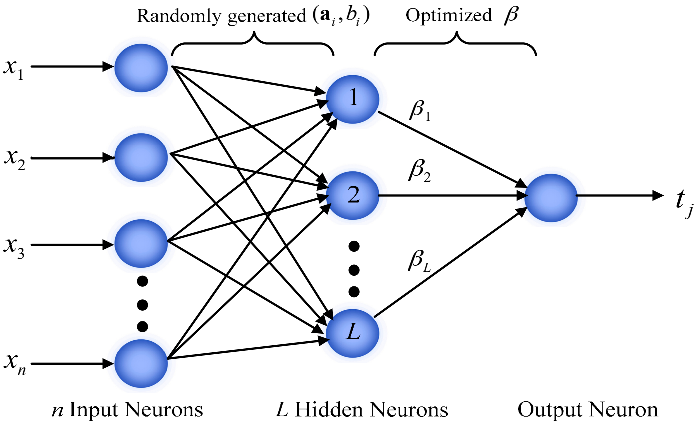

Figure 1. Basic architecture of ELM.

2.2. *Online Sequential Extreme Learning Machine*

The online version of ELM, which does not need to retrain the model with all the observations, but instead updates the model only with the newly arriving data was proposed by Liang et al. and named OS-ELM [15]. OS-ELM is similar to recursive least squares (RLS) based on the matrix inversion lemma to estimate the output weight vector [39,40]. OS-ELM consists of two phases: the initialization phase (a small-scale ELM) and the sequential learning phase.

For the initial data $\aleph_0 = \{x_j, t_j | j = 1, \ldots, N_0\}$, $N_0 \geq L$ in the initialization phase. According to the ELM theory, the output weight is initialized by:

$$\beta^{(0)} = \mathbf{P}_0 \mathbf{H}_0^{\mathbf{T}} \mathbf{T}_0 \tag{8}$$

where $\mathbf{T}_0 = [t_1, t_2, \cdots, t_{N_0}]$ and $\mathbf{P}_0 = (\frac{I}{\gamma} + \mathbf{H}_0^{\mathbf{T}} \mathbf{H}_0)^{-1}$.

In the sequential learning phase, the output weight is computed using an iteration technique because more and more observations arrive continuously. When the (k + 1)th chunk of the new observation: $\aleph_{k+1} = \{x_j, t_j | j = (\sum_{j=0}^{k} N_j) + 1, \ldots, \sum_{j=0}^{k+1} N_j\}$ is presented, where N_{k+1} denotes the number of newly arriving data in the (k + 1)th chunk, the new matrices $\mathbf{H}_{k+1}$ and $\beta^{(k+1)}$ in the (k + 1)th are calculated as follows:

$$\mathbf{K}_{k+1} = \frac{\mathbf{P}_k \mathbf{H}_{k+1}^T}{I + \mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T} \tag{9}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{K}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_k \tag{10}$$

$$e_{k+1} = \mathbf{T}_{k+1} - \mathbf{H}_{k+1} \beta^{(k)} \tag{11}$$

$$\beta^{(k+1)} = \beta^{(k)} + \mathbf{P}_{k+1} \mathbf{H}_{k+1}^T e_{k+1} \tag{12}$$

For further details of this updating method readers may refer to [3,15]. Thus, the output estimation function for OS-ELM is given by:

$$f_{OS-ELM}(x) = \sum_{i=1}^{L} \beta_i^{(k+1)} G(\mathbf{a}_i, b_i, x_i) \tag{13}$$

3. Details of the Proposed DU-OS-ELM

In real world applications, the production process is continuous and unstable with time, that is to say, the acquired data are changing. In order to better adapt to this situation, the contribution of the newly arriving data should be greater than that of previous ones. In addition, with learning ongoing and new incremental data for OS-ELM arriving sequentially, the storage space for the matrix will increase infinitely, which will lead to a loss of weight correction ability so that a data saturation problem finally appears. Therefore, we put forward the novel DFF and USS for OS-ELM to enhance the performance of sequential learning in time-varying environments.

3.1. Dynamic Forgetting Factor

The original optimization problem of OS-ELM can be represented by Equation (5). The purpose of adding the forgetting factor is to minimize the norm of the output weight and the weighted sum of squared error, so Equation (5) becomes:

$$\begin{aligned} &\min : L = \tfrac{1}{2} \|\beta\|^2 + \tfrac{\gamma}{2} \sum_{i=1}^{t} \lambda^{t-i} e_i^2 \\ &s.t. : h(\mathrm{x}_i)\beta = t_i^T - e_i^T \end{aligned} \tag{14}$$

Thus, we can obtain the similar expression with RLS:

$$\mathbf{K}_{k+1} = \frac{\mathbf{P}_k \mathbf{H}_{k+1}^T}{\lambda + \mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T}, \tag{15}$$

$$\mathbf{P}_{k+1} = \frac{1}{\lambda} (\mathbf{P}_k - \mathbf{K}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_k), \tag{16}$$

$$e_{k+1} = \mathbf{T}_{k+1} - \mathbf{H}_{k+1}\beta^{(k)}, \tag{17}$$

$$\beta^{(k+1)} = \beta^{(k)} + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^{T}e_{k+1}, \tag{18}$$

where $\lambda \in (0, 1]$ is the forgetting factor.

In the practical engineering problems, the dynamic characteristics of the system are not always changing with the same rules. For these kinds of situations, a constant forgetting factor will not ensure the satisfactory performance of the model. Therefore, we should automatically adjust the forgetting factor with the dynamic changes of the characteristic to obtain more accurate results. When the system parameters mutate, we select a smaller forgetting factor, which makes the model track the changing trend of the characteristics quickly. On the contrary, by choosing a larger forgetting factor to increase the memory length in the stable status and improve the accuracy of the model. Therefore, we present a novel DFF to improve the dynamic tracking ability of the original OS-ELM when it needs to update.

The essence of the novel DFF is to adjust λ to minimize the mean square error (MSE) and increase the convergence speed. Hence, we develop the DFF based on the predicted errors, which are updated recursively as:

$$\lambda_k = \lambda_{\min} + (1 - \lambda_{\min}) \times e^{-\mu\|e(k)\|}, \tag{19}$$

where λ_k is the forgetting factor for the kth chunk of new observations, μ is the step size to control the rate of λ to trend to 1, which will be discussed later. The additional computational complexity with respect to the original OS-ELM is only a few arithmetic calculations.

We set the boundary constraint of the DFF to avoid it too large or too small. The constraint conditions are as follows [41]:

$$\lambda_k = \begin{cases} \lambda_{\min}, & if\ \lambda_k < \lambda_{\min} \\ \lambda_{\max}, & if\ \lambda_k > \lambda_{\max} \end{cases} \tag{20}$$

Trivially, the forgetting factor in the exponential window can be used to control the effective memory of the window. λ can be written as $\lambda = e^{-1/\tau}$, where τ is the effective time constant that is related to data memory length. Thus, $\lambda_{\max}$ is set as 0.999 when $\tau = 1000$, and $\lambda_{\min}$ is set as 0.6 when $\tau = 2$ [42]. Figure 2 presents the changes of λ in the case of different μ. We can find that λ has a faster rate approaching 1 with the smaller μ. Under the same error, the larger μ usually has a corresponding smaller λ, thus, the memory length becomes shorter. In addition, with the decrease of μ, the change of λ is more stable. Therefore, we should determine μ according to the different data characteristics.

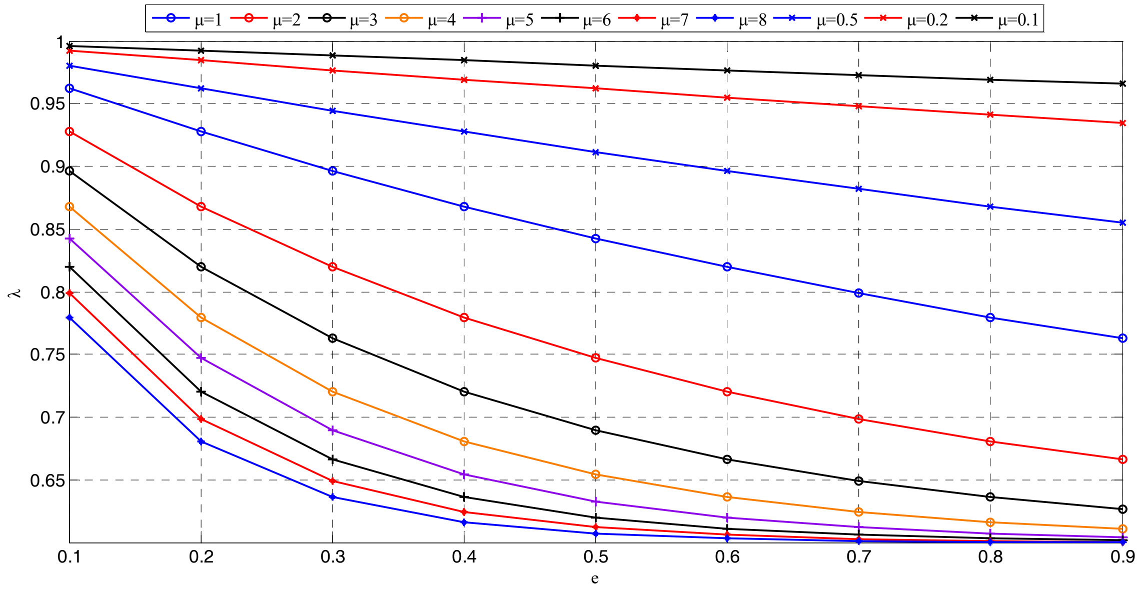

Figure 2. Changes of λ with different μ.

The behavior of the DFF is as follows: the distribution of the data in the time-varying environment influences the calculation of λ_k. When the system parameters are fluctuating, $e(k)$ is increased, and λ_k is decreased according to Equation (19). On the other hand, the impact of the old observations is reduced in the calculation of the output weight $\beta^{(k+1)}$ in Equation (18).

3.2. Updated Selection Strategy

In brief, when the new data arrive one-by-one or chunk-by-chunk for the OS-ELM training model, the output weight needs to be updated. However, when the production process is stable or the current model can meet the prediction requirements, there is no need to update the existing model in every time step [43]. Thus, a criterion is added to judge whether the training model needs to be updated:

$$\beta^{(k+1)} = \begin{cases} \beta^k + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^T e_{k+1}, e_{k+1} > e_k \\ \beta^k, e_{k+1} \le e_k \end{cases}. \tag{21}$$

If the predicted error of the new data by the current model is larger than the previous one, the existing model needs to be updated. Otherwise, the existing model does not need to be updated. Thus, this can not only reduce the time consumption, but also avoid the accumulation of errors.

