An adaptive least squares support vector machine model with a novel update for NOx emission prediction

You Lv *, Tingting Yang, Jizhen Liu

The State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Changping District, 102206 Beijing, China

ARTICLE INFO

Article history:
Received 1 January 2015
Received in revised form 16 March 2015
Accepted 10 April 2015
Available online 1 May 2015

Keywords:
Data-driven model
Model update
Least squares support vector machine
NOx emissions
Coal-fired boiler

ABSTRACT

This paper presents an adaptive least squares support vector machine (LSSVM) model with a novel update to tackle process variations. The key idea of the update is to divide the process variations into two main categories, namely, irreversible and reversible variations. Correspondingly, sample addition and sample replacement are proposed to update the model. The incremental LSSVM algorithm and detailed update procedure are also provided. A benchmark simulation with a time-varying nonlinear function is conducted to evaluate the effectiveness of the update algorithm. Finally, the proposed method is applied to predict the nitrogen oxide (NOx) emissions of a coal-fired boiler using real operation data from a power plant. Results reveal that the LSSVM model with the novel update maintains high prediction accuracy despite different process characteristics. Meanwhile, the time consumed in the update process is decreased because of the incremental form compared with the model reconstruction.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In industrial processes, certain types of primary variables in industrial processes, such as product qualities and flue gas concentration, should be measured accurately and reliably at all times. These variables may be required to be maintained within specified limitations in accordance with government regulation and manufacturing criteria. Besides, they are also very important in guiding optimal operation [1,2]. However, the accurate measurement of primary variables is hindered by high costs and technical limitations. Although online analyzers are available in a number of plants, these hardware-based instruments are not only highly vulnerable to failure because of being operated under harsh environments but are also expensive and difficult to maintain. Therefore, a new way of realizing a redundant measurement has crucial significance to ensure the safe, economical and efficient operation of plants [3,4].

Soft sensor techniques are widely accepted for estimating primary variables with the use of other relevant variables that are easy to measure online. Soft sensors can operate in parallel with hardware-based sensors, thereby providing a back-up and redundant measurement. Moreover, if the soft sensor model that describes the relationship between the primary and other operating variables is developed, the process operation can also be optimized by regulating the operating parameters [5,6].

Soft sensor models are generally established on the basis of first-principle and data-driven methods. Although often desirable, first-principle models are impractical in most cases because they involve immense complexity and often require several differential equations to be solved mathematically [7]. Meanwhile, data-driven models have wide applications because they can be directly developed based on operation data and detailed fundamental knowledge about the process is unnecessary. Many data-driven modeling methods have been proposed recently, and an extensive review can be found in the work of Kadlec, Gabrys and Strandt [8]. In particular, artificial intelligence techniques such as artificial neural networks (ANN) and support vector machine (SVM) have been widely applied in designing data-driven soft sensors because of their capability to describe highly nonlinear processes in chemical and energy industries. SVM is established based on the structural risk minimization (SRM) principle [9] and exhibits better generalization performance than ANN. An important variation of SVM is the least squares support vector machine (LSSVM), which uses squared errors and equality constraints to substitute inequality ones [10]. Thus, results can be obtained by solving a group of linear equations instead of a quadratic programming problem, thereby significantly reducing the training time of the model.

However, the data source is an important factor to consider when establishing a data-driven model. Several methods have been studied on the basis of experimental data acquired from field experiments [6]. The field experiment can be effectively designed to cover the majority of possible operating conditions, which makes it easier to implement a global approximation [11]. However, all of the concerned parameters
are required to be set to specified values in this approach, and the normal operation and production are disturbed. Therefore, conducting a field experiment is costly and time-consuming. On the contrary, operation data can be easily stored and accessed in real time because of the wide availability of distributed control systems (DCS) and supervisory information systems (SIS). The accumulated historical data represent the real operation status of the industry process, thus, they can provide useful information for developing data-driven soft sensors [12].

Although excellent prediction results have been achieved with the application of soft-sensing techniques, some difficulties are still involved in practical applications. Prediction accuracy often decreases over time because of process drift, plant equipment abrasion, as well as the variations in the external environment and input materials. Thus, the most important problem is maintaining high prediction accuracy at all times after a soft-sensor model is initially established [13–15]. The factors that influence the accuracy of models based on historical operation data can be summarized into two points.

The first point is when the initial model is developed, operation segments with extensive representation should be selected as the training samples to ensure that a large operating region can be covered. Reliable predictions are more likely to be obtained if the current operation condition is within the coverage of the training dataset [16]. However, guaranteeing that all probable working regimes have been incorporated in the initial data set is almost impossible because a number of new working operation conditions could appear at a later time. In other words, collecting adequate representative operation samples is a successive and accumulating process.

The other important point is related to internal changes. For example, irreversible variations such as the abrasion of plant equipment and the alteration of input materials can change the characteristics of the industrial process, thus rendering the initial model incapable to describe the new relationship [17]. ‘Irreversible’ means that the process characteristics would not be recovered once this type of variation occurs.

Therefore, a maintenance and update strategy should be performed on the model. The most straightforward way to reduce the degradation is to reconstruct a new model periodically. The reconstruction can be implemented based on time difference of the variables, or using the data that are recently measured or have high similarities with the prediction [18]. However, in such an approach, previous training results would be abandoned, thus resulting in a heavily repeated computation. Moreover, reconstructed models are inclined to specialize in a narrow operation range, and they cannot predict the process variations accurately [19]. The model has to be trained again to be adequate for the former process if the previous operation conditions appear.

