

# Local Parameter Optimization of LSSVM for Industrial Soft Sensing With Big Data and Cloud Implementation

Xinyu Zhang and Zhiqiang Ge<sup>✉</sup>, Senior Member, IEEE

**Abstract**—Due to the advantages of high prediction accuracy, least squares support vector machine (LSSVM) has been widely utilized for soft sensor developments in industrial processes. The hyper-parameters of LSSVM are often determined by minimizing the predicted error of validation set based on the intelligent optimization algorithm, which may lead to excessive optimization and model overfitting when validation set are selected improperly. Meanwhile, online parameters optimization is difficult to implement, which results in poor effect of local modeling. This paper proposes UMDA-LOS-LSSVM that is a LSSVM with parameters optimization in local objective set (LOS-LSSVM) by univariate marginal distribution algorithm (UMDA) based on the idea of local modeling. First, the local objective set is extracted in the candidate set based on the testing samples. Then, UMDA is utilized for minimize the predicted error of the objective set and provides the optimized parameters. Finally, training and testing of LSSVM are carried out based on the optimal parameters. In addition, this paper provides the distributed parallel form of the proposed method, which can be used for big data modeling and soft sensor development. The proposed method is applied in a CO<sub>2</sub> absorbing column unit to estimate the residual CO<sub>2</sub> content, which is implemented through an industrial big data distributed analytics platform. The results show a significant improvement of proposed method based soft sensor, compared to traditional methods.

**Index Terms**—Cloud computing, distributed parallel, estimation of distribution algorithm (EDA), local objective set, least squares support vector machine (LSSVM).

## I. INTRODUCTION

IN MODERN industrial processes, many key process variables and quality-related variables need to be measured accurately, which can improve the production process [1]. However, due to the limitation of actual measuring instruments,

Manuscript received November 27, 2018; accepted February 13, 2019. Date of publication February 20, 2019; date of current version February 6, 2020. This work was supported in part by the National Natural Science Foundation of China (NSFC) (61833014, 61722310, 61673337), the Natural Science Foundation of Zhejiang Province (LR18F030001), and the Fundamental Research Funds for the Central Universities under grant 2018XZZX002-09. Paper no. TII-18-3153. (*Corresponding author: Zhiqiang Ge*)

The authors are with the Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: zhangxinyu1995@126.com; gezhiqiang@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2019.2900479

many pivotal variables can hardly be obtained directly, such as the quality of product, the content of process gas, and some melt indices [2]. Therefore, data-driven soft sensors emerge as the times require and have been developed and implemented for several decades [3]. The principle of soft sensors can be roughly described that it utilizes an amount of historical data that contains the easy-to-measure variables and the information of the variables to be estimated for establishing the models which can output the variables to be estimated by inputting a set of new easy-to-measure variables [4], [5]. Therefore, soft-sensors play a key role in producing high quality products when hardware sensors are not available.

There are many soft sensor methods based on statistical and machine learning methods [6], such as Partial Least Squares Regression (PLS), Multivariate Linear Regression (MLR), Gaussian Process Regression (GPR), Support Vector Machine (SVM), etc., in which least squares support vector machine (LSSVM) with the advantages of fewer training samples and low difficulty in training stands out and has been widely used to predict key variables in the steel, chemical, and other industries [7]. It can transform the quadratic programming problem for solving the parameters in SVM into linear equations by constructing equation constraints, which reduces the difficulty of solving model [8].

However, the two types of parameters in LSSVM including the penalty factor and the parameter of kernel function which can affect the predicted accuracy of model need to be set artificially [9]. Due to the equation constraints, the higher penalty factor can result in a better performance in training set, which will affect the generalization of the model [10]. Then, the kernel function can map samples to high-dimensional space. The parameters of kernel function can control the relative position and separation degree of samples in high-dimensional space, therefore, the extreme parameters are prone to over-fitting [11].

In determination of those two parameters or called as hyper-parameters, many scholars have done many relevant research works. In 1990s, Suykens has proposed the grid search and cross validation for searching hyper-parameters in LSSVM [12]. It is the earliest method of LSSVM hyper-parameter optimization. However, accuracy and efficiency of grid search are very low. Cross validation set will consume a lot of time. Then, based on the cross validation set, G Rubio derives the partial derivative of cross-validation error relative to hyper-parameters, which can

be solved by the gradient descent method [13]. But gradient descent is easy to fall into local optimum so that it is difficult to converge to the optimal solution. Cross validation set has the same problem as the last method.

Then, due to the faster search speed and the ability of searching global optimal solutions in most cases [14], intelligent optimization algorithms based on Biocomputing such as genetic algorithm (GA) [15], particle swarm optimization (PSO) [16], etc. have been widely used in hyper-parameters in LSSVM [17]. Although intelligent optimization algorithms compared with grid search and gradient descent owns the faster speed and the better search performance, the cross validation still owns the same problem [18]. In order to solve the time-consuming problem of cross validation, intelligent optimization combined with single validation set was proposed [19], [20], single validation set with similar distribution to the training data can be selected as the hyper-parameter optimization. Furthermore, intelligent optimization combined with all training set was proposed to simplify optimization process. Collecting validation set is sometimes cumbersome [21], therefore, it is often convenient to use all training samples as validation set for optimization, which mainly plays a fine-tuning role in hyper-parameters optimization. This method always requires a presetting for scope of hyper-parameters, then completes the fine-tuning. It is actually a common method in engineering [22].

Meanwhile, in actual industrial process, there are a lot of non-linear, dynamic, and state shifting data. It is difficult for a single model to keep original performance on these data [23]. In order to make soft sensing more adaptable and local learning frameworks have been proposed by the means of a combination of local models based on the distance between training samples and testing samples [24]. About local modeling of LSSVM, as early as 2007, Liu *et al.* [25] have proposed the LSSVM with local modeling and applied it in batch processes. However, they only used grid search and cross validation to select hyper-parameters once, and used this parameters in subsequent local modeling. They also analyzed that only one-off parameter selection would lead to worse results in subsequent models. Then Igne *et al.* [26] applied local weighted LSSVM to the prediction of soil parameters. They utilized intelligent optimization and K-fold cross validation for selecting hyper-parameters, and supposed that the predictive results would be better if online parameters optimization can be implemented in local modeling.

Furthermore, in the modern industry, more and more process data have been collected, which leads to a big data problem [27]. In this case, the computation of kernel matrix of LSSVM, local modeling mechanism and parameter optimization will be quite time-consuming [28], which limits the application of LSSVM in industrial big data.