3.3. Convergence Analysis

We present the convergence analysis of the proposed approach in this subsection.

Theorem 1. *Consider the proposed DU-OS-ELM and its output weight updated process, the estimation of output weight $\hat{\beta}^{(k)}$ will converge to the true value $\beta^{(k)}$. In addition, the output weight updated error has following upper bounds:*

$$E\left[\|\widetilde{\beta}^{(k)}\|^2\right] \le \lambda_{\max}^2 \left(C/\prod_{i=1}^{k-1}\lambda_i\right)^2 \sigma_0\delta_0, \tag{22}$$

where $\widetilde{\beta}^{(k)} = \hat{\beta}^{(k)} - \beta^{(k)}$, $\sigma_0 = \xi_{\max}^2(P_0^{-1})$, $C = \xi_{\max}(P_0)$, and $\delta_0 = E\left[\|\widetilde{\beta}^{(0)}\|^2\right]$. $\xi_{\max}(x)$ is the maximum eigenvalue of the matrix x.

Proof. Define the output weight updated error vector as:

$$\widetilde{\beta}^{(k)} = \hat{\beta}^{(k)} - \beta^{(k)}. \tag{23}$$

Using Equations (2), (15), (17) and (18), we can obtain:

$$\widetilde{\beta}^{(k)} = (I - \frac{\mathbf{P}_{k-1}\mathbf{H}_k^T\mathbf{H}_k}{\lambda_k + \mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^T})\widetilde{\beta}^{(k-1)}. \tag{24}$$

According to Equations (16) and (24), we have:

$$\mathbf{P}_k^{-1}\widetilde{\beta}^{(k)} = \lambda_k\mathbf{P}_{k-1}^{-1}\widetilde{\beta}^{(k-1)}. \tag{25}$$

Thus:

$$\widetilde{\beta}^{(k)} = \lambda_k\mathbf{P}_k^{-1}\mathbf{P}_{k-1}^{-1}\widetilde{\beta}^{(k-1)} = \prod_{i=1}^{k}\lambda_i\mathbf{P}_k\mathbf{P}_0^{-1}\widetilde{\beta}^{(0)}. \tag{26}$$

Consider the update of P_k^{-1}, then:

$$P_k^{-1} = \lambda_k\mathbf{P}_{k-1}^{-1} + \mathbf{H}_k^T\mathbf{H}_k. \tag{27}$$

From which we have:

$$\xi_{\min}(\mathbf{P}_k^{-1}) \geq \lambda_k \xi_{\min}(\mathbf{P}_{k-1}^{-1}) \geq \cdots \geq \prod_{i=1}^{k} \lambda_i \xi_{\min}(\mathbf{P}_0^{-1}). \tag{28}$$

Assuming $\xi_{\max}(P_0) \leq C$, the Equation (28) becomes:

$$\xi_{\max}(\mathbf{P}_{k-1}) \leq \frac{1}{\prod\limits_{i=1}^{k-1} \lambda_i} \xi_{\max}(\mathbf{P}_0) \leq \frac{C}{\prod\limits_{i=1}^{k-1} \lambda_i}. \tag{29}$$

Furthermore, for $0 < \lambda_k < 1$, the DFF satisfies:

$$0 < (\lambda_{\min})^k \leq \prod_{i=1}^{k} \lambda_i \leq (\lambda_{\max})^k \leq \lambda_{\max} < \infty, \tag{30}$$

then:

$$\begin{aligned} E\left[\|\widetilde{\beta}^{(k)}\|^2\right] &= E\left[\|\prod_{i=1}^{k} \lambda_i \mathbf{P}_k \mathbf{P}_0^{-1} \widetilde{\beta}^{(0)}\|^2\right] \\ &= \left(\prod_{i=1}^{k} \lambda_i\right)^2 E\left[\|\mathbf{P}_k \mathbf{P}_0^{-1} \widetilde{\beta}^{(0)}\|^2\right] \\ &\leq \left(\prod_{i=1}^{k} \lambda_i\right)^2 \xi_{\max}^2(\mathbf{P}_k) \xi_{\max}^2(\mathbf{P}_0^{-1}) E\left[\|\widetilde{\beta}^{(0)}\|^2\right] \\ &\leq \lambda_{\max}^2 \left(C / \prod_{i=1}^{k-1} \lambda_i\right)^2 \sigma_0 \delta_0 \end{aligned}. \tag{31}$$

This completes the proof of this theorem. □

In addition, we discuss the error limitation. Define a non-negative function S_k as:

$$S_k = \frac{[\widetilde{\beta}^{(k)}]^T \mathbf{P}_k^{-1} \widetilde{\beta}^{(k)}}{\prod\limits_{i=1}^{k} \lambda_i}. \tag{32}$$

Then, we have:

$$\begin{aligned} S_k - S_{k-1} &= \frac{[\widetilde{\beta}^{(k)}]^T \mathbf{P}_k^{-1} \widetilde{\beta}^{(k)}}{\prod\limits_{i=1}^{k} \lambda_i} - \frac{[\widetilde{\beta}^{(k-1)}]^T \mathbf{P}_{k-1}^{-1} \widetilde{\beta}^{(k-1)}}{\prod\limits_{i=1}^{k-1} \lambda_i} = \frac{1}{\prod\limits_{i=1}^{k-1} \lambda_i} \left(\widetilde{\beta}^{(k)} - \widetilde{\beta}^{(k-1)}\right)^T \mathbf{P}_{k-1}^{-1} \widetilde{\beta}^{(k-1)} \\ &= -\frac{1}{\prod\limits_{i=1}^{k-1} \lambda_i} \left(\frac{[\widetilde{\beta}^{(k-1)}]^T \mathbf{H}_k^T \mathbf{H}_k \widetilde{\beta}^{(k-1)}}{\lambda_k + \mathbf{H}_k \mathbf{P}_{k-1} \mathbf{H}_k^T}\right) = -\frac{e_k^2}{\prod\limits_{i=1}^{k} \lambda_i + \prod\limits_{i=1}^{k-1} \lambda_i \mathbf{H}_k \mathbf{P}_{k-1} \mathbf{H}_k^T} \\ &\leq -\frac{e_k^2}{\prod\limits_{i=1}^{k} \lambda_i + \prod\limits_{i=1}^{k-1} \lambda_i \xi_{\max}(\mathbf{P}_{k-1}) \|\mathbf{H}_k\|^2} \leq 0 \end{aligned}. \tag{33}$$

From Equation (33), i.e., $S_k - S_{k-1} \leq 0$, it is easy to see that S_k converges, then we can obtain:

$$\lim_{k \to \infty} \frac{e_k^2}{\prod\limits_{i=1}^{k} \lambda_i + \prod\limits_{i=1}^{k-1} \lambda_i \xi_{\max}(\mathbf{P}_{k-1}) \|\mathbf{H}_k\|^2} = 0. \tag{34}$$

Substituting Equation (29) into (34), we have:

$$\lim_{k\to\infty} \frac{e_k^2}{\prod_{i=1}^{k} \lambda_i + C\|\mathbf{H}_k\|^2} = 0, \tag{35}$$

where $\prod_{i=1}^{k} \lambda_i > 0$, $\|\mathbf{H}_k\|^2 > 0$, so we obtain $\lim_{k\to\infty} e_k = 0$.

As mentioned above, we can summarize the proposed DU-OS-ELM in the following steps:

Step 1: Determine the model parameters by initial data $S_0 = \{x_j, t_j | j = 1, \ldots, N_0\}$, such as hidden nodes number L, activation function $G(\mathbf{a}_i, b_i, x_i)$ and the initial value of the DFF;

Step 2: Randomly generate the hidden layer parameters $(\mathbf{a}_i, b_i)$;

Step 3: Calculate the initial hidden layer output matrix $\mathbf{H}_0$ and the initial output weight $\beta^{(0)}$ by Equation (8), then generate the initial model;

Step 4: Calculate the predicted error of the newly incremental data by the current model;

Step 5: Judge whether the model should be updated by Equation (21), if the model needs not to be updated, the new data will be added directly and the output weight will remain unchanged. If the model needs to be updated, λ_k is calculated by Equation (19) and then the output weight will be recalculated by Equation (18);

Step 6: Output the final output weight $\beta^{(k+1)}$.

It should note that the steps 4–5 are used for sequential learning, so the algorithm will repeat these two steps when there are newly arriving data.

4. Soft-Sensor Model for GUR Prediction in BF Based on DU-OS-ELM

As shown in Figure 3, a BF is essentially a large countercurrent heat exchanger and chemical reactor. The reactants including iron ore, coke and limestone are fed into the top layer by layer. In addtion, preheated air and additional oxygen are occasionally blown into the bottom through the tuyeres. A series of complicated chemical reactions and heat transport phenomena occur in the different zones throughout the BF under the different temperatures due to the downward movement of raw materials and the upward flow of hot gases, which can produce a large amount of heat energy that can make the temperature of a BF reach about 2000 °C. The molten pig iron and slag are produced as separating output streams at certain intervals as the production continues. The whole ironmaking cycle will take about 6–8 h [37,44]. The main chemical reactions during the ironmaking process are described by the following equations [25]:

$$3Fe_2O_3 + CO \xrightarrow{450^\circ C} 2Fe_3O_4 + CO_2, \tag{36}$$

$$Fe_3O_4 + CO \xrightarrow{600^\circ C} 3FeO + CO_2, \tag{37}$$

$$FeO + CO \xrightarrow{\geq 700^\circ C} Fe + CO_2, \tag{38}$$

$$FeO + C \xrightarrow{\geq 700^\circ C} Fe + CO. \tag{39}$$

Meanwhile, the coke descends downward and reacts with the upwardly flowing preheated air to produce heat immediately:

$$C + O_2 \rightarrow CO_2. \tag{40}$$

Then, the carbon dioxide reacts with excess carbon to produce carbon monoxide with high temperature:

$$CO_2 + C \rightarrow 2CO. \tag{41}$$

According to the above analysis, carbon monoxide plays a major role when the iron ore is reduced. Thus, the degree of utilization of carbon monoxide directly affects the progress of the chemical reactions occurring inside the furnace and affects the energy consumption of each ton of iron. Therefore, GUR is an important indicator of energy consumption and operating status in BF iron-making processes. The existing studies on GUR prediction mainly focus on mechanism analysis and fixed leaning samples modeling. However, the BF ironmaking process is multi-variable, nonlinear, time-variant and high complex in nature [45], therefore, it is difficult to implement fast and accurate GUR predictions with the existing models and they are not adequate for online use. Besides, due to the fast development of measurement instruments and sensor techniques, we can obtain a large number of data from BF ironmaking processes. Therefore, we propose an online soft-sensor model for GUR prediction based on the proposed DU-OS-ELM.