To tackle this issue, several update approaches have been presented, including the moving window method [20,21], recursive method [22], and their variants. Numerous studies have utilized the abovementioned frameworks to update data-driven soft sensor models. For example, Qin [23] proposed a recursive-PLS algorithm to tackle model degradation. Kaneko, Arakawa and Funatsu [24] presented a moving window method that uses recently acquired data to update the model. Online SVM and LSSVM models with time variables were also proposed to improve the prediction accuracy when the process changed [25–27]. In addition, a number of studies focused on the aspects of maintenance-free learning [28] and local learning, such as just-in-time learning, instance-based learning, and lazy learning [29,30], in which local models were established adaptively using relevant data samples to describe the current process characteristics. Further, Kaneko and Funatsu [31] classified the degradation of the models into three types and discussed different update effects based on the classification results. Besides that, they proposed an index to manage the database considering the variation, based on which the prediction accuracy of adaptive soft sensor models increased [19]. In addition, Kadlec, Grbić and Gabrys [32] and Šlišković, Grbić and Hocenski [33] presented comprehensive reviews on different types of update techniques.

The conventional update framework is implemented by prioritizing samples that are newly measured or similar to the prediction data to solve the model degradation when the process characteristics vary [27]. However, a detailed analysis of the fundamental causes of the process variations can be helpful in the implementation of the update measures, which is seldom discussed in previous studies. This paper presents an adaptive LSSVM with a novel update based on the analysis of process variations, which are categorized into irreversible and reversible variations according to the similarity criterion. Addition and replacement strategies are used on the samples to update the LSSVM model incrementally according to the variation types. This method is then assessed with a benchmark nonlinear function and applied to predict the nitrogen oxide (NOx) emissions of a coal-fired boiler. Finally, the performance of the proposed model is compared with that of the original LSSVM without updates.

The paper is organized as follows. The process variations are analyzed and the update scheme is provided in Section 2. Section 3 introduces the incremental LSSVM. Section 4 presents the detailed update procedure based on the LSSVM model. A benchmark simulation is conducted in Section 5, and a real industrial application is described in Section 6. Section 7 presents the conclusions.

2. Process variations and model update

2.1. Process variation characteristics

Based on the changes in the explanatory and primary variables, and the rapidity of the changes, Kaneko and Funatsu [31] made a systematic study on the degradation of the models. Different from that, in this paper, the process variation characteristics are classified into two categories according to actual process characteristics and the variation causes. The first is irreversible variation, which is caused by internal factors such as changes in fuel property, abrasion and maintenance of plant equipment. This type of variation often occurs irreversibly because the characteristics of fuel and equipment cannot be reverted back to the previous condition. The other category is reversible variation, which is caused by external factors such as production load changes and valve opening alteration. Unlike the case with irreversible variations, the operation characteristics with reversible variations could return to the previous condition as the process proceeds.

A specific example of a single-in-single-out system with process variations is illustrated in Fig. 1. The initial operation condition is denoted as I in Fig. 1(a). The soft-sensing model \( y = f(x) \ (x \in [x_1, x_2]) \) is established based on the representative training samples selected from the historical operation database. New operation instructions are received with the changes in the production loads. Accordingly, the operating variable \( x \) is set to a new value, and the process condition shifts to II \( (x \in [x_3, x_4]) \) and III \( (x \in [x_5, x_6]) \). Poor accuracy is obtained if the initial model \( y = f(x) \ (x \in [x_1, x_2]) \) is still applied to predict the process characteristics. Notably, the abovementioned operation variation is reversible because the condition could possibly return to the initial area I with certain changes in the operation instructions. Therefore, the new samples collected in conditions II and III should be incorporated with condition I to update the initial model. The operation space is then expanded from \( x \in [x_1, x_2] \) to \( x \in [x_1, x_6] \), which is shown in Fig. 1(b).

Fig. 1(c) illustrates an irreversible variation from condition I to IV, which can be primarily attributed to the abrasion and maintenance of plant equipment. Different from the previous variation, the process characteristics could no longer change back. Therefore, the old samples representing former characteristics are already of little use and should be replaced by the newly-collected ones. On the basis of the above analysis, the final model should be adequate to describe the new process characteristics shown in Fig. 1(d) after the model update.
2.2. Solution measures

In consideration of the reversible and irreversible process variations, two types of measures are proposed to update the model, namely, (I) merging new samples into old ones and (II) replacing old samples with new ones. For reversible variations, the new samples should be directly added to the previous dataset; for irreversible variations, the new samples should be used to replace the corresponding unnecessary ones in the previous dataset. Moreover, the two update measures can be implemented by two types of procedures, namely, exclusion of old samples and addition of new samples. The first update measure can be accomplished by directly adding newly-collected samples, whereas the other measure can be achieved by first removing the unnecessary samples and then adding the new ones.

3. Incremental LSSVM algorithm

To maximize the former training results and at the same time reduce the time consumed in the model update, the incremental learning method can be integrated with the two-measure update framework and improve the model performance adaptively. The incremental LSSVM is detailed in this section.