In this paper, aiming at the targets of hyper-parameters optimization and local modeling with online optimization in LSSVM, LOS-LSSVM has been proposed. It can select a “substitute” as local objective set or sample (LOS) from candidate set for each testing sample by the distance between samples. The parameters selected by local objective set can make LSSVM achieve the best performance when it predicts each testing sample, which can avoid the loss of predictive accuracy caused by the mismatch of parameters selected by cross or single

validation sets above with state-shifting testing samples, and implement local modeling by adjusting hyper-parameters while retaining the information of all training samples. Then, aiming at optimization algorithms, this paper utilizes UMDA as a kind of estimation of distribution algorithms [30] for optimizing local objective set. Empirically, the hyper-parameters with better performance should be concentrated and fixed in a certain range. Therefore, they can be considered as random variables subject to a certain probability distribution. Besides, this paper also proposes distributed parallel (dp) UMDA-LOS-LSSVM based on the MapReduce computing framework, which can significantly reduce the time of local optimization and modeling. In addition, the dp-UMDA-LOS-LSSVM are constructed for soft sensor applications of an industrial process based on industrial big data analytics cloud platform.

## II. LSSVM WITH LOCAL OBJECTIVE SET FOR PARAMETER OPTIMIZATION BASED ON UMDA

### A. Least Square Support Vector Machine

In SVM, it usually utilizes quadratic programming method to determine the parameters, while LSSVM can use a set of linear equations for the same purpose [29]. Suppose the training dataset contains  $N$  training samples with input variables  $\mathbf{X} \in R^{N \times m}$  and output variable  $\mathbf{Y} \in R^{N \times r}$ , where  $m$  and  $r$  are dimensions. The model structure of LSSVM can be given as follows:

$$\hat{\mathbf{y}}_i = \mathbf{w}^T \Phi(\mathbf{x}_i) + b, i = 1, 2, \dots, N \quad (1)$$

where  $\{\mathbf{w}, b\}$  are the parameters of the linear regression function.  $\Phi(\cdot)$  is the nonlinear mapping function, which can map input samples to a high dimensional feature space. The parameters of the linear regression function can be obtained through solving the following constrained optimization problem [29]:

$$\begin{cases} \min_{\mathbf{w}, b, e} J(\mathbf{w}, e) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{i=1}^l e_i^2 \\ \text{s.t.} \quad \mathbf{y}_i = \mathbf{w}^T \Phi(\mathbf{x}_i) + b + e_i \end{cases} \quad (2)$$

where  $J(\mathbf{w}, e)$  is the loss function,  $\mathbf{w}$  is the weight vector,  $b$  is the error variable,  $e_i$  is the deviation variable, and  $\gamma$  is the penalty coefficient. Lagrangian multiplier method can be applied in solving the problem (2), according to the KKT condition, the solution of (2) is equal to the solution of a set of following equations [29]:

$$\begin{cases} \partial L / \partial \mathbf{w} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^l \alpha_i \phi(\mathbf{x}_i) \\ \partial L / \partial b = 0 \rightarrow \sum_{i=1}^l \alpha_i = 0 \\ \partial L / \partial e_i = 0 \rightarrow \alpha_i = \gamma e_i \\ \partial L / \partial \alpha_i = 0 \rightarrow \mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i = \mathbf{y}_i \end{cases} \quad (3)$$

where  $L$  is the Lagrangian multiplier, the model parameters  $\{\alpha_i, b\}$  can be obtained by solving (3) and the LSSVM regression model can be represented as follow:

$$\hat{\mathbf{y}} = \sum_{i=1}^N \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b. \quad (4)$$

In this paper, Radial Basis Function (RBF) is applied in LSSVM regression model, which can be formulated as

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right) \quad (5)$$

where  $\sigma^2$  is the parameter of RBF [29].

### B. LSSVM Modeling With Local Objective Set

The parameters including the penalty coefficient and the parameter of RBF, which can affect the fitting performance and generalization of LSSVM need to be seriously determined [29]. As have been introduced, it is the most common practice to use an intelligent optimization algorithm to minimize the prediction error of the whole training set. However, it may lead to the extreme model parameters and overfitting problem. In this part, the local objective set is proposed to improve the traditional method. The LSSVM model can be formulated as follows, where  $\mathbf{S}$  denotes the training data set,  $\gamma, \sigma^2$  denote the penalty coefficient and the parameter of RBF

$$\mathbf{Y} = \text{LSSVM}_s(\mathbf{X}, \gamma, \sigma^2). \quad (6)$$

First, the data collected from the process can be divided into two parts: the training data set  $\mathbf{S}_{\text{std\_train}}$ , the candidate data set  $\mathbf{S}_{\text{std\_candidate}}$ , which are normalized. The numbers of them can be represented as  $N, N_c$ . Meanwhile, the testing samples normalized by the means and standard deviation of training set can be denoted by  $\mathbf{S}_{\text{std\_test}}$ , of which the number can be 1 or  $N_t (N_t > 1)$ . Then, by selecting the most similar samples in the candidate set with testing samples as the local objective set or samples  $\mathbf{S}_{\text{obj}}$ . The similarity degree can be measured by Euclidean distance

$$d_{ij} = \|\mathbf{S}_{\text{std\_test}}(x_i) - \mathbf{S}_{\text{std\_candidate}}(x_j)\|_2 \quad (7)$$

where  $i = 1 \dots N_t, j = 1 \dots N_c$ , apparently, the number of samples in the objective set are the same as the number of testing samples. Apparently, considering the flow data, when testing samples go into the model one by one, the size of  $\mathbf{S}_{\text{obj}}$  is one, then, testing samples enter the model in batch order, the size of  $\mathbf{S}_{\text{obj}}$  is  $N_t$ , meanwhile,  $\mathbf{S}_{\text{obj}}$  is the subset of  $\mathbf{S}_{\text{std\_candidate}}$ . According to (6), utilize  $\mathbf{S}_{\text{std\_train}}$  and  $\mathbf{S}_{\text{obj}}$  for training LSSVM, which can be given as follows:

$$\hat{\mathbf{Y}}_{\text{obj}} = \text{LSSVM}_{\mathbf{S}_{\text{std\_train}}}(\mathbf{S}_{\text{obj}}, \gamma, \sigma^2). \quad (8)$$

$\hat{\mathbf{Y}}_{\text{obj}}$  and  $\mathbf{Y}_{\text{obj}}$  represent the predicted and the actual value of the objective set respectively. Therefore, the parameters optimization model of LSSVM can be constructed as follow. The range of hyper-parameters can be set broadly in advance, such as 0.001~1000

$$f = \min \left( \sum \left| \hat{\mathbf{Y}}_{\text{obj}} - \mathbf{Y}_{\text{obj}} \right| \right)$$

$$\text{s.t. } \begin{cases} \gamma_{\min} \leq \gamma \leq \gamma_{\max} \\ \sigma_{\min}^2 \leq \sigma^2 \leq \sigma_{\max}^2 \end{cases}. \quad (9)$$

The optimized parameters  $\gamma_{\text{best}}^{\text{obj\_set}}$  and  $\sigma_{\text{best}}^{2\text{obj\_set}}$  can be obtained by solving (9). Finally, utilize the training set and the

optimized parameters for training LSSVM model again and complete the test

$$\hat{\mathbf{Y}} = \text{LSSVM}_{\mathbf{S}_{\text{std\_train}}} \left( \mathbf{S}_{\text{std\_test}}, \gamma_{\text{best}}^{\text{obj\_set}}, \sigma_{\text{best}}^{2\text{obj\_set}} \right). \quad (10)$$

The LSSVM with local objective set for optimization can be called LOS-LSSVM for short. In contrast, the LSSVM with K folds cross validation set, single validation set, and all training set for optimization can be called KCV-LSSVM, SV-LSSVM, and ATS-LSSVM for short respectively. Meanwhile, LOS-LSSVM is a model for online soft sensing. When the number of testing set is 1, LOS-LSSVM can be modeled for every testing sample sent in real time.