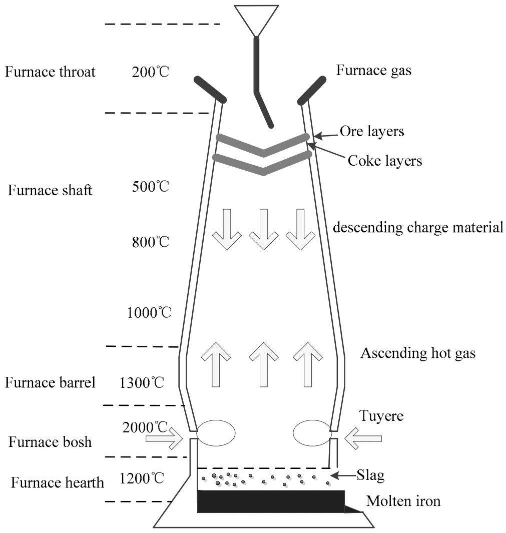

Figure 3. Blast furnace ironmaking process.

There are many parameters that can influence GUR due to BFs' high complexity. It should be noted that the selection of too many parameters as input variables may increase the complexity of model while too few parameters will degrade the model performance. Hence, the influence of parameter selection for a GUR prediction model plays an important role in improving the model accuracy. The GUR depends on the chemical reactions inside the furnace during the ironmaking process. Blast temperature is the power of the BF gas upward movement. The higher the blast temperature is, the greater the heat blowing into the BF. Thus, increasing the blast temperature can effectively promote the chemical reactions and provide temperature compensation. Under the impact of the pressure difference between the blast pressure and top pressure, gas flows from the bottom upwards to ensure that the chemical reactions between the ore and the coke are carried out normally. If the top pressure increases, the gas velocity will slow down and chemical reactions occur more fully. Blast pressure and blast volume are usually characterized by the permeability index, which has a significant influence on the block zone and gas flow distribution. The top temperature directly reflects the degree of heat exchange between the gas flow and the furnace charge. The purpose of oxygen enrichment is to increase the oxygen content in the air, which contributes to the chemical reaction processes. Blast velocity affects the activity degree of the hearth, the size of the combustion zone and gas flow distribution. Injection of fuels can provide more reducing agents and heat, and also promote the progress of indirect reduction reactions. Thermal load can reflect the reactions between the edge charge and gas. Based on BF expert knowledge and the mechanism of the iron production process, 10 parameters, i.e., blast temperature, blast pressure, blast volume, top pressure, top temperature, permeability index, oxygen enrichment percentage, blast velocity, injecting coal rate, and thermal load, are chosen as the candidate inputs features for modeling. Simultaneously, a grey relational analysis (GRA) method [46] is used to analyze the correlation between these parameters and GUR. The required input variables and correlation analysis results by GRA are listed in Table 1.

Table 1. Input variables description and correlation analysis results.

Variable Name	Unit	Correlation Grades
Blast volume	m^3/min	0.8662
Blast temperature	°C	0.9237
Blast pressure	kPa	0.8795
Top pressure	kPa	0.7064
Top temperature	°C	0.8276
Permeability index	$m^3/min{\cdot}kPa$	0.7856
Oxygen enrichment	wt%	0.8136
Blast velocity	m/s	0.4844
Pulverized coal injection	t	0.5023
Thermal load	kJ/h	0.4087

The greater the grey correlation grades between these parameters and GUR, the stronger the correlation degree is. Here we choose the input parameters with relatively high correlation with GUR. According to the experiential knowledge and results obtained by GRA, the parameters with correlation grades greater than 0.7, i.e., blast temperature, blast pressure, blast volume, top pressure, top temperature, permeability index, and oxygen enrichment percentage, are selected as the final input variables of proposed soft-sensor model, and GUR is the only output variable. Eventually, the proposed DU-OS-ELM is applied to establish the GUR prediction model, which is illustrated in Figure 4.

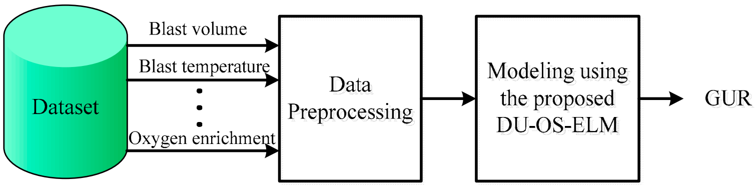

Figure 4. Main frame of the soft-sensor model for GUR prediction.

5. Simulation Results and Analysis

In this section, we present the simulation results of the proposed DU-OS-ELM applied to the GUR prediction model. In addition, we compare the performance of DU-OS-ELM with ELM, OS-ELM and FOS-ELM. All the data in the experiments are from the actual production of a steel works located in China. Besides, in order to verify the validity of the USS, we employ a DU-OS-ELM without the USS as the comparison, which is named DOS-ELM. The sigmoidal additive activation function, i.e., $G(a,b,x) = 1/(1+\exp(-(a \cdot x + b)))$, is adopted in the following experiments, where the input weights and the bias are randomly chosen from the range $[-1, 1]$. All the experiments are carried out in MATLAB 7.11.0 environment running on a desktop computer equipped with an AMD Athlon(tm)IIX2 250 processor operating at 3.00 GHz and with 2.00 GB RAM.

Remark 1. *In order to further illustrate the effectiveness of proposed DU-OS-ELM, we apply DU-OS-ELM to the well-known time series prediction problem and hot metal temperature prediction problem. The detailed description of these simulation results which is omitted here for sake of brevity is presented in Appendixs A and B.*

5.1. Data Preprocessing

The data are collected from a medium-size BF with an inner volume of about 2500 m^3. We collect 1500 data pairs in all after processing the outlier values. The first 1200 input-output instances are used for training while the others are used for testing to evaluate the performance of the model.

Next, we calculate some important statistical properties of the variables on the selected data and make the further analysis to get deeper understanding. The statistical properties include maximum, minimum, mean, standard deviation (SD). The results are shown in Tables 2 and 3, respectively. Table 2 illustrates the statistical properties of GUR in respect of the divided training and testing sets. According to Table 2, the training and testing sets have different statistical properties, so it can better explain the performance of the predicted results. Table 3 details the statistical properties of the variables. Based on the statistical properties, it can be found that the selected data has the characteristics of violent fluctuation. Figure 5 demonstrates the series of GUR and blast volume measured from BF. According to Table 3 and Figure 5, the magnitudes of the variables have big difference clearly. In fact, the effect of the variables with a large magnitude on the modeling is larger than the one with a small magnitude, thus it is not appropriate to directly take the data to establish the model [47]. Thus, all the data are normalized into (0, 1) with the same magnitude to eliminate the influence of dimension among variables before applying in the experiments. The method can be referred Equation (A3) in the Appendix C.

Table 2. Statistical properties of GUR in terms of divided training and testing set.

Set	Maximum	Minimum	Mean	SD
Train	51.208	47.651	50.271	0.4404
Test	50.574	47.879	49.476	0.5379

Table 3. Statistical properties of the variables.

No.		Variable	Maximum	Minimum	Mean	SD
input	1	Blast temperature	1180.4	1144.4	1163.6	7.29
	2	Blast pressure	347.11	309.72	337.36	4.44
	3	Blast volume	4353.2	4199.4	4287.0	20.98
	4	Top pressure	190.40	177.83	185.38	1.74
	5	Top temperature	305.49	95.35	182.52	36.30
	6	Permeability index	0.81	0.58	0.75	0.0277
	7	Oxygen enrichment	44.79	37.68	41.24	1.1469
output	1	Gas utilization ratio	51.208	47.651	50.112	0.561

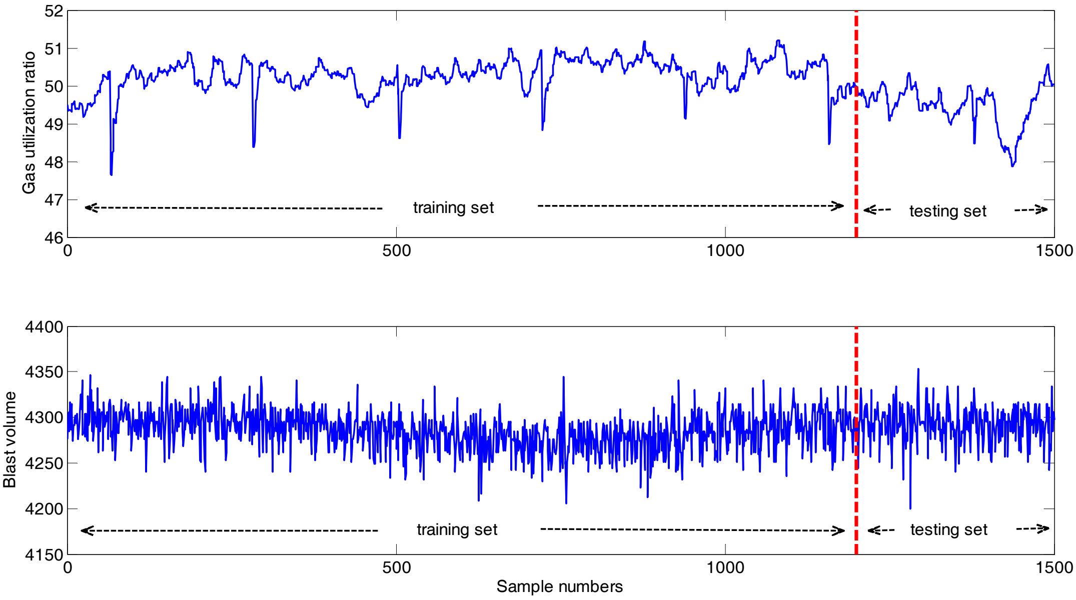

Figure 5. Series of GUR and blast volume in BF.