3.1. LSSVM

Considering the training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ that consist of $n$ samples with $p$ input variables $\mathbf{x}_i \in \mathbb{R}^p$ and response variable $y_i \in \mathbb{R}$, $i = 1, 2, \ldots, n$, the LSSVM regression model can be provided as follows [34]:

$$y_i = \mathbf{w}^T \varphi(\mathbf{x}_i) + b \quad (1)$$

where $\varphi(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^q$ is a nonlinear function mapping input variables from the original space $\mathbb{R}^p$ into a high feature space $\mathbb{R}^q$, $\mathbf{w}$ is a weight vector, and $b$ is a bias.

The squared errors are used as the cost function according to the SRM principle. The LSSVM model can then be developed by using the following formulation:

$$\min_{\mathbf{w}, b, \xi} J(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{i=1}^n \xi_i^2$$

s.t. $y_i = \mathbf{w}^T \varphi(\mathbf{x}_i) + b + \xi_i, \quad i = 1, \ldots, n \quad (2)$

where $\gamma$ is a penalty parameter that balances model complexity and approximation accuracy, $\xi_i$ is the $i$th error variable, and $\xi = [\xi_1, \ldots, \xi_n]^T$. Then, by using the Lagrangian method, the optimization problem can be converted into a group of linear equations:

$$L(\mathbf{w}, b, \xi, \alpha) = J(\mathbf{w}, \xi) + \sum_{i=1}^n \alpha_i [\mathbf{w}^T \varphi(\mathbf{x}_i) + b + \xi_i - y_i] \quad (3)$$

where $\alpha = [\alpha_1, \ldots, \alpha_n]^T$ is the Lagrangian multiplier vector. The solutions for optimality are obtained based on the Karush–Kuhn–Tucker conditions:

$$\begin{align*}
\frac{\partial L}{\partial \mathbf{w}} &= 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i \varphi(\mathbf{x}_i) \\
\frac{\partial L}{\partial b} &= 0 \Rightarrow \sum_{i=1}^n \alpha_i = 0 \\
\frac{\partial L}{\partial \xi_i} &= 0 \Rightarrow \alpha_i = \gamma \xi_i \\
\frac{\partial L}{\partial \alpha_i} &= 0 \Rightarrow \mathbf{w}^T \varphi(\mathbf{x}_i) + b + \xi_i + y_i = 0
\end{align*} \quad (4)$$

The solutions can be given in a group of linear equations by eliminating the variables $\mathbf{w}$ and $\xi_i$:

$$\begin{bmatrix} 0 & 1^T \\ \Omega + \frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} \alpha \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (5)$$

Fig. 1. Process variation characteristics and model update: (a) operation expansion; (b) the model after expansion; (c) operation change; and (d) the final model considering the operation expansion and change.
where \( \mathbf{1} = [1, \ldots, 1]^T, \mathbf{y} = [y_1, \ldots, y_n]^T, \mathbf{I} \), is an \( n \times n \) element matrix, \( \Omega = \{(\Omega_{ij})_{ij} = 1, \ldots, n\}, \) and \( \Omega_{ij} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \) is the kernel function. The radial basis function (RBF) is used in this paper, and the expression is detailed as follows:

\[
K(x, x_i) = \exp(-\|x-x_i\|^2/\sigma^2) 
\]  

(6)

where \( \sigma \) is the kernel parameter.

The parameters \( \mathbf{a} \) and \( \mathbf{b} \) can be calculated by solving Eq. (5) as follows:

\[
\begin{aligned}
\mathbf{b} &= \frac{1}{\mathbf{H}^T \mathbf{y}} \\
\mathbf{a} &= \mathbf{H}^{-1} (\mathbf{y} - \mathbf{1} \mathbf{b})
\end{aligned} 
\]  

(7)

where \( \mathbf{H} = \mathbf{I} + \frac{1}{\gamma} \mathbf{K} \) and \( \mathbf{K} \) is a positive definite matrix. Finally, the LSSVM regression model can be obtained by substituting Eq. (7) into Eq. (1) as follows:

\[
y(x) = \mathbf{w}^T \phi(x) + b = \sum_{i=1}^{n} a_i K(x, x_i) + b
\]  

(8)

### 3.2. Exclusion of the old samples

The task of the model update can be achieved by two procedures, namely, exclusion of the old samples and addition of the new ones. In addition, the model should accordingly be trained again after data samples are altered. The most time-consuming part of the training process is calculating \( \mathbf{H}^{-1} \) in Eq. (7). To maximize the previously attained training information, incremental LSSVM method is deployed to execute the model update [35,36].

Considering the current training samples \( \{(x_i, y_i)\}_{i=1}^{n} \), with matrix \( \mathbf{H} \), suppose that the \( s \)th sample \( (x_s, y_s) \) is to be removed. First, the \( s \)th row and column are switched with the \( n \)th row and column, respectively. The matrix \( \mathbf{H} \) can then be transformed to \( \mathbf{H}_0 \) as follows:

\[
\begin{bmatrix}
K(x_1, x_1) + \frac{1}{\gamma} & \cdots & K(x_1, x_n) \\
\vdots & \ddots & \vdots \\
K(x_n, x_1) & \cdots & K(x_n, x_n) + \frac{1}{\gamma}
\end{bmatrix}
\]  

\[
\begin{bmatrix}
H_0^{-1} \\
\mathbf{k}_s \\
k
\end{bmatrix}
\]  

(9)

where \( \mathbf{k}_s = (K(x_s, x_1), \ldots, K(x_s, x_n), K(x_1, x_s), \ldots, K(x_n, x_s)) \) and \( k = K(x_s, x_s) + \frac{1}{\gamma} \).