### C. Parameters Solution of LOS-LSSVM With UMDA

Estimation of Distribution Algorithm (EDA) is a kind of evolutionary algorithms developed by the genetic algorithm (GA). It first samples and extracts information from the master population with better fitness. Then, it utilizes the information for building the proper probability model. At last, it uses the probability model and individuals with better fitness to update the population. The way to select new solutions can keep the population diversity and speed up the convergence [30].

Univariate Marginal Distribution Algorithm (UMDA) as a kind of EDA algorithms can be used to deal with the problem of the independent continuous variables [31], which utilizes the N one-dimensional Gaussian distribution functions for calculating the probability vectors of N variables [32]. The function can be represented as follow:

$$f(X_i = x_i) = \int_{-\infty}^{x_i} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(x_i - \mu_i)^2}{2(\sigma_i)^2}\right\} \quad (11)$$

where  $i = 1 \dots N$ ,  $X_i$  is the random variable,  $x_i$  is the value of  $X_i$ ,  $f$  is the probability of  $X_i$ ,  $\mu_i$  and  $\sigma_i$  are represented as the mean and standard deviation of variable  $i$  in the population respectively, which can be updated by following formulas, where  $M$  is the population size

$$\mu_i = \frac{\sum_{j=1}^M x_i^j}{M} \quad (i = 1, \dots, N) \quad (12)$$

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^M (x_i^j - \mu_i)^2}{M}} \quad (i = 0, \dots, N). \quad (13)$$

The master population can be selected by the truncation selection which is given as follows, where selectrate is the truncation selection rate, MS is the size of the master population, floor represents “round down”

$$\text{MS} = \text{floor}(\text{selectrate} \times M). \quad (14)$$

The algorithm of minimization based on UMDA are shown in Algorithm I [32].

Procedures of utilizing UMDA to optimize LOS-LSSVM are displayed in the flowchart in Fig. 1.

**Algorithm I:** The Algorithm of Minimization Based on UMDA.

**Input:**  $n$ : number of optimal variables;  $\mathbf{X}_i (i = 1 \dots n)$ : optimal variables;  $\mathbf{X}_i^{\min}, \mathbf{X}_i^{\max}$ : range of optimal variables;  $\text{pop}$ : population;  $m$ : population size;  $t$ : times of iteration;  $ms$ : number of master samples;  $\text{max\_iter}$ : max number of iteration;  $\text{rand}$ : generate a random number between 0 and 1;  $\text{calobj}$ : the objective function;  $\text{index}$ : the index of individuals selected;  $\text{sort}$ : sort function;  $\text{concat}$ : concatenate function;

**Output:** the final optimization variables  $\mathbf{X}_{\text{best}}$  and best fitness  $\text{fit}_{\text{best}}$

**Main Process:**

- 1: Generate  $m \times n$  Initial population  $\text{pop}^t = \mathbf{X}^{\min} + \text{rand} * (\mathbf{X}^{\max} - \mathbf{X}^{\min})$
- 2: For  $t = 1 : \text{max\_iter}$ 
  - 1)  $\text{fit} = \text{calobj}(\text{pop}^t)$  (size :  $m \times 1$ );
  - 2)  $\text{index} = \text{sort}(\text{fit}, \text{by} = \text{ascending}, \text{sele} = ms)$  (size :  $ms \times 1$ );
  - 3)  $\mathbf{X}_{\text{best}}^t = \text{pop}^t(\text{index}[0]), \text{fit}_{\text{best}}^t = \text{fit}(\text{index}[0])$ ;
  - 4)  $\text{pop}_{\text{master}}^t = \text{pop}^t(\text{index})$  (size :  $ms \times n$ );
  - 5) calculate  $\mu^t$  and  $\sigma^t$  of  $\text{pop}_{\text{master}}^t$  by (12), (13) ( $\mu^t, \sigma^t$  are  $n \times 1$ );
  - 6) utilize  $\mu^t, \sigma^t$  to update (11);
  - 7) generate  $m - ms$  probability vector  $\mathbf{p}^t$  (size :  $1 \times n$ );
  - 8) put all  $\mathbf{p}^t$  into updated (11), sample  $m - ms$  individuals denoted by  $\text{pop}_{\text{part}}^t$  (size :  $(m - ms) \times n$ );
  - 9)  $\text{pop}^{t+1} = \text{concat}(\text{pop}_{\text{master}}^t, \text{pop}_{\text{part}}^t, \text{by} = \text{rows})$  (size :  $m \times n$ );
- 3: Let  $t + 1$ ;
- 4: End for;
- 5: Output  $\mathbf{X}_{\text{best}}^t$  and  $\text{fit}_{\text{best}}$ .

### III. DISTRIBUTED AND PARALLEL UMDA-LOS-LSSVM IN DISTRIBUTED DATA ANALYTICS PLATFORM

#### A. Distributed and Parallel UMDA-LOS-LSSVM Based on MapReduce

When the number of training set is large, the modeling process for the LSSVM method becomes quite difficult. LSSVM in modeling needs to calculate kernel function matrix between samples and samples in training set. When the number of training samples is  $N$ , the time complexity of kernel function matrix is  $O(N^2)$ . Therefore, the scale of training samples is large, the calculation of kernel function matrix will be time-consuming. Meanwhile, UMDA-LOS-LSSVM needs to optimize hyper-parameters for each testing sample, and the selection of local objective set needs to calculate the distance between test samples and each training sample, of which the time complexity is still quadratic. Therefore, how to save the time of two steps above is the key to apply UMDA-LOS-LSSVM to large-scale data. Therefore, the idea of data parallelism training based on MapReduce can be proposed.