5.2. Performance Evaluation

To construct the GUR prediction model using the proposed DU-OS-ELM, it is necessary to determine the related parameters. In DU-OS-ELM, there are three important parameters which need to be determined, i.e., the regularization factor γ, the number of hidden nodes L, and the step size μ for controlling the forgetting rate. After some experiments we set $\gamma = 10$ in this paper. The number of hidden nodes with the minimum training error is taken as the optimal L [14]. The model selection of the optimal L for DU-OS-ELM is shown in Figure 6. According to Figure 6, the green and blue lines on the left stand for the training and testing errors (averaged over 50 trials), and the red line on the right stands for the training time, respectively. In this simulation, we increase L from 1 to 50. Accordingly, the intervals from 1 to 15 are relatively small and the hidden nodes increase 5 by 5 from 15 to 50. As shown in Figure 6, with the increase of L, the root mean squared error (RMSE) of the model is smaller gradually and the training time is increased. The lowest testing error is achieved when L is within the range (15, 50) with smooth RMSE curves. Therefore, one can select the optimal L from this range. Considering the computational complexity and testing accuracy, we choose L equals to 20 as a good trade-off.

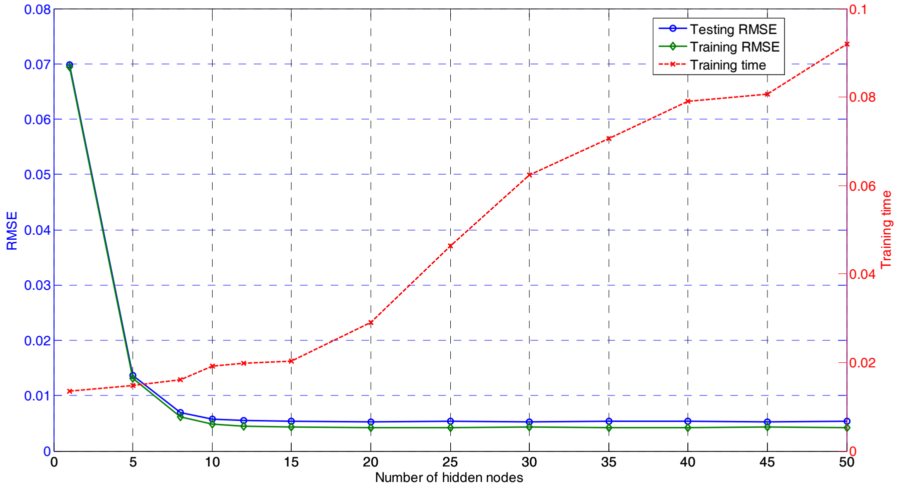

Figure 6. Selection of the hidden nodes number.

One of the contributions of the proposed DU-OS-ELM is the utilization of a novel DFF. In Equation (19), the step size μ is an important user specified parameter that also needs to be determined. Figure 7 presents the testing accuracy with different step sizes μ in the range [1, 10]. We increase μ 1 by 1. As observed from Figure 7, the minimum testing accuracy is obtained when μ equals 5, which will be used in the following experiments.

The convergence of DU-OS-ELM is analyzed theoretically in Section 3.3, and a simulation evaluation is also performed to show the convergence speed among DU-OS-ELM, DOS-ELM and OS-ELM. The number of initial training data N_0 has been taken as 100 and the size of block of data learned in each step is 20 in this simulation. Figure 8 details the changing trends among DU-OS-ELM, DOS-ELM and OS-ELM. According to this figure, as the number increment proceeds, the testing error tends to decrease, and DU-OS-ELM obtains the smallest testing error with the fastest convergence speed compared with DOS-ELM and OS-ELM, which illustrates that the proposed DU-OS-ELM has better dynamic tracking ability due to the utilization of the DFF and the USS. DOS-ELM tends to be stable when the number of increment approximately equals to 35 in the black line. However, OS-ELM tends to be stable when the number of increments is approximately equal to 50 in the blue line. The comparison results show that DOS-ELM has faster convergence speed than OS-ELM. In addition, we can find that the red line is smoother than others, which implies that DU-OS-ELM is more suitable for time-varying environments. According to the aforementioned analysis, the proposed DU-OS-ELM has faster convergence speed and is more effective for the time-varying environments.

The changes of the DFF λ_k in Equation (19) of DU-OS-ELM (red line) and DOS-ELM (black line) are depicted in Figure 9. We can find that the DFF of DOS-ELM is updated iteratively depending on the predicted error. However, due to the addition of the USS, the DFF of DU-OS-ELM is not updated in each iteration, for example, there are constant segments when the number of increments equals 23, 34, 50, respectively. This proves the effectiveness of the USS and indicates that there is no need to update the model in each time step. In addition, according to the changes of the DFF, we can understand the changing status of the production process.

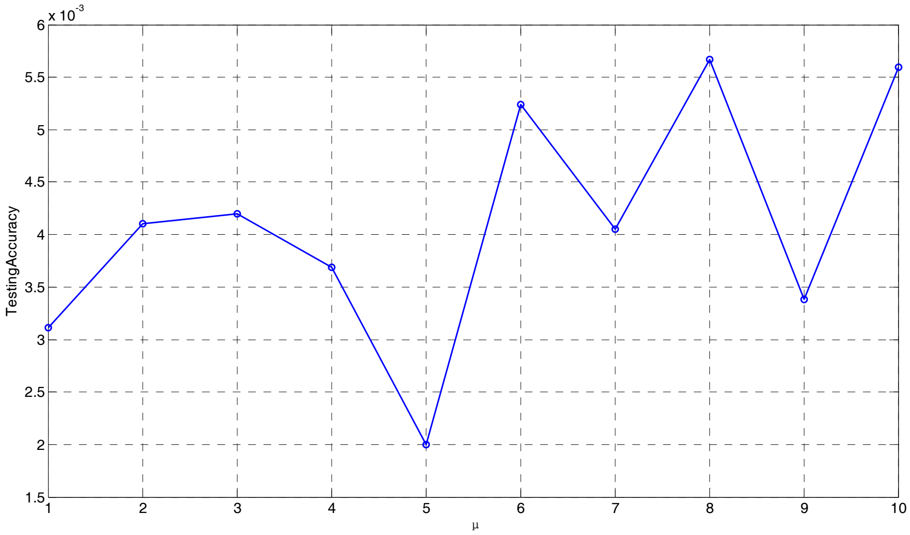

Figure 7. Selection of step size μ.

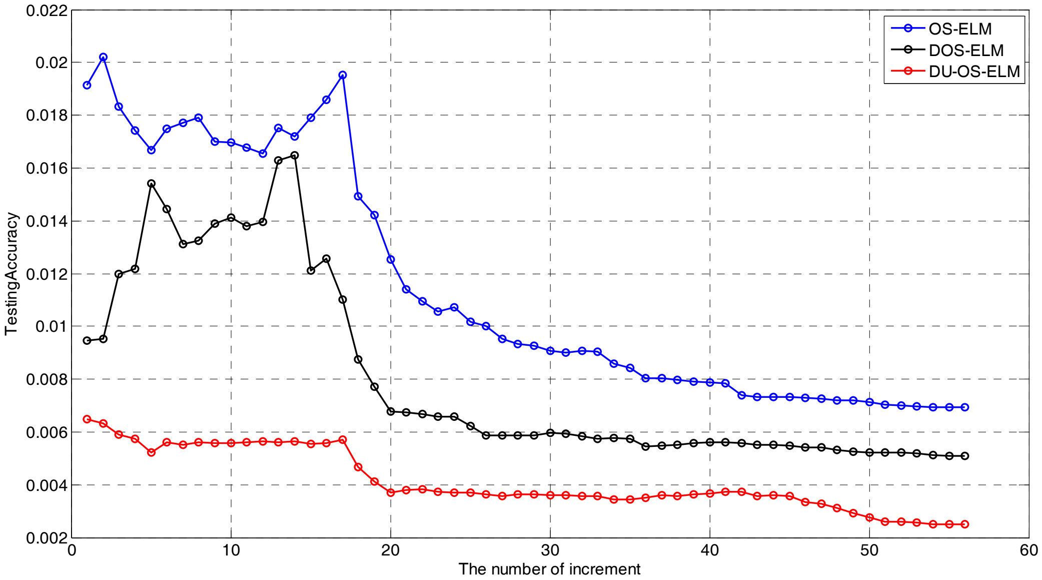

Figure 8. Change trends comparison among OS-ELM, DOS-ELM and DU-OS-ELM.

Figure 10 shows comparison results of DU-OS-ELM, DOS-ELM, FOS-ELM, and OS-ELM in a single trial. The parameter is set as $s = 4$ in FOS-ELM. The number of initial data is 100 and the size of block of data learned in each step is 10 in DU-OS-ELM, DOS-ELM and OS-ELM. The black line is the actual output value. As observed from Figure 10, the predicted results of the four approaches are basically consistent with the actual trend, which can reflect the changing trend of GUR, but there are differences in the agreement degree. Obviously, the predicted result of OS-ELM shows the lowest agreement with the actual values. The predicted results of DOS-ELM, FOS-ELM and OS-ELM have little difference, but DU-OS-ELM is better than the other three approaches. In order to identify the simulation results more clearly, the results between 60 and 100 and between 220 and 250 are magnified. As can be found from local amplified drawing in Figure 10, DU-OS-ELM is more close to the actual output value compared with other three approaches, which shows that the proposed DU-OS-ELM provides more accurately predicted results.