The elementary matrices are denoted as \( \mathbf{T}_{n \rightarrow n-1} \) and \( \mathbf{T}_{n-1 \rightarrow n} \), which correspond to the switching transformations. \( \mathbf{T}_{n \rightarrow n-1} \) is identical to \( \mathbf{T}_{n-1 \rightarrow n} \) and uniformly denoted as \( \mathbf{T}_{n \rightarrow n} \). Thus, the following equation can be obtained:

\[
\mathbf{H}_0 = \mathbf{T}_{n \rightarrow n} \mathbf{H} \mathbf{T}_{n \rightarrow n}
\]  

(10)

The inverse of \( \mathbf{H}_0 \) can also be provided:

\[
\mathbf{H}_0^{-1} = \mathbf{T}_{n \rightarrow n} \mathbf{H}^{-1} \mathbf{T}_{n \rightarrow n} = \mathbf{T}_{n \rightarrow n} \mathbf{H}^{-1} \mathbf{T}_{n \rightarrow n}
\]  

(11)

If the inverse of \( \mathbf{H}_0 \) is denoted as follows:

\[
\mathbf{H}_0^{-1} = \begin{bmatrix}
\mathbf{h}_{11} & \mathbf{h}_{12} \\
\mathbf{h}_{12}^T & \mathbf{h}_{22}
\end{bmatrix}
\]  

(12)

then the matrix \( \mathbf{H}_0 \) can be rewritten in the following form based on block-wise inversion theory:

\[
\mathbf{H}_0^{-1} = \begin{bmatrix}
(h_{11} - h_{12} h_{22}^{-1} h_{12}^T)^{-1} & -h_{11}^{-1} h_{12} (h_{22}^{-1} h_{12}^T h_{11}^{-1} h_{12})^{-1} \\
-h_{12}^{-1} h_{11} (h_{11}^{-1} h_{12} h_{22}^{-1} h_{12}^T)^{-1} & (h_{22}^{-1} h_{12} h_{11}^{-1} h_{12}^T)^{-1}
\end{bmatrix}
\]  

(13)

The following equation can be obtained by comparing Eqs. (9) and (13):

\[
\mathbf{H}_0^{-1} = h_{11} - h_{12} h_{22}^{-1} h_{12}^T.
\]  

(14)

\( \mathbf{H}_0^{-1} \) can then be calculated using Eq. (11) because the current matrix \( \mathbf{H}^{-1} \) is known, and \( h_{11}, h_{12}, \) and \( h_{22} \) can be determined accordingly. The inverse of the new matrix \( \mathbf{H}_1 \) can then be obtained according to Eq. (14). Finally, the new model parameters can be calculated based on Eq. (7). Thus, the calculation of \( \mathbf{H}_1^{-1} \) is significantly simplified.

At this point, only a single sample \( (x_s, y_s) \) is considered in the above analysis. If there are multiple samples to tackle, they can be removed one by one through the abovementioned steps.

### 3.3. Addition of the new samples

Suppose that the new sample to be added is \( (x, y) \) and the current matrix is also denoted as \( \mathbf{H} \); then, the new matrix can be written as follows:

\[
\begin{bmatrix}
K(x_1, x_1) + \frac{1}{\gamma} & \cdots & K(x_1, x_n) \\
\vdots & \ddots & \vdots \\
K(x_n, x_1) & \cdots & K(x_n, x_n) + \frac{1}{\gamma}
\end{bmatrix}
\]  

\[
\begin{bmatrix}
\mathbf{h}_1 \\
\mathbf{k}_t \\
k
\end{bmatrix}
\]  

(15)

where \( \mathbf{k}_t = (K(x_t, x_1), \ldots, K(x_t, x_n), K(x_1, x_t), \ldots, K(x_n, x_t)) \) and \( k = K(x_t, x_t) + \frac{1}{\gamma} \). Based on the Sherman–Woodbury–Morrison formula, the inverse of \( \mathbf{H}_2 \) can be rewritten in another form:

\[
\mathbf{H}_2^{-1} = \begin{bmatrix}
\mathbf{H}^{-1} + \mathbf{H}^{-1} \mathbf{k}_t \rho^{-1} \mathbf{k}_t^T \mathbf{H}^{-1} & -\mathbf{H}^{-1} \mathbf{k}_t \rho^{-1} \\
-\mathbf{k}_t^T \mathbf{H}^{-1} & \rho^{-1}
\end{bmatrix}
\]  

(16)

where \( \rho = k - \mathbf{k}_t^T \mathbf{H}^{-1} \mathbf{k}_t \).

Subsequently, the inverse of \( \mathbf{H}_2 \) can be obtained by using the matrix \( \mathbf{H}^{-1} \). The new values of parameters \( \mathbf{a} \) and \( \mathbf{b} \) can be calculated by substituting Eq. (16) into Eq. (7). Similarly, only sample \( (x, y) \) is concerned and multiple samples can be added into the dataset one by one.