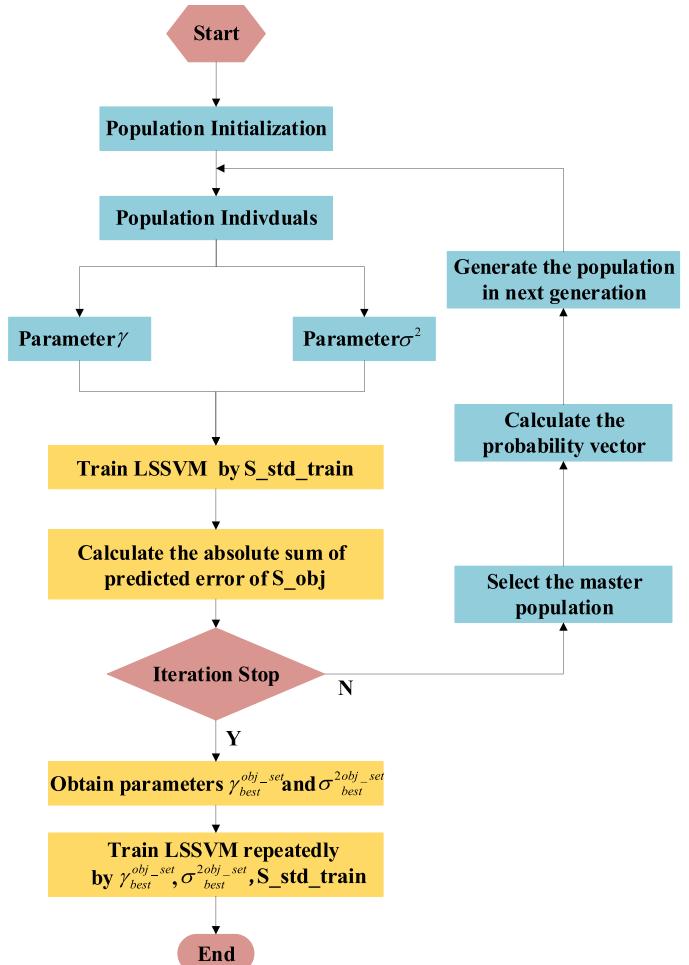


Fig. 1. Flowchart of UMDA-LOS-LSSVM for modeling.

MapReduce is a programming platform for modeling and analyzing big data in a distributed cluster of computing nodes. In the Map stage, the data in chunks will be read line by line in parallel and mapped into “key value pairs.” In the Reduce stage, the key value pairs with the same key will be reduced together and complete the follow-up operation in parallel [33], [34].

Distributed modeling of UMDA-LOS-LSSVM can be designed by two MapReduce.

The first one is that split UMDA-LOS-LSSVM into distributed and parallel submodels based on the idea of bagging in ensemble learning [36], and train LSSVM models by divide and conquer, then fuse the results in each sub models. The split numbers can be calculated by the ratio  $\alpha (\alpha > 1)$  of the number of training samples in every data block to the number of testing samples, which can control the loss of predicted precision due to deblock. The formula is given as follows

$$\text{Split\_Num} = \text{floor}(n / (\alpha \cdot n_t)). \quad (15)$$

The procedures are summarized as follows: First, determine the number of splitting blocks, in order to ensure the accuracy of model after divide and conquer, the number of blocks should not be excessive. We can control it by formula (15). Then sample with replacement from the training data stored in distributed

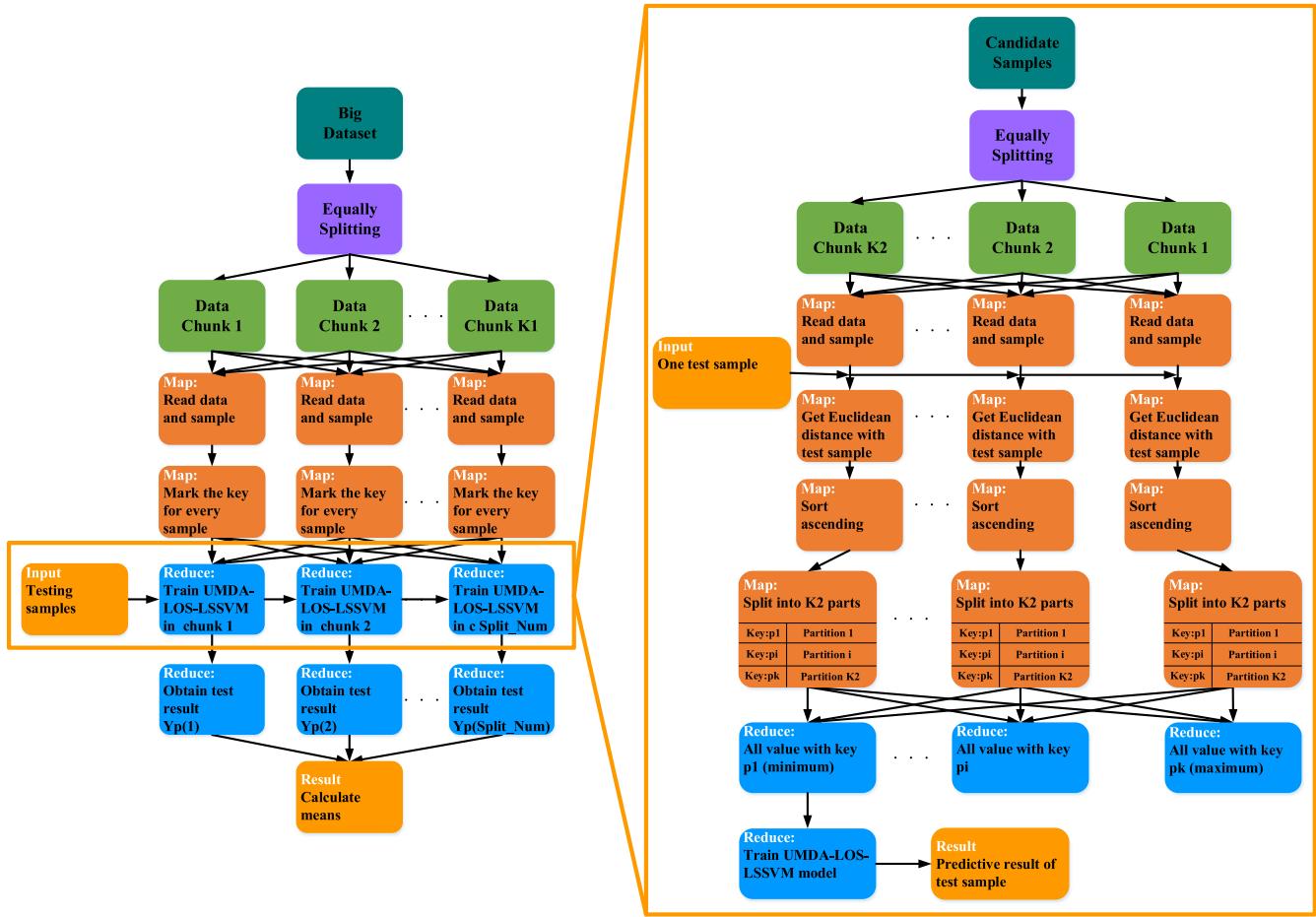


Fig. 2. Flowchart of distributed and parallel UMDA-LOS-LSSVM.

storage spaces equally and mark Split\_Num (K1 in Fig. 2) kinds of keys for these samples equally in Map stage. Finally, the samples with the same key can be clustered in a Reduce chunk and complete sub models training and hyper-parameter optimization, and results fusion by taking the average. The whole process can be illustrated in the left of Fig. 2.