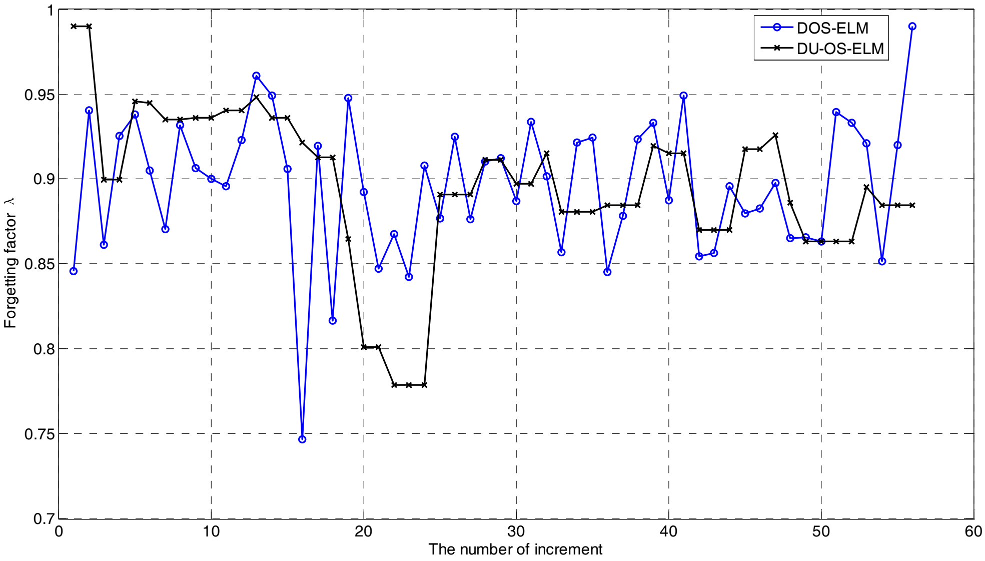

Figure 9. λ_k by DOS-ELM and DU-OS-ELM.

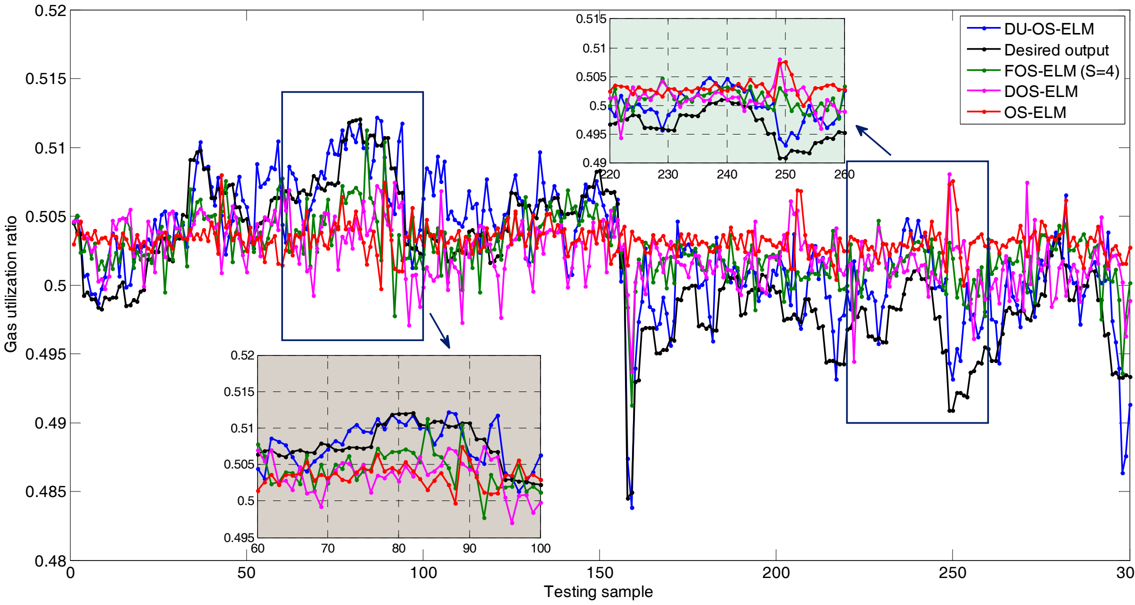

Figure 10. Simulation results of different approaches.

Figure 11 presents the correlation between the predicted values of the different approaches and the desired values. As observed from Figure 11, most of the predicted values of the proposed DU-OS-ELM are close to $y = x$, but the predicted values of the other approaches are relatively far away from $y = x$. In addition, the calculated results show that the correlation coefficient of DU-OS-ELM, DOS-ELM, FOS-ELM and OS-ELM equal to 0.8813, 0.7301, 0.7275 and 0.6613, respectively, which indicates that the proposed DU-OS-ELM has better prediction performance than others.

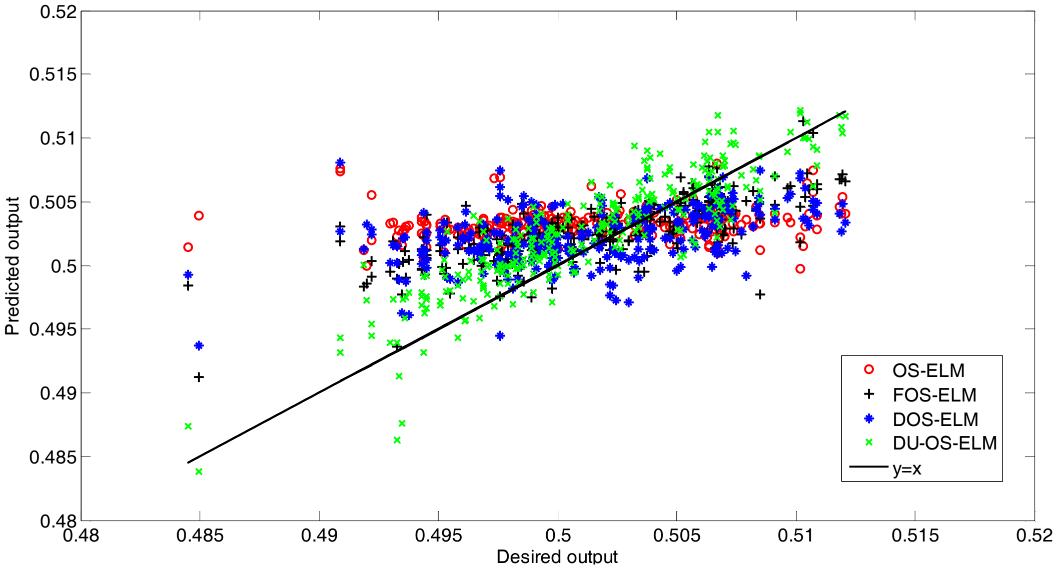

Figure 11. The correlation between the predicted values and desired values.

In order to evaluate the performance of the proposed DU-OS-ELM quantitatively, the following frequently criteria are used in the experimental results: training time, testing time, training RMSE, testing RMSE, mean absolute percentage error (MAPE) [48] and SD [6,7,15]. RMSE makes an excellent general purpose error metric for prediction model. MAPE is a relative value and usually expresses accuracy as a percentage. The smaller the RMSE and MAPE, the better the prediction accuracy is. SD is a measure of the dispersion degree of many experiment results. The smaller the SD, the more stable the algorithm is. The mathematical representations of the three statistics are shown in Equations (A4)–(A6) in Appendix C.

The performance of the proposed DU-OS-ELM is evaluated by one-by-one learning mode and chunk-by-chunk learning mode with 5 and 10 chunk sizes. Fifty trials are carried out in each case. The results are averaged and summarized in Table 4. According to Table 4, the training time decreases as the chunk size increases. In addition, it is worth noting that the testing accuracy only has a little change as the chunk size increases. For the same learning mode, the training time (learning time) of DU-OS-ELM is relatively longer than DOS-ELM due to the utilization of the USS, but the output weight is not always updated, thus, the training time of DU-OS-ELM does not increase obviously. Overall, the predicted accuracy of DU-OS-ELM is better than DOS-ELM. Meanwhile, both of the two approaches have smaller SD, however, DU-OS-ELM is smaller than DOS-ELM. It shows that the former is more stable than the latter. The above analysis once again proves the effectiveness of the USS in DU-OS-ELM for time-varying environments.

Table 4. Comparison results of DU-OS-ELM and DOS-ELM.

Algorithm	Learning Mode	Training Time (s)	RMSE		SD		MAPE	#nodes
			Training	Testing	Training	Testing		
DOS-ELM	1 by 1	0.1053	0.0031	0.0034	5.8480×10^{-4}	7.6142×10^{-4}	0.0622	20
	5 by 5	0.0406	0.0030	0.0033	5.4541×10^{-4}	7.2880×10^{-4}	0.0629	20
	10 by 10	0.0257	0.0032	0.0036	7.4076×10^{-4}	6.4927×10^{-4}	0.0637	20
DU-OS-ELM	1 by 1	0.1193	0.0029	0.0032	5.0033×10^{-4}	6.5348×10^{-4}	0.0611	20
	5 by 5	0.0616	0.0030	0.0034	4.9858×10^{-4}	6.4049×10^{-4}	0.0626	20
	10 by 10	0.0328	0.0029	0.0033	4.2275×10^{-4}	5.0429×10^{-4}	0.0619	20

The compared results of different approaches are given in Table 5. Since batch ELM does not have the ability to learn sequentially, we use the entire training set to train, and other approaches are updated with the new data online. A fixed chunk size of 10 is selected for DU-OS-ELM, DOS-ELM and OS-ELM in the chunk-by-chunk learning mode. It can be found from Table 5 that DU-OS-ELM achieves the best generalization performance with the smallest RMSE and MAPE compared with other approaches. In terms of training time, batch ELM takes the least time. After adding DFF and USS, the training time of DU-OS-ELM is a little longer than that of FOS-ELM, but shorter than that of OS-ELM, which shows that DU-OS-ELM does not add too much computational complexity and is easy to implement. In addition, the time consumption of DU-OS-ELM is acceptable in actual production. Therefore, DU-OS-ELM can achieve the accurate GUR prediction, and can satisfy the production requirements.

Table 5. Comparison results of different approaches.

Algorithm	Training time(s)	RMSE		SD		MAPE	#nodes	Remark
		Training	Testing	Training	Testing			
ELM	0.0156	0.0068	0.0076	2.6834×10^{-5}	7.3714×10^{-5}	0.1086	20	batch
OS-ELM	0.0413	0.0055	0.0064	3.5139×10^{-5}	9.3708×10^{-5}	0.0822	20	10 by 10
FOS-ELM	0.0281	0.0043	0.0051	5.3216×10^{-4}	7.7670×10^{-4}	0.0794	20	s = 4
DOS-ELM	0.0257	0.0032	0.0036	7.4076×10^{-4}	6.4927×10^{-4}	0.0637	20	10 by 10
DU-OS-ELM	0.0328	0.0029	0.0033	4.2275×10^{-4}	5.0429×10^{-4}	0.0619	20	10 by 10

6. Discussion

The motivations of the proposed DU-OS-ELM are mainly in the following three aspects:

(1) Give more attention on the newly acquired data and forget the old collected data gradually.
(2) Save storage space for matrix and avoid data saturation problem.
(3) Improve the dynamic tracking ability, reduce the time consumption and avoid the errors accumulation.