### 4. Update procedure

#### 4.1. Strategy selection criterion

Given that two types of process variations should be considered when applying the update strategy, the first step is to determine
which type of variations has occurred. That is, the characteristic of the newly-collected data sample should be detected to determine which update category it belongs to. If the sample represents an irreversible variation, the corresponding old samples should be displaced by this new sample. If it represents a reversible variation (i.e., new operation conditions emerge), this sample should be added to the previous training dataset.

In this study, variations are identified in terms of the similarity between the current sample and corresponding ones in the old training dataset. As is shown in Fig. 2, for the newly-acquired sample \((x_q, y_q)\), the most relevant sample in the old training data set is denoted as \((x_0, y_0)\). If the sample \((x_q, y_q)\) belongs to area (a), the prediction accuracy is good because the corresponding similar samples \((x_0, y_0)\) have been incorporated in the training dataset. If the prediction error of the sample \((x_q, y_q)\) exceeds the given threshold which means the model needs to be updated, this sample is considered to belong to areas (b) and (c). If the sample \((x_q, y_q)\) belongs to area (b), \([i.e., (x_{q1}, y_{q1})]\), a significant change in the condition point \(x ≈ x_{q1}\) is assumed because the predicted value \(y\) is identified to be different from the old one. In this case, process variation is assumed to be reversible. This sample will be merged into the data set, and update strategy (I) is performed.

If the performance deterioration of the model exceeds the limit on the new sample \((x_q, y_q)\), the update is implemented based on the following rules:

1. If \( ||x_k - x_q||_2 > \delta_1 \), update strategy (I) is performed, and the new sample \((x_q, y_q)\) is added into the training set.
2. If \( ||x_k - x_q||_2 ≤ \delta_1 \), update strategy (II) is performed, and the relevant old samples \((x_i, y_i)\) that satisfy the criterion \( ||x_k - x_i||_2 ≤ \delta_2 \) are replaced by the new sample \((x_q, y_q)\).

Here, \(\delta_1\) is determined by the mean values of the distance between each sample in the training dataset, and \(\delta_2\) is set to 0.5\(\delta_1\).

The parameter \(\delta_1\) affects the type of the update strategy. If \(\delta_1\) is small, the update strategy (I) is inclined to be carried out, otherwise, the update strategy (II) is inclined to be carried out. The parameter \(\delta_2\) affects the number of samples that need to be replaced during the update. If \(\delta_2\) is larger, more old samples will be replaced to make the performance improved. The two parameters can also be fine-tuned properly according to the specific training dataset and update effect.

### 4.2. Update time sequence

The two main forms of time sequences in the model update are sample wise and batch wise [32]. In sample-wise form, the model is updated immediately when the latest sample becomes available [21]. In batch-wise form, the model is updated after a group of data samples has been collected [37]. Theoretically, the sample-wise update can more likely capture any slight changes in the process, however, excessive frequent updates are unnecessary and also increase the computation time.

---

Fig. 2. Criterion to detect the variation type.

Fig. 3. Time sequence of model update.
load. Moreover, the operation state of the new data cannot be distinguished only by one sample because no information is available on the subsequent samples. Therefore, whether the current sample is appropriate for model updates is difficult to determine. On the contrary, it is beneficial to select useful samples to update the model if a section of consecutive data samples is available. The concept of a buffer block is then presented, where two types of update strategies are integrated together.

The buffer block, in which the latest data samples are temporarily stored, is constructed based on the block-wise framework. Model performance is evaluated after a block of data samples are fully collected, and then the update strategy is triggered if necessary. In particular, the new block of samples is first preprocessed and useful data samples are picked out. The selected samples are then used to update the LSSVM model in sample-wise form with the use of the aforementioned incremental algorithm. The timing sequence of the model update is illustrated in Fig. 3. The performance detection and update strategy of the model are invoked when the buffer block is full. Simultaneously, the newly-acquired sample is stored into the next block. When the update is finished, the next buffer is not filled because the time consumption in the update is short. The update strategy is executed again after the next buffer block of samples becomes available.

It can be concluded that the main task of the buffer block is to store enough data samples, thus making it convenient to select appropriate samples to update the model. The detailed update procedure remains in sample-wise form to decrease computing complexity.

The entire modeling and update steps are detailed as follows:

S1: Establish the model using the initial training data set and consider it as the current soft sensor;
S2: Construct a new buffer block;
S3: Acquire the new sample and store it into the current buffer block;
S4: Check whether the buffer block is full, if yes go to S5, and simultaneously, next new buffer block is created to store newly-acquired samples, else return to execute S3;
S5: Calculate the prediction error;
S6: Check the model performance and if the error exceeds the limit, go to S7, else go to S3;
S7: Update the model;
S7.1: Preprocess the data block and get appropriate samples to update the model;
S7.2: Conduct the incremental update using the aforementioned strategy;
S7.3: Complete the update and return S3 to check the next buffer block.

It should be noticed that the subsequent data sample is simultaneously stored into the next new buffer block when executing S5. In addition, the time consumed in executing S5 to S7 is short, and the update is definitely accomplished before the next buffer block is filled up. The entire procedure is illustrated in Fig. 4.