The second MapReduce is distributed and parallel extraction of local objective set for testing sample aiming at the section “Train UMDA-LOS-LSSVM in chunk i” in the first MapReduce.

The procedures are summarized as follows: Supposed that the number of parallel blocks is K2. First, samples from distributed candidate data chunks equally, calculate the Euclidean distance between each candidate sample and the test sample, and sort all distances ascending. These operators are executed in parallel in Map stage. Then, still in Map stage, split the sorted data in each map into K2 parts equally and mark the key of each sample with their part number respectively. The samples in each map marked as part 1 are closest to the test sample. Finally, in reduce stage, all samples marked as part 1 required can be gathered in a reduce process, and choose the first sample in this reducer block as the local objective set for the testing sample. After extracting local objective set, hyper-parameters can be optimized by UMDA and prediction result can be outputted. The whole process can be illustrated in the right of Fig. 2.

TABLE I  
APPROXIMATE TIME COMPLEXITY ANALYSIS OF MAIN STEPS

Main Steps	dp-UMDA- LOS-LSSVM	UMDA- LOS-LSSVM
Computation of kernel matrix	$O(N^2/K_1^2)$	$O(N^2)$
Solve equations in LSSVM for training	$O(N^2/K_1^2)$	$O(N^2)$
Distance between testing and training samples	$O(N/K_2)$	$O(N)$
Sorting of distance	$O(2N/K_2)$	$O(N)$

Suppose that the number of splitting in obtaining submodels (Split\_Num) is K1, and the splitting number of extracting local objective set (K) is K2, the number of training and candidate samples are both N, the best sorting time complexity is  $O(N)$ . The approximate time complexity analysis of main steps for one testing sample in distributed model and nondistributed model is shown in Table I. From the approximate time complexity analysis, the value of K1 and K2 that we can control can adjust

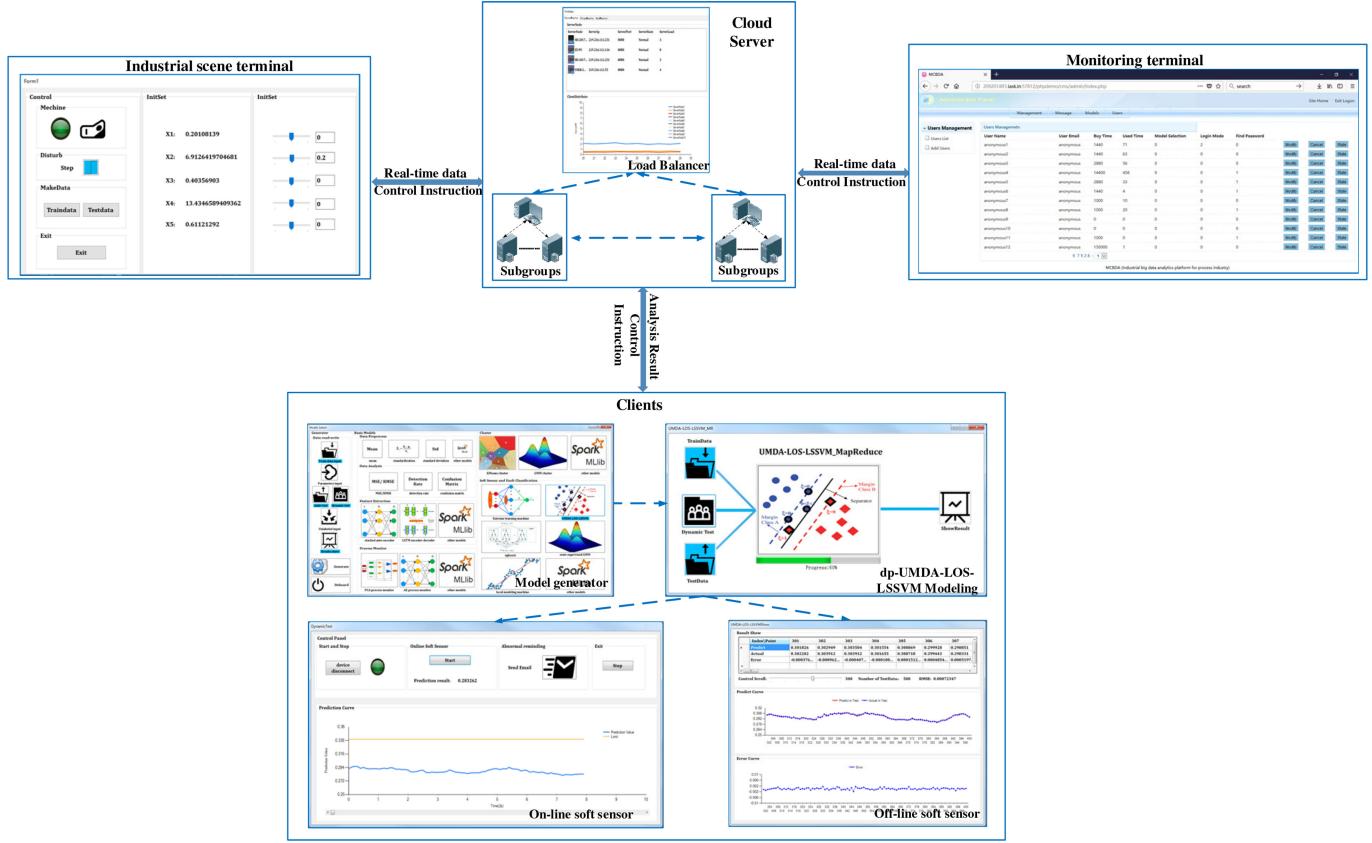


Fig. 3. Implement and architecture of industrial big data distributed analytics cloud platform.

the training time of dp-UMDA-LOS-LSSVM. Generally, K1 is smaller, which needs to maintain the predictive accuracy of LSSVM.

### B. Practical Application of UMDA-LOS-LSSVM Based on Industrial Big Data Distributed Analytics Cloud Platform.

In this section, we utilize self-developed industrial big data distributed analytics cloud platform with multifunctional subclusters in process industry (MCBDA) for implementing dp-UMDA-LOS-LSSVM so that complete the follow-up simulation experiments and actual industrial application. The architecture of MCBDA including “sub clusters and load balancer in clouds,” “industrial scene terminal,” “monitoring terminal,” and “clients terminal” have been shown in Fig. 3.