Therefore, the main contributions of the proposed DU-OS-ELM include: (1) a novel DFF with low computational complexity depending on the estimation errors is presented to improve the dynamic tracking ability in time-varying environments. The trained model can more closely track dynamic changes of data; (2) the USS is developed to judge whether the model should be updated with the newly coming data in each time step to avoid the errors accumulation and reduce the time consumption; (3) an online soft-sensor model is established based on the proposed DU-OS-ELM to implement the fast and accurate GUR prediction in BF ironmaking process. In order to further illustrate the superiority of the proposed approach, the characteristics including advantage and limitation of various approaches (including ELM, OS-ELM, FOS-ELM, DOS-ELM and DU-OS-ELM) are summarized in Table 6.

As observed from Table 6, the proposed DU-OS-ELM approach belongs to online learning with forgetting mechanism and update strategy for non-stationary or time-varying environments. However, according to the above experiments, the predicted results of DU-OS-ELM depend on the step size μ in Equation (19). A reasonable μ needs to be determined. It is worth noting that DU-OS-ELM becomes OS-ELM when $\lambda = 1$. Thus, OS-ELM can be seen as a special case of DU-OS-ELM when all the observations are treated equally. In addition, a fixed sliding window, i.e., the collected old data are given 0 weight and the newly acquired data are given 1 weight in active area, is added in the FOS-ELM. From an extreme point of view, FOS-ELM can also be regarded as a special case of DU-OS-ELM.

The GUR soft-sensor model can reflect well the strong nonlinear relationships between the energy consumption and operating parameters, which is difficult to describe by any mechanistic model due to the high complexity of the reactions occurring inside a furnace. In addition, the application of this work will help BF operators judge the changes of energy consumption indicator in time properly, which provides a guide for them to determine the direction of controlling BF in advance.

Table 6. Characteristics comparison of different approaches.

Algorithm	Learning Mode	Forgetting Mechanism	Update Strategy	Data Saturation	Application Scope	Limitation
ELM	Batch learning	-	-	-	Finite number of samples	The predicted results are limited by the initial training samples
OS-ELM	Online learning	No	No	Yes	Real-time and stationary situations	All the observations are treated equally
FOS-ELM	Online learning	Fixed sliding window	No	No	Short-term online prediction	The old data are discard directly when the new ones arrived
DOS-ELM	Online learning	Adaptive forgetting factor	No	No	Nonstationary or time-varying environments	Step size μ needs to be reasonably defined
DU-OS-ELM	Online learning	Adaptive forgetting factor	Yes	No	Nonstationary or time-varying environments	Step size μ needs to be reasonably defined

7. Conclusions

This paper proposes an effective online learning approach named DU-OS-ELM with better generalization performance in the GUR soft-sensor model. In the proposed DU-OS-ELM, the novel DFF and USS are developed under the framework of OS-ELM to improve the performance in time-varying production environments. The DFF can automatically adjust with the dynamic change of production process to improve the dynamic tracking ability of the model. On the other hand, the USS is added to further meet the actual production situation. The real production data are employed to validate the effectiveness of the proposed DU-OS-ELM. The comparison experimental results indicate that the GUR soft-sensor model based on the proposed DU-OS-ELM has higher accuracy and smaller SD than others. The soft-sensor model can provide effective decision support for the optimization operation, energy savings and emission reduction of BF ironmaking processes.

Acknowledgments: This work is supported by National Natural Science Foundation of China (NSFC Grants no. 61333002 and no. 61673056) and Beijing Key Subject Construction Projects (no. XK100080573).

Author Contributions: Yanjiao Li proposed DU-OS-ELM algorithm and the soft-sensor model for GUR prediction. Sen Zhang supervised the work and revised the paper. Yixin Yin and Wendong Xiao involved in the discussion of the paper and provided many helpful advices. Jie Zhang helped to perform some experiments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this appendix, we verify the effectiveness of the proposed DU-OS-ELM using the well-known Mackey-Glass chaotic time series. The data are generated by the following time-delay differential equation [15,20]:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t), \tag{A1}$$

where $x(t)$ is the output value at time t, τ is delay time and $\tau = 17$. We set the initial value $x(0) = 1.2$ and $x(t) = 0$ for $t < 0$. In order to create the time-vary time series, a sinusoid with amplitude 0.2 is added to the series as shown in Figure A1. We predict the output $y(t) = x(t+6)$ from the past values of this time series, that is, $x(t)$, $x(t-6)$, $x(t-12)$, $x(t-18)$, so the data format is:

$$[x(t), x(t-6), x(t-12), x(t-18); x(t+6)]. \tag{A2}$$

We collect 2200 data pairs of the above format. The first 1700 input-output instances are used for training while the others are used for testing. In this simulation, OS-ELM [15] and FOS-ELM [21] are conducted for comparison. The number of initial data is 200 and the size of block of data learned in each step is 10 in DU-OS-ELM, DOS-ELM and OS-ELM. The parameter is set as s = 4 in FOS-ELM. Figure A2 shows the simulation results of Mackey-Glass Chaotic Time Series with different approaches.

Figure A3 presents the error comparison results of different approaches. In order to identify the simulation results more clearly, the results between 180 and 220 are magnified. As can be found from local amplified drawing in Figure A3, DU-OS-ELM has less error compared with other three approaches, which shows that the proposed DU-OS-ELM presents more accurate predicted results on Mackey-Glass chaotic time series data.

The performance comparison results are shown in Table A1. As observed from Table A1, DU-OS-ELM achieves the best performance with the smallest RMSE and MAPE. The time consumption of DU-OS-ELM is similar to others. Therefore, the addition of DFF and USS does not bring computational burden.

Table A1. Comparison results of different approaches.

Algorithm	Training Time (s)	RMSE		MAPE	#Nodes	Remark
		Training	Testing			
ELM	0.5148	0.0461	0.0441	0.0296	200	batch
OS-ELM	1.2792	0.0445	0.0432	0.0274	200	10 by 10
FOS-ELM	1.2366	0.0436	0.0420	0.0229	200	$s = 4$
DOS-ELM	1.2324	0.0420	0.0404	0.0198	200	10 by 10
DU-OS-ELM	1.2480	0.0411	0.0398	0.0183	200	10 by 10

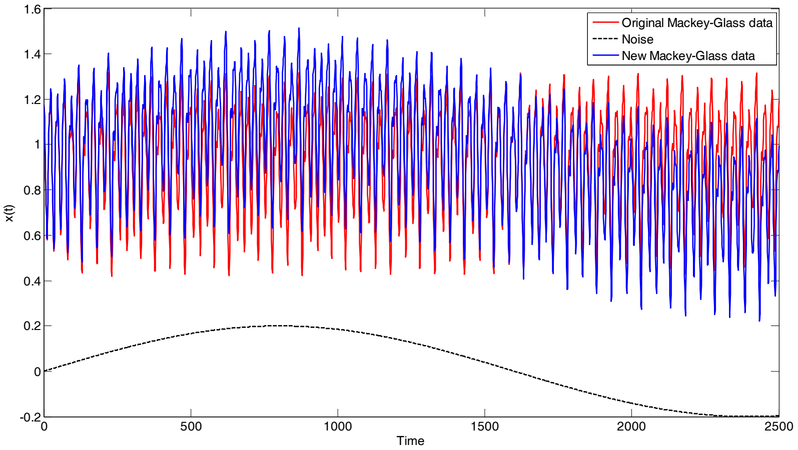

Figure A1. Mackey-Glass chaotic time series.

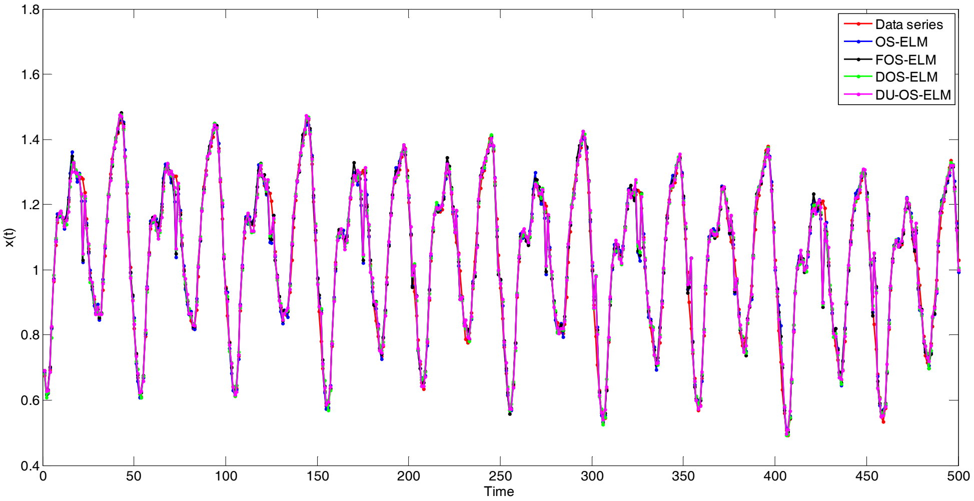

Figure A2. Simulation results of Mackey-Glass chaotic time series.

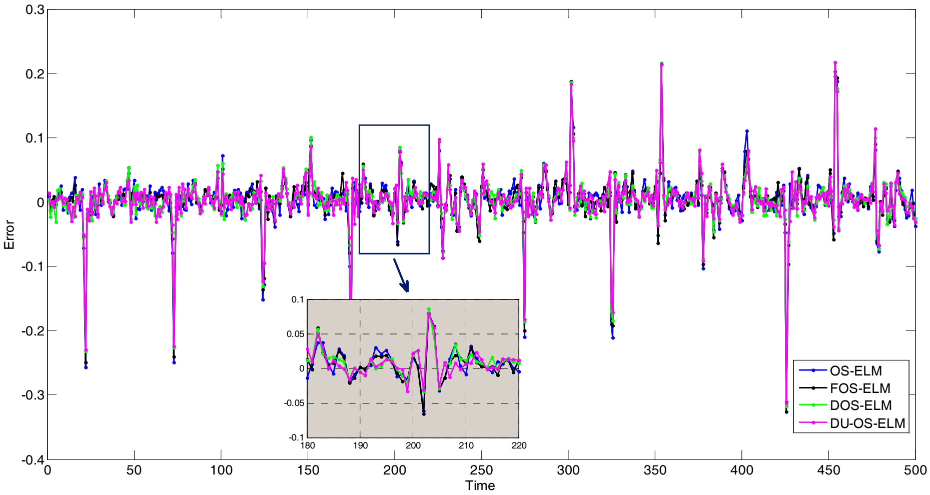

Figure A3. Error results of Mackey-Glass chaotic time series.