5. Case simulation

5.1. Benchmark function

In this experiment, the typical nonlinear sinc function is simulated as a benchmark to validate the update strategy proposed in this paper. The expression of the sinc function is defined as follows:

$$\text{sinc}(x) = \frac{\sin(x)}{x}. \tag{18}$$

Fig. 5 shows the training and testing datasets, which are generated based on the following rules to illustrate the adaptability to process variations.

5.1.1. Initial training dataset

A total of 100 data samples are generated uniformly in the range of $[-8, 2]$ based on the original relationship $\text{sinc}(x)$, which constitute the training dataset that establishes the initial model.

5.1.2. Testing dataset

The testing dataset mainly refers to the newly acquired samples after the model is established. The testing data samples are generated based on three rules to represent different operation conditions:

(i) The first 20 testing samples are generated uniformly in the range of $[2, 6]$ that represent new operation conditions;
(ii) The second 20 testing samples are generated uniformly in the range $[-2.2, 1.8]$ that represent former historical operation conditions;
(iii) The last 20 testing samples are generated uniformly in the range $[-5, -2.5]$ based on the new relationship of $y = \sin(0.9x)/x$ [denoted as sinc (0.9x) for short] that represent the varied process characteristics.

All three types of testing data are used to investigate the adaptability of the model with the update strategy. The training data are considered as known samples to develop the initial model. The testing data samples
are provided one by one to simulate the real industrial collection process. The root mean square error (RMSE) and mean relative error (MRE) criteria are used to evaluate prediction accuracy and model performance, which are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$  \hspace{1cm} (19)

$$\text{MRE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$  \hspace{1cm} (20)

where \(y\) and \(\hat{y}\) are the actual and predicted values, and \(N\) is the number of data samples.

The LSSVM model is developed based on the initial 100 training samples, and the RBF is used as the kernel function. The kernel parameter \(\sigma\) and the penalty parameter \(\gamma\) both have a considerable influence on model performance and should be determined beforehand. Cross-validation is a common and effective method. The two parameters are determined by a 5-fold cross-validation using grid search in the hyper-space range of \([2^{-2}, 2^{0}] \times [2^{-1}, 2^{15}]\). In addition, previous studies have shown that identifying a good parameter pair by searching exponentially in the sequences is more practical and less time-consuming, i.e., \(\sigma = 2^{-2}, 2^{-1}, \ldots, 2^6; \gamma = 2^{-1}, 2^0, \ldots, 2^{15}\) [38,39]. The final values of the two parameters are \(\sigma = 0.82\), and \(\gamma = 982\). The testing data are then predicted based on the LSSVM model with the proposed update method. The buffer length is tentatively set to 1 (i.e., the update is implemented immediately after a data sample is captured). The prediction results using the traditional LSSVM model without updates are also provided for comparison.

5.2. Results and discussion

In fact, the LSSVM model has been proved to have a good approximation capability on the training samples, which is beyond the scope of this paper. Instead, the testing performance is focused on. The simulation results on the testing samples are shown in Fig. 6(a) and (b). Fig. 6(a) is \(y-x\) relation, whereas Fig. 6(b) is shown in a sample-sequence form. The models with and without updates are denoted by LSSVMUP and LSSVM, respectively.

As is shown in Fig. 6, high prediction accuracy is achieved on the testing samples that belong to the historical operation range \(x \in [-2.2, 1.8]\) using the LSSVM model without updates. However, the prediction errors of the testing samples in the ranges of \([2, 6]\) and \([-5, -2.5]\) increase, indicating that the model performance is degraded when new operation conditions occur or when the process characteristics vary. The RMSE and MRE of the testing samples reach 0.087 and 0.69, respectively.

By contrast, the prediction accuracy of LSSVMUP remains high on all the testing samples. The prediction errors of RMSE and MRE are only 0.0083 and 0.15, which are much smaller than the model without updates. These values suggest that the update strategy is significantly meaningful and effective for improving the model performance when the process characteristics vary.

The detailed update process is illustrated in Fig. 7. The implemented update strategy is represented by the variable Flag. Flag is denoted with 1 to indicate that new data samples are merged into old ones. Flag is set to 2 to show that new data samples are used to replace the relevant old ones. Flag is set to 0 to represent that no updates are executed. Fig. 7 shows that both update strategies are implemented with new testing samples collected.

The discussion above is focused on the assumption that the buffer length is 1. However, as previously mentioned, such a situation is difficult to achieve in a real industrial process. Fig. 8 presents the simulation results when the buffer length is set to 3. The RMSE and MRE values are 0.02 and 0.19, respectively. The prediction accuracy worsens compared with the previous instance with buffer length set to 1. Nevertheless, the prediction errors are still much lower than the model without updates.

6. Application to an industrial power plant

NOx emissions from coal-fired boilers have become a worldwide concern with the increasing requirements for environmental protection [40]. At present, a continuous emission monitoring system (CEMS) is used to realize the online measurement on NOx emissions in power plants. However, CEMS often requires offline maintenance because
of harsh working environments. A soft sensor model can work in parallel with CEMS and provide a redundant measurement on NOx emissions. Moreover, NOx emissions can be lowered by optimal combustion if the relationship between other operating parameters is obtained [38,39]. The soft sensor is applied in this section to measure the NOx emissions of a 660 MW coal-fired boiler.