The process of modeling has been shown in “Clients” section of Fig. 3. First, users need to select the “UMDA-LOS-LSSVM” basis model and some data input-output modules in model generator so that MCBDA matches these modules and will automatically turn into “dp-UMDA-LOS-LSSVM Modeling” interface. Then, users can upload training set and off-line testing set, and the parameters of model have been predefined.

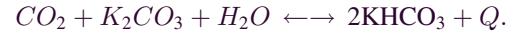
In the testing stage, the predictive results of off-line testing has been shown at the bottom right of Fig. 3. In the dynamic testing stage, users need to communicate with actual industrial object remotely. If the communication is successful, the online

soft sensing can be started. The online predictive results can be shown at the bottom left of Fig. 3. Meanwhile, if predictive results are abnormal and exceed a certain limit, the system will send messages to the mobile phone and trig an alarm in time. The platform MCBDA can help dp-UMDA-LOS-LSSVM to be widely used in simulation experiments and actual industries.

## IV. CASE STUDY

### A. Introduction of CO<sup>2</sup> Absorbing Column Unit

The principle of CO<sup>2</sup> absorption can be described by the following chemical reaction equation:



The CO<sup>2</sup> absorption process is shown by Fig. 4. After the initial cooling of the process gas from the previous unit, the process gas is cooled again in the process condensing tank 05F003 and enter the absorption column 05C001. Next, the process gas after the absorption column is sent into the mist removal separation tank 06F001, meanwhile, the residual CO<sup>2</sup> content is recorded by AR06001. Absorption fluid after absorbing CO<sup>2</sup> in 05C001 can be transformed from poor liquid and semi poor liquid to rich liquid which can be reabsorbed in the follow-up process.

In this process, the objective variable for soft sensor development is residual CO<sup>2</sup> content in the process gas. Inputs include 11 process variables that can be illustrated in Table II.

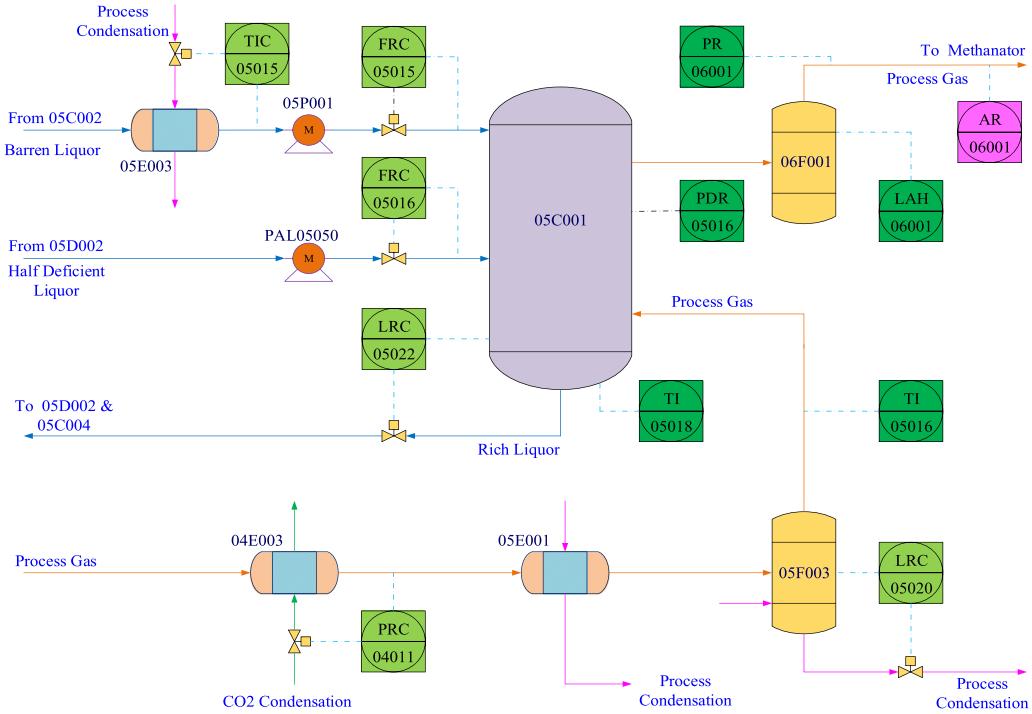


Fig. 4. Process chart of  $\text{CO}_2$  absorption column and recovery unit.

TABLE II  
DESCRIPTIONS OF THE 11 PROCESS VARIABLES

No.	Tags	Descriptions
1	PC04011.PV	Press. of process gas to 05E001
2	LC05020.PV	Level of 05F003
3	TC05015.PV	Exit poor liquid temp. of 05E003
4	FC05015.PV	Flow-rate of poor liquid to 05C001
5	FC05016.PV	Flow-rate of semi poor liquid to 05C001
6	TI05016.PV	Exit process gas temp. of 05F003
7	PDR05016.PV	Pressure difference between entrance and exit of 05C001
8	TI05018.PV	Exit rich liquid temp. of 05C001
9	LC05022.PV	Level of 05C001
10	LA06001.PV	High level alarm value of 06F001
11	PC06001.PV	Press. of process gas to 06 unit

TABLE III  
PARAMETERS SETTING OF UMDA

Population Size	Maximum Iterations	Selection Rate
70	10	0.4

### B. Prediction of Residual $\text{CO}_2$ Content Based on UMDA-LOS-LSSVM

In this part, 3000 samples have been collected from the process, 1500 of them are utilized for training, 500 of them are regarded as the testing samples, and 1000 of them are used to construct the candidate set.

The parameters setting of UMDA is shown in Table III, the optimization range of  $\gamma$  is 0.001~1000 and the range of

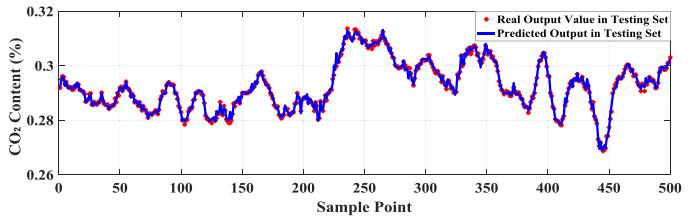


Fig. 5. Predicted curves in testing samples by UMDA-LOS-LSSVM.

$\sigma^2$  is 0.001~50. The predictive curve in testing samples has been shown in Fig. 5. The red points represent real value of testing samples, and the blue line represents predictive results of UMDA-LOS-LSSVM. Testing RMSE is 0.00072347, and maximum error is 0.0047.