Appendix B

Another important issue in the BF ironmaking process is the hot metal temperature prediction. A soft-sensor model of hot metal temperature is established to further prove the contribution of the proposed DU-OS-ELM in this appendix. Experiments are reported to demonstrate the performance and effectiveness of the proposed approach.

Hot metal temperature is used as a chief indictor of the thermal status inside the BF, which determines the quality of the resulting hot metal. Therefore, it needs to be controlled within a certain range. However, the measurement of hot metal temperature can be implemented only off-line by soft-sensor methods, so the result is lagging behind and against the real-time performance of hot metal temperature monitoring [49,50]. The effective soft-senor approach is a good choice to establish the hot metal temperature prediction based on the easily measured parameters. Additionally, BF ironmaking is a typical dynamic and time-varying production process with hundreds of physical and chemical reactions occurring [37]. The prediction model based on fixed learning samples cannot satisfy the needs of online use. Besides, all the data treated equally may not be sufficient to track all the system dynamics. For actual production situation, when the system is relatively stable or the current training model can meet the prediction requirement, it may not need to update in each time step. Therefore, the proposed DU-OS-ELM is more suitable for establishing the prediction model of hot metal temperature.

Six relevant variables, including blast volume, blast temperature, blast present, pulverized coal injection, oxygen enrichment and permeability index, are selected as input variables. The hot metal temperature is the only output variable. We also collect 1500 data pairs in all. The first 1200 input-output instances are used for training while others are used for testing to evaluate the performance of the model. In order to make comparison, we also use the OS-ELM, FOS-ELM, DOS-ELM and DU-OS-ELM to build the prediction model of hot metal temperature. The simulation results of these learning approaches are shown in Figure A4. In order to identify easily, we magnify the results between 70 and 110, where one can found that DU-OS-ELM can track the actual value better than other three approaches. Simultaneously, the correlation between the predicted values of the different approaches and actual values are demonstrated in Figure A5 to further illustrate the superiority of the proposed approach. As observed from Figure A5, the predicted values of the proposed DU-OS-ELM is closer to y = x than those of other approaches. The correlation coefficient of DU-OS-ELM, DOS-ELM, FOS-ELM and OS-ELM equal to 0.9621, 0.9579, 0.9087 and 0.8751, respectively, which indicates that DU-OS-ELM has better prediction performance than the other methods. Table A2 summarizes the performance comparison results. From Table A2, the results also show that DU-OS-ELM achieves the best performance. According to the above experimental results, DU-OS-ELM can achieve more accurate online prediction of hot metal temperature. Thus, compared with other approaches, DU-OS-ELM is more suitable for hot metal temperature in time-varying environments. In addition, the results can guide the BF operators to take reasonable measures (including control span and the control direction) in the next step.

Table A2. Comparison results of different approaches.

Algorithm	Training Time (s)	RMSE		MAPE	#nodes	Remark
		Training	Testing			
ELM	1.1700	0.0685	0.0754	0.1442	500	batch
OS-ELM	4.6176	0.0621	0.0688	0.1284	500	10 by 10
FOS-ELM	4.3992	0.0603	0.0638	0.1166	500	$s = 4$
DOS-ELM	4.2900	0.0536	0.0561	0.0961	500	10 by 10
DU-OS-ELM	4.3212	0.0527	0.0548	0.0922	500	10 by 10

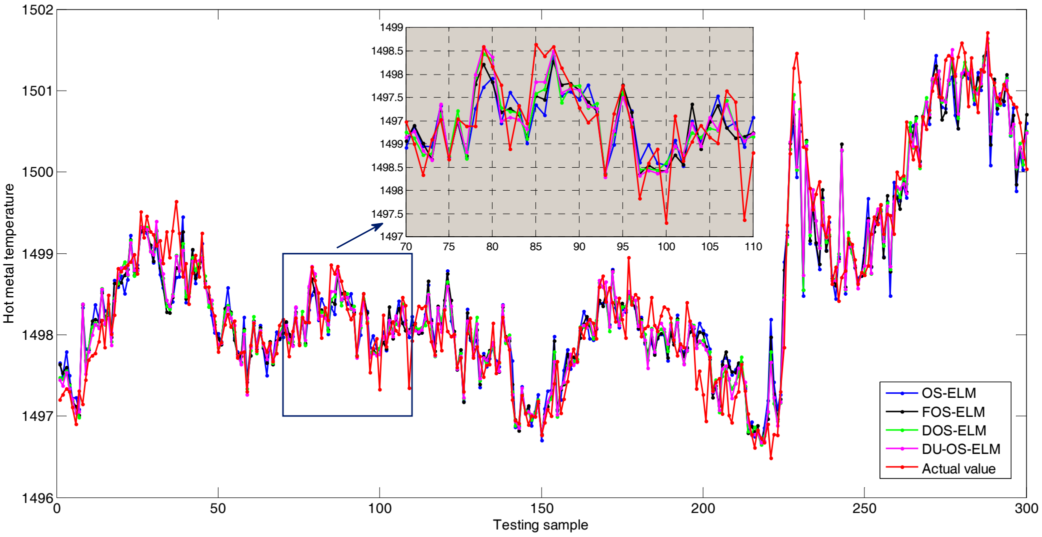

Figure A4. Prediction results of hot metal temperature based on different algorithms.

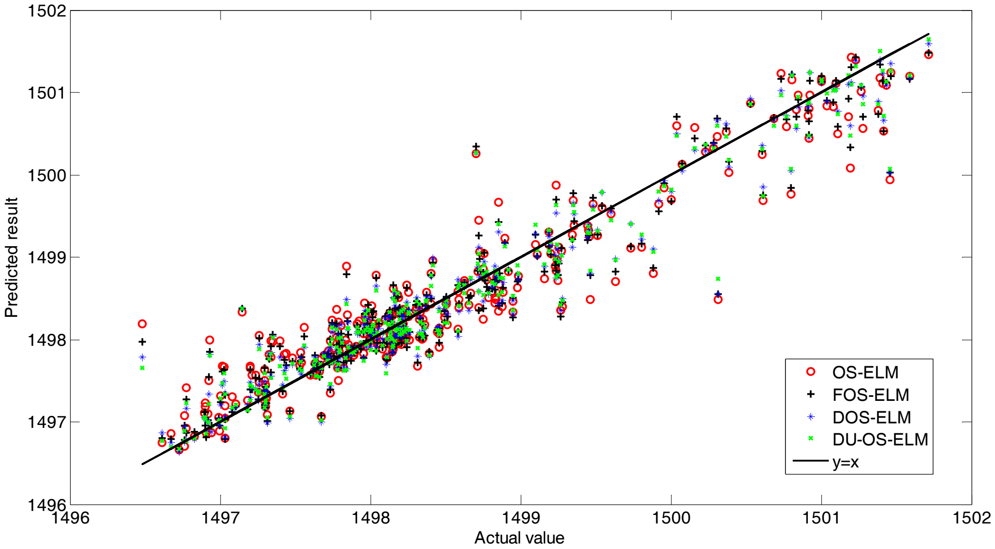

Figure A5. The correlation between the predicted values and actual values.

Appendix C

List of related equations:

Data normalization processing method:

$$x_k = \frac{\widetilde{x}_k - \min(\widetilde{x}_k)}{\max(\widetilde{x}_k) - \min(\widetilde{x}_k)}, \tag{A3}$$

where $\widetilde{x}_k$ and x_k represent the kth variables values before and after change respectively, $\max(\widetilde{x}_k)$ and $\min(\widetilde{x}_k)$ are the maximum and minimum values of the kth variable before normalized.

RMSE:

$$RMSE = \frac{1}{50}\sum_{k=1}^{50}\left(\sqrt{\frac{\sum\limits_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}\right), \tag{A4}$$

MAPE:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\%, \tag{A5}$$

SD:

$$SD = \sqrt{\sum_{i=1}^{n}\frac{\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right| - \bar{e}\right)^2}{n}}, \tag{A6}$$

where k is the kth trial, n is the number of the data in the testing set, y_i is the actual value, $\hat{y}_i$ is the predicted value, and $\bar{e}$ is the average value of the relative error.

References

1. Huang, G.B.; Chen, Y.Q.; Babri, H.A. Classification Ability of Single Hidden Layer Feedforward Neural Networks. *IEEE Trans. Neural Netw.* **2000**, *11*, 799–801. [CrossRef] [PubMed]
2. Ferrari, S.; Stengel, R.F. Smooth Function Approximation Using Neural Networks. *IEEE Trans. Neural Netw.* **2005**, *16*, 24–38. [CrossRef] [PubMed]
3. Asteris, P.G.; Roussis, P.C.; Douvika, M.G. Feed-Forward Neural Network Prediction of the Mechanical Properties of Sandcrete Materials. *Sensors* **2017**, *17*, 1344. [CrossRef] [PubMed]
4. Heskes, T.M.; Slijpen, E.; Kappen, B. Learning in Neural Networks with Local Minima. *Phys. Rev. A* **1992**, *46*, 5221–5231. [CrossRef] [PubMed]
5. Wilamowski, B.; Yu, H. Neural Network Learning without Backpropagation. *IEEE Trans. Neural Netw.* **2010**, *21*, 1793–1803. [CrossRef] [PubMed]
6. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme Learning Machine: Theory and Applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
7. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Trans. Syst. Man Cybern.* **2012**, *42*, 513–529. [CrossRef] [PubMed]
8. Huang, G.B. An Insight into Extreme Learning Machine: Random Neurons, Random Features and Kernels. *Cogn. Comput.* **2014**, *6*, 376–390. [CrossRef]
9. Fu, A.M.; Dong, C.R.; Wang, L.S. An Experimental Study on Stability and Generalization of Extreme Learning Machine. *Int. J. Mach. Learn. Cyber.* **2015**, *6*, 129–135. [CrossRef]
10. Cao, J.W.; Zhao, Y.F.; Lai, X.P.; Ong, M.E.H.; Yin, C.; Koh, Z.X.; Liu, N. Landmark Recognition with Sparse Representation Classification and Extreme Learning Machine. *J. Franklin Inst.* **2015**, *352*, 4528–4545. [CrossRef]
11. Xiao, W.; Zhang, J.; Li, Y.J.; Zhang, S.; Yang, W.D. Class-Specific Cost Regulation Extreme Learning Machine for Imbalanced Classification. *Neurocomputing* **2017**, *261*, 70–82. [CrossRef]
12. Jian, Y.L.; Huang, D.Y.; Yan, J.; Lu, K.; Huang, Y.; Wen, T.L.; Zeng, T.Y.; Zhong, S.J.; Xie, Q.L. A Novel Extreme Learning Machine Classification Model for e-Nose Application Based on the Multiple Kernel Approach. *Sensors* **2017**, *17*, 1434. [CrossRef] [PubMed]