6.1. Plant description

The investigated boiler has a large furnace with a 19.08 m × 19.08 m cross-section and a 65.1 m height. Six medium-speed pulverizers are used to supply pulverized coal, which is then mixed with the primary air and blown into the furnace through the burner nozzles. An imaginary horizontal circle with a 7.69 m diameter is then formed in the...
of the furnace and burner layout is shown in Fig. 9. Six layers of primary air (A, B, C, D, E, and F) and eight layers of secondary air (AA, AB, BC, CC, DD, DE, EF, and FF) are distributed alternatively in a vertical direction. Six layers of circumferential air (A–F) are arranged surrounding the burner nozzles to supply sufficient air and guarantee the combustion stability. Four layers of over fire air (OFA) are installed over the upper nozzles to replenish the air in the later combustion phase for better combustion efficiency. The schematic diagram of the furnace and burner layout is shown in Fig. 9.

6.2. Input variable selection

The formation mechanism and operation experience indicate that NOx emissions have a close relationship with the following factors: boiler load, total fuel rate, total air rate, coal-feed rate, primary air rate, secondary air rate, circumferential air rate, OFA air rate, and O2 concentration in the flue gas. Theoretically, all of the related parameters should be taken as the model inputs, and the ideal model structure is illustrated in Fig. 10(a). However, it is obviously impractical because of two reasons:

1. After analyzing the operation data, a strong correlation exists among the 18 variables: coal-feed rates (A–F), primary air rates (A–F), and circumferential air rates (A–F). The correlation brings redundant information to the inputs, thereby increasing model complexity and ultimately degrading the model performance. Principal component analysis (PCA) is then used to eliminate the correlation, and the variance extraction is shown in Fig. 11. It is shown that 6 components extracted using PCA are adequate to represent the information of all of the variables (96.63% variance). Similarly, a correlation relationship also exists among the variables of load, total fuel rate, and total air rate, as shown in Table 1. Unlike the previous one, the correlation coefficient of each variable is close to 1, and the boiler load can sufficiently describe the information of all three variables. Therefore, only the boiler load is taken as the model input without considering the variables of the total fuel and total air rates.

2. Coal quality is an important factor that affects NOx formation. However, the coal has been mixed well according to the design requirement before it is burned in the furnace under normal operation conditions, which means that the coal quality maintains roughly unchanged. Besides, there is no real-time data about the coal quality because the on-line analyzer is unavailable in the power plant. Coal quality can also be reflected by the fuel and air rates. Therefore, coal quality is not included in the input variables.

Based on the two above points, the following parameters are taken as the model inputs: boiler load, 6 PCA components extracted from the 18 variables of primary air rates (A–F), circumferential air rates (A–F) and coal-feed rates (A–F), 8 layers of secondary air rates (AA–FF), 4 layers of OFA air rates (UA–UD), and O2 concentration in the flue gas. The NOx emission is taken as the model output. The practical model structure is given in Fig. 10(b). All data are normalized into \([-1, 1]\), and the result is transformed back after prediction.

6.3. Model construction

Several sections of historical operation data for 15 days are acquired from the DCS database with a resolution of 1 min, and approximately 12,700 data samples are obtained. Given noise interferences and sensor failures, abnormal points or outliers exist in the raw DCS database. Therefore, data preprocessing is conducted to eliminate certain unnecessary samples before utilizing the raw data samples. In addition, unsteady-state data samples are also removed in advance because these samples exhibit different characteristics from regular operation conditions, and there is little corresponding relation between the input and the response variables [28,41]. These samples can also be regarded as a reflection of variations caused by rapid changes of operating parameters. It is difficult to construct an accurate model for a transient situation when sufficient data is not available. The operation state can be distinguished by the change rate and variance of the samples, and the detailed description can be found in the related work [17,39]. Finally, 1780 data samples are selected from the raw data.

![Fig. 11. Variance extraction based on PCA.](image)

**Table 1**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Load</th>
<th>Fuel</th>
<th>Air</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boiler load</td>
<td>1</td>
<td>0.9848</td>
<td>0.9732</td>
</tr>
<tr>
<td>Total fuel rate</td>
<td>0.9848</td>
<td>1</td>
<td>0.9773</td>
</tr>
<tr>
<td>Total air rate</td>
<td>0.9732</td>
<td>0.9773</td>
<td>1</td>
</tr>
</tbody>
</table>

![Graph showing comparisons of prediction errors on testing samples between LSSVM and LSSVMUP.](image)

**Table 2**

Comparisons of prediction errors on testing samples between LSSVM and LSSVMUP.

<table>
<thead>
<tr>
<th>Models</th>
<th>RMSE (mg m⁻³)</th>
<th>MRE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–180</td>
<td>181–680</td>
<td>All samples</td>
</tr>
<tr>
<td>LSSVM</td>
<td>16.87</td>
<td>38.26</td>
</tr>
<tr>
<td>LSSVMUP</td>
<td>15.24</td>
<td>15.13</td>
</tr>
</tbody>
</table>

![Graph showing prediction results on training data set.](image)
Moreover, the training data should cover a large span of boiler load to provide extensive representation to the initial model. Typical operation segments with 1100 samples for the early 6 days, during which the boiler load varies from 300 MW to 660 MW, are selected as the training dataset to construct the initial model. The remaining segments with 680 samples are taken as the testing dataset to investigate the model generalization capability and update strategy. Similarly, all the training samples are provided initially, whereas the testing samples are obtained one by one.