Section IV-C-G will compare common methods to illustrate the advantages of UMDA-LOS-LSSVM in various aspects.

### C. Advantages of Hyper-Parameters Optimization in Local Objective Set About UMDA-LOS-LSSVM

In this section, the popular hyper-parameters of LSSVM selection methods include five folds cross validation from training set, single validation set with the same distribution as training set and all training set as validation. The three methods need to utilize UMDA to select one-off hyper-parameters. Their prediction and error curves compared with UMDA-LOS-LSSVM have been shown in following figures, and red points and blue line are the same as shown in Fig. 5. The green line represents the predictive results of these popular methods. The predictive

**TABLE IV**  
EVALUATION INDICATORS OF EXPERIMENTS

Evaluation Indicator	UMDA-LOS-LSSVM	UMDA-5CV-LSSVM	UMDA-SV-LSSVM	UMDA-ATS-LSSVM
Testing RMSE	0.00072347	0.00084572	0.00096108	0.0017
Testing Max Error	0.0047	0.0051	0.0066	0.0124
Optimized Parameters	$\gamma=249.3007$ $\sigma^2=1.99743$	$\gamma=391.2386$ $\sigma^2=0.70295$	$\gamma=532.6514$ $\sigma^2=0.52147$	$\gamma=926.32825$ $\sigma^2=0.23957$

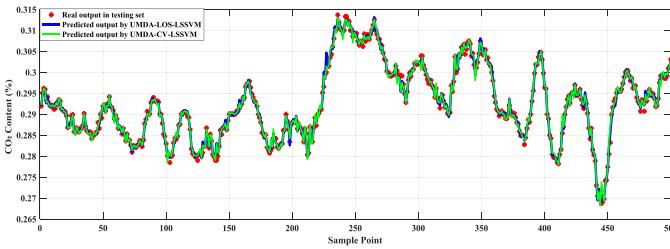


Fig. 6. Compared curves of UMDA-LOS/5CV-LSSVM.

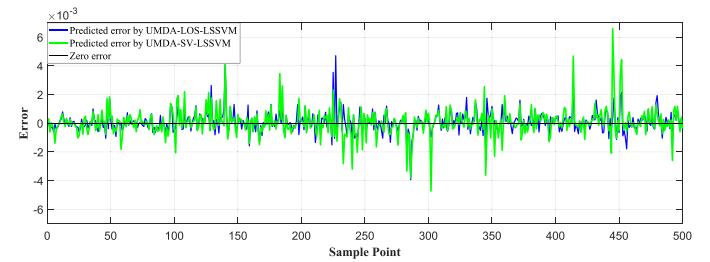


Fig. 9. Compared error of UMDA-LOS/SV-LSSVM.

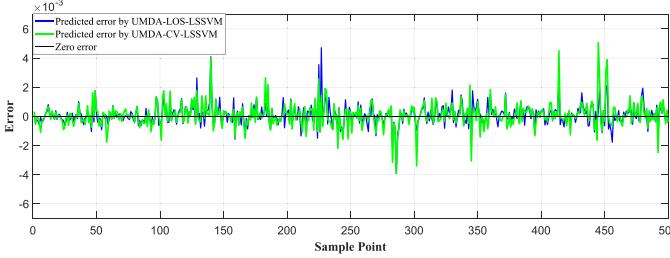


Fig. 7. Compared error of UMDA-LOS/5CV-LSSVM.

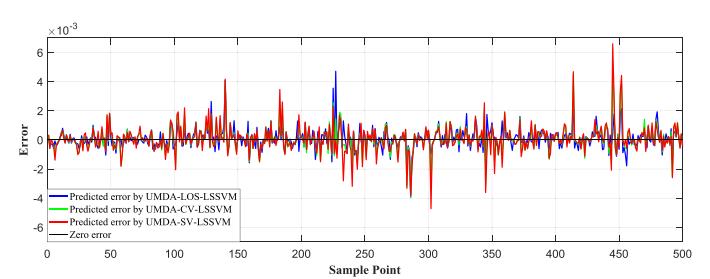


Fig. 10. Compared curves of UMDA-LOS/5CV/SV-LSSVM.

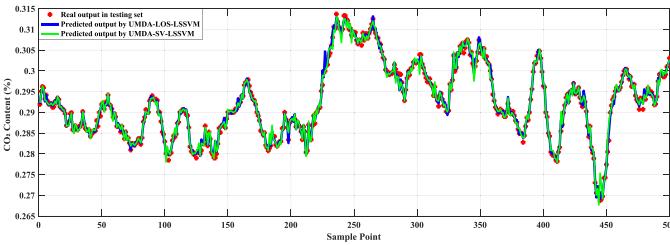


Fig. 8. Compared curves of UMDA-LOS/SV-LSSVM.

evaluation indicators have been shown in Table IV (Only a set of parameters optimized for a testing sample is shown in UMDA-LOS-LSSVM).

From the results of UMDA-5CV-LSSVM shown in Figs. 6 and 7, although cross validation can select parameters many times by splitting training set, there is still a gap between validation set and actual testing samples. However, optimization of local objective set is more effective for each testing sample. Therefore, the proposed model owns better performance. From the results of UMDA-SV-LSSVM shown in Figs. 8 and 9, single validation set only selects parameters once, and the effect of selection is related to the selection of validation set. Therefore, it is usually worse than cross validation set. Meanwhile, Fig. 10 has shown

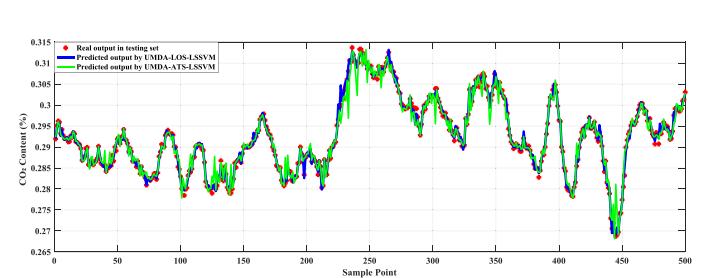


Fig. 11. Compared curves of UMDA-LOS/ATS-LSSVM.

the error curves of UMDA-LOS-LSSVM and validation set methods including UMDA-5CV/SV-LSSVM, which can show more intuitively that optimization based on local objective set is superior to the methods based on validation set and one-time optimization. From the results of UMDA-ATS-LSSVM shown in Fig. 11, because it is an empirical method for fine-tuning, when the range of hyper-parameters is too large, such as the range of parameters set in this experiment, the optimized parameters will lead to serious over-fitting, and the parameters optimized are extreme. Therefore, from Fig. 11, it is obvious that the performance of proposed method is better.