13. Zhang, J.; Xiao, W.D.; Zhang, S.; Huang, S.D. Device-free Localization via an Extreme Learning Machine with Parameterized Geometrical Feature Extraction. *Sensors* **2017**, *17*, 879. [CrossRef] [PubMed]
14. Zou, H.; Lu, X.X.; Jiang, H.; Xie, L.H. A Fast and Precise Indoor Localization Algorithm Based on an Online Sequential Extreme Learning Machine. *Sensors* **2015**, *15*, 1804–1824. [CrossRef] [PubMed]
15. Liang, N.Y.; Huang, G.B.; Sarachandran, P.; Sundararajan, N. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1423. [CrossRef] [PubMed]
16. Lan, Y.; Soh, Y.C.; Huang, G.B. Ensemble of Online Sequential Extreme Learning Machine. *Neurocomputing* **2009**, *72*, 3391–3395. [CrossRef]
17. Deng, W.Y.; Ong, Y.S.; Tan, P.S.; Zheng, Q.H. Online Sequential Reduced Kernel Extreme Learning Machine. *Neurocomputing* **2016**, *174*, 72–84. [CrossRef]
18. Huang, S.; Wang, B.T.; Chen, Y.M.; Wang, G.R.; Yu, G. An Efficient Parallel Method for Batched OS-ELM Training Using MapReduce. *Memetic Comput.* **2016**. [CrossRef]
19. Matias, T.; Souza, F.; Araujo, R.; Goncalves, N.; Barreto, J.P. On-line Sequential Extreme Leaning Machine Based on Recursive Partial Least Squares. *J. Process Contr.* **2015**, *27*, 15–21. [CrossRef]
20. Wang, X.Y.; Han, M. Online Sequential Extreme Learning Machine with Kernels for Nonstationary Time Series Prediction. *Neurocomputing* **2014**, *145*, 90–97. [CrossRef]
21. Zhao, J.W.; Wang, Z.H.; Park, D.S. Online Sequential Extreme Learning Machine with Forgetting Mechanism. *Neurocomputing* **2012**, *87*, 79–89. [CrossRef]
22. Gu, Y.; Liu, J.F.; Chen, Y.Q.; Jiang, X.L.; Yu, H.C. TOSELM: Timeliness Online Sequential Extreme Learning Machine. *Neurocomputing* **2014**, *128*, 119–127. [CrossRef]
23. Lim, J.; Lee, S.; Pang, H.S. Low Complexity Adaptive Forgetting Factor for Online Sequential Extreme Learning Machine (OS-ELM) for Application to Nonstationary System Estimations. *Neural Comput. Appl.* **2013**, *22*, 569–576. [CrossRef]
24. Yang, K.; Choi, S.; Chung, J.; Yagi, J.I. Numerical Modeling of Reaction and Flow Characteristics in a Blast Furnace with Consideration of Layered Burden. *ISIJ Int.* **2010**, *50*, 972–980. [CrossRef]
25. Zeng, J.S.; Gao, C.H.; Su, H.Y. Data-driven Predictive Control for Blast Furnace Ironmaking Process. *Comput. Chem. Eng.* **2010**, *34*, 1854–1862. [CrossRef]
26. Pasquale, C.; Angelo, P. Optimization of Blast Furnace Productivity Coupled with CO_2 Emissions Reduction. *Steel Research Int.* **2014**, *85*, 89–98.
27. Helle, H.; Helle, M.; Pettersson, F.; Saxen, H. Multi-objective Optimization of Ironmaking in the Blast Furnace with Top Gas Recycling. *ISIJ Int.* **2010**, *50*, 1380–1387. [CrossRef]
28. Shi, L.; Wen, Y.B.; Zhao, G.S.; Yu, T. Recognition of Blast Furnace Gas Flow Center Distribution Based on Infrared Image Processing. *J. Iron Steel Res. Int.* **2016**, *23*, 203–209. [CrossRef]
29. Helle, H.; Helle, M.; Saxen, H. Nonlinear Optimization of Steel Production Using Traditional and Novel Blast Furnace Operation Strategies. *Chem. Eng. Sci.* **2011**, *66*, 6470–6480. [CrossRef]
30. Zhao, J.; Wang, W.; Liu, Y. A Two-stage Online Prediction Method for a Blast Furnace Gas System and Its Application. *IEEE Trans. Contr. Syst. Trans.* **2011**, *19*, 507–520. [CrossRef]
31. Xiao, D.F.; Yang, C.J.; Song, Z.B. The Forecasting Model of Blast Furnace Gas Output Based on Improved BP Network. *J. Zhejiang Univ. (Eng. Sci.)* **2012**, *46*, 2104–2108.
32. An, J.Q.; Chen, Y.F.; Wu, M. A Prediction Method for Carbon Monoxide Utilization Ratio of Blast Furnace Based on Improved Support Vector Regression. *CIESC J.* **2015**, *66*, 206–214.
33. Xiao, D.F.; An, J.Q.; Wu, M.; He, Y. Chaotic Prediction of Carbon-monoxide Utilization Ratio in the Blast Furnace. In Proceedings of the 35th Chinese Control Conference, Chengdu, China, 27–29 July 2016; pp. 9499–9503.
34. Xiao, D.F.; An, J.Q.; He, Y.; Wu, M. The Chaotic Characteristic of the Carbon-monoxide Utilization Ratio in the Blast Furnace. *ISA Trans.* **2017**, *68*, 109–115. [CrossRef] [PubMed]
35. Saxén, H.; Pettersson, F. Nonline Prediction of the Hot Metal Silicon Content in the blast furnace. *ISIJ Int.* **2007**, *47*, 1732–1737. [CrossRef]
36. Jian, L.; Gao, C.H.; Xia, Z.Q. A Sliding-window Smooth Support Vector Regression Model for Nonline Blast Furnace System. *Steel Res. Int.* **2011**, *82*, 169–179. [CrossRef]
37. Jian, L.; Gao, C.H. Binary Coding SVMs for the Multiclass Problem of Blast Furnace. *IEEE Trans. Ind. Electron.* **2013**, *60*, 3846–3856. [CrossRef]

38. Gao, C.H.; Ge, Q.H.; Jian, L. Rule Extraction from Fuzzy-based Blast furnace SVM Multiclassifier for Decision-Making. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 586–596. [CrossRef]
39. Scardapane, S.; Comminiello, D.; Scarpiniti, M.; Uncini, A. Online Sequential Extreme Learning Machine with Kernels. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2214–2220. [CrossRef] [PubMed]
40. Lim, J.S. Partitioned Online Sequential Extreme Learning Machine for Large Ordered System Modeling. *Neurocomputing* **2013**, *102*, 59–64. [CrossRef]
41. Liang, W.; Reza, L. A Variable Forgetting Factor RLS Algorithm with Application to Fuzzy Time-varying Systems Identification. *Int. J. Syst. Sci.* **1996**, *27*, 205–214.
42. So, C.F.; Ng, S.C.; Leung, S.H. Gradient Based on Variable Forgetting Factor RLS Algorithm. *Signal Process.* **2003**, *83*, 1163–1175. [CrossRef]
43. He, X.; Wang, H.L.; Lu, J.H.; Jiang, W. Weight Ensemble Sequential Extreme Learning Machine with Selection and Compensation and Its Application. *Syst. Eng. Theory Pract.* **2015**, *35*, 2152–2157.
44. Jian, L.; Gao, C.H.; Xia, Z.H. Constructing Multiple Kernel Learning Framework for Blast Furnace Automation. *IEEE Trans. Autom. Sci. Eng.* **2012**, *9*, 763–777. [CrossRef]
45. Jha, R.; Sen, P.K.; Chakraborti, N. Multi-objective Genetic Algorithms and Genetic Programming Models for Minimizing Input Carbon Rates in a Blast FurnaceCompared with a Conventional Analytic Approach. *Steel Res. Int.* **2014**, *85*, 219–232. [CrossRef]
46. Shi, J.; Ding, Z.H.; Lee, W.J.; Yang, Y.P.; Liu, Y.Q.; Zhang, M.M. Hybrid Forecasting Model for Very-short Term Wind Power Forecasting Based on Grey Relational Analysis and Wind Speed Distribution Features. *IEEE Trans. Smart Grid* **2014**, *5*, 521–526. [CrossRef]
47. Jian, L.; Song, Y.Q.; Shen, S.Q.; Wang, Y.; Yin, H.Q. Adaptive Least Squares Support Vector Machine Prediction for Blast Furnace Ironmaking Process. *ISIJ Int.* **2015**, *55*, 845–850. [CrossRef]
48. Wan, C.; Xu, Z.; Pinson, P.; Dong, Z.Y.; Wong, K.P. Probabilistic Forecasting of Wind Power Generation Using Extreme Learning Machine. *IEEE Trans. Power Syst.* **2014**, *29*, 1033–1044. [CrossRef]
49. Gui, G.M.; Li, J.; Zhang, Y.; Li, Z.D.; Ma, X. Prediction Modeling Study for Blast furnace Hot Metal Temperature Based on T-S Fuzzy Neural Network Model. *Iron Steel* **2013**, *48*, 11–15.
50. Martín, R.D.; Obeso, F.; Mochón, J.; Barea, R.; Jiménez, J. Hot Metal Temperature Prediction in Blast Furnace Using Advanced Model Based on Fuzzy Logic Tools. *Ironmak. Steelmak.* **2007**, *34*, 241–247. [CrossRef]