As mentioned above, the kernel function and penalty parameters are searched exponentially based on the 10-fold cross-validation in the range of \([2^{-2}, 2^{5}] \times [2^{-1}, 2^{15}]\), and the obtained parameters are \(\sigma = 3.01\) and \(\gamma = 283.4\). RMSE and MRE are still used to evaluate the model performance. The original LSSVM model without updates and LSSVMUP are both established to verify the effect of the update measures.

### 6.4. Results and discussion

As expected, the prediction effect on the training dataset (1–1100 samples) are accurate with RMSE = 3.24 and MRE = 0.83%. The detailed results are shown in Fig. 12. The prediction results on the testing dataset are the primary focus of this paper.

The prediction errors of the two models on the testing data set are shown in Table 2. The prediction errors of the LSSVM model on the first 180 testing samples are 16.87 mg m\(^{-3}\) and 4.38%, which implies that the model has good prediction accuracy in the beginning. However, it is gradually deteriorated with the operation proceeding and the prediction errors increase quickly. The RMSE and MRE errors on samples 181 to 680 reach as high as 38.26 mg m\(^{-3}\) and 11.57%, respectively. By contrast, the LSSVMUP model exhibits much better estimation accuracy on the testing data set, and the RMSE and MRE errors on the entire testing samples remain around 15.1 mg m\(^{-3}\) and 3.3%, which are much smaller than the LSSVM model without updates. The results indicate that the update measures are conducive for improving the model performance.

![Fig. 13. Prediction results of testing data set.](image)

![Fig. 14. Simulation time distribution in the model update progress.](image)

The actual measured values and prediction results of the two models are compared in Fig. 13. The prediction accuracy of LSSVM on new data samples remains high for a short time after the initial model training. However, the deviation begins to increase from the 181st sample. Soon afterwards, the error becomes increasingly larger with new samples collected. In comparison, the prediction accuracy of the LSSVMUP model with the update maintains high all the time even after the process characteristics vary. It is concluded that the model update is significantly meaningful and effective in enhancing the prediction precision.

The time consumed in the update progress is shown in Fig. 14. Due to the incremental method used in the update procedure, accomplishing the computation in every update takes no more than 0.5 s. However, the time spent in model reconstruction is approximately 400 s \([39]\). The calculation time is significantly decreased. In addition, the time interval between two data samples is 60 s, which provides sufficient time to complete the update before the buffer block is filled up.

In addition, the update with buffer length setting to 1 (denoted as LSSVMUP\(^*\)) is also constructed, in which the model is updated immediately after a new data sample is captured. The model performance between LSSVMUP and LSSVMUP\(^*\) on the prediction of testing samples is compared, and the results are shown in Table 3.

<table>
<thead>
<tr>
<th>Models</th>
<th>RMSE (mg m(^{-3}))</th>
<th>MRE (%)</th>
<th>Update number</th>
<th>Update time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSSVMUP</td>
<td>15.15</td>
<td>3.31</td>
<td>13</td>
<td>4.8</td>
</tr>
<tr>
<td>LSSVMUP(^*)</td>
<td>13.44</td>
<td>3.10</td>
<td>32</td>
<td>15.8</td>
</tr>
</tbody>
</table>

It can be found the prediction accuracy of LSSVMUP has almost the same accuracy as LSSVMUP\(^*\). The prediction errors of the LSSVMUP are a little larger compared to the LSSVMUP\(^*\) model with an immediate update. However, the computation time and update number of the LSSVMUP are decreased a lot. The results show that excessive frequent updates are not necessary, and the buffer block is helpful to decrease the number of updates while maintaining the prediction accuracy.

### 7. Conclusions

A novel update method is proposed in this paper to establish an adaptive LSSVM model for tackling the time-varying characteristics of real industrial processes. First, the intrinsic reasons that lead to process variations are analyzed, based on which the variations are classified into two main categories. Accordingly, two types of measures are proposed to update the initial LSSVM model. The incremental framework is used to reduce the time consumed in computation when executing the update. The update sequence is also detailed. A benchmark function is designed to prove the effectiveness of the update strategy. The simulation results show that the LSSVMUP model exhibits better performance compared with the original LSSVM model without updates in terms of decreasing the prediction errors on testing samples. The method is then applied to predict the NOx emissions of a coal-fired boiler using real operation data. The RMSE and MRE errors on the newly acquired testing samples are only 15.1 mg m\(^{-3}\) and 3.3% respectively, which are much smaller than the LSSVM model without updates. The comparison results reveal that the model update maintains the prediction accuracy at a high level even if the process characteristics vary. Besides, updating the model at a time takes no more than 0.5 s due to the incremental algorithm. Lastly, it should be mentioned that the update framework proposed in this paper is not confined to the LSSVM model. It can also be suitable for other data-driven techniques if the update time is acceptable.
Conflict of interest

There is no conflict of interest.

Acknowledgments

This project was funded by the National Basic Research Program of China (‘973’ Project) (No. 2012CB215203) and the Fundamental Research Funds for the Central Universities.

References