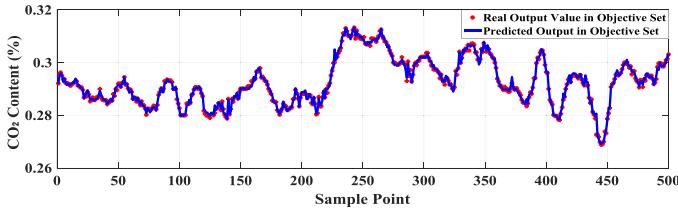


Fig. 12. Predicted curves in local objective set by UMDA-LOS-LSSVM.

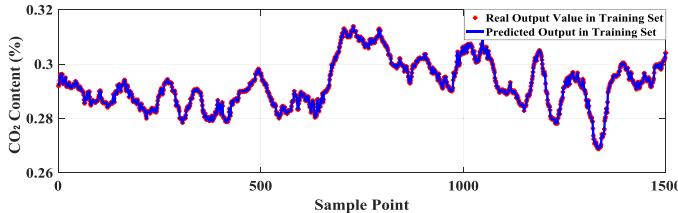


Fig. 13. Predicted curves in all training set by UMDA-ATS-LSSVM.

TABLE V  
EVALUATION INDICATORS OF EXPERIMENTS

Evaluation Indicator	UMDA-LOS -LSSVM	UMDA-ATS -LSSVM
RMSE of objective/training set	0.00081697 (objective set)	0.0000093382 (training set)
RMSE of testing set	0.00072347	0.0017
Optimized Parameters	$\gamma=249.30073$ $\sigma^2=1.99743$	$\gamma=926.32825$ $\sigma^2=0.23957$

#### D. Explanation of Advantages of Local Objective Set From the Point of Degree of Convergence to Testing Samples

In this section, compared with UMDA-ATS-LSSVM, it is more obvious that the optimization of local objective set is helpful to the fitting of testing samples. Fig. 12 has shown the predictive curves of local objective samples in UMDA-LOS-LSSVM. Fig. 13 showed the predictive curves of all training set in UMDA-ATS-LSSVM. The predictive evaluation indicators of local objective set and all training set have been shown in Table V. From Figs. 5 and 12 and their RMSE, the prediction result in testing set of UMDA-LOS-LSSVM is similar to the result in local objective set, which means that the local objective sample can serve as the “substitutes” of testing sample. However, from Figs. 11 and 13, due to the large optimization range of hyper parameters and all training set as optimization target, UMDA-ATS-LSSVM has been over-fitted, which make the prediction results of all training set are wonderful, but the result in testing samples are bad.

Figs. 14 and 15 show the RMSE drop curves (yellow line) in testing samples and convergence curves (blue line) of UMDA-LOS/ATS-LSSVM. Fig. 14 belongs to LOS, which means that each iteration of UMDA reduces the fitness error once, the RMSE corresponding to LSSVM with the parameters will decrease synchronously, which shows optimization of each step based on LOS is effective. However, Fig. 15 has shown

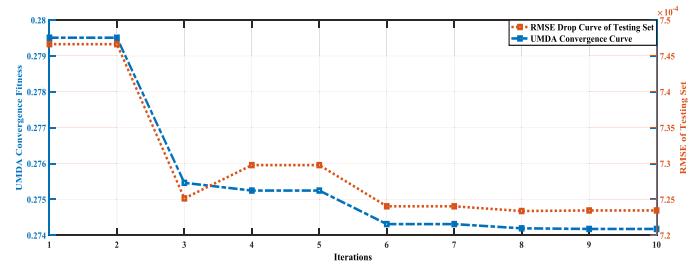


Fig. 14. RMSE drop and convergence curves by UMDA-LOS-LSSVM.

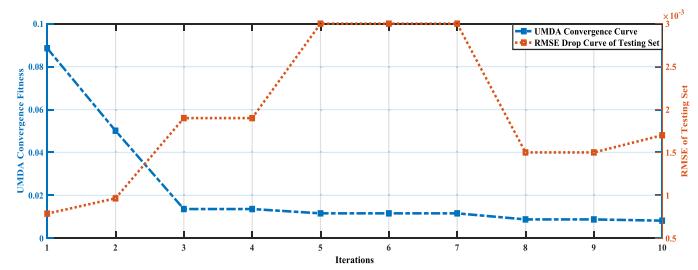


Fig. 15. RMSE drop and convergence curves by UMDA-ATS-LSSVM.

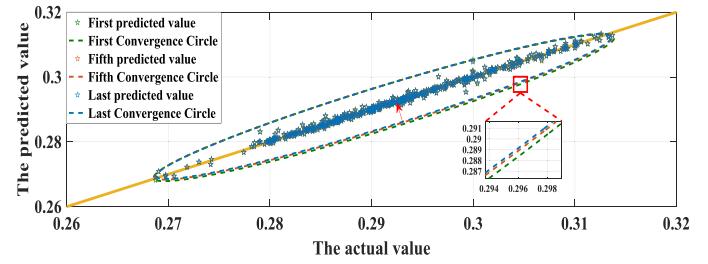


Fig. 16. Convergence process in testing set of the UMDA-LOS -LSSVM.

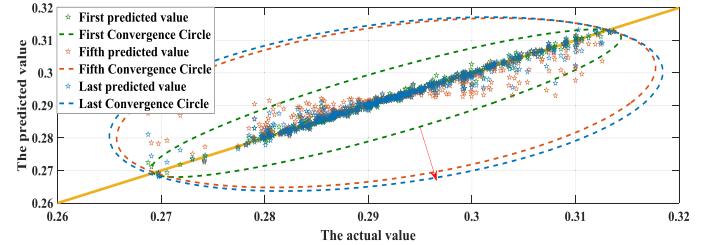


Fig. 17. Convergence process in testing set of the UMDA-ATS -LSSVM.

that the optimization process and RMSE drop are not synchronized, which shows optimization of each step based on ATS is ineffective.

Figs. 16 and 17 have shown that mapping all prediction results into a slant line. The more concentrated projection points mean the better predictive effect. The optimization times of UMDA is 10, we map prediction results corresponding to the first, fifth, and last optimized parameters into the slant line and draw enveloping lines of each predicted results. Fig. 16 belongs to LOS where the three enveloping lines are contractive, which means that the effect of optimization is uniformly valid. However, Fig. 17

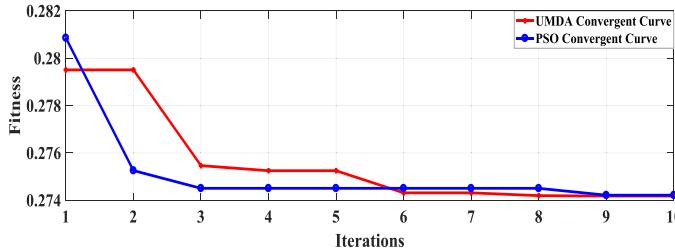


Fig. 18. Convergent curves of UMDA and PSO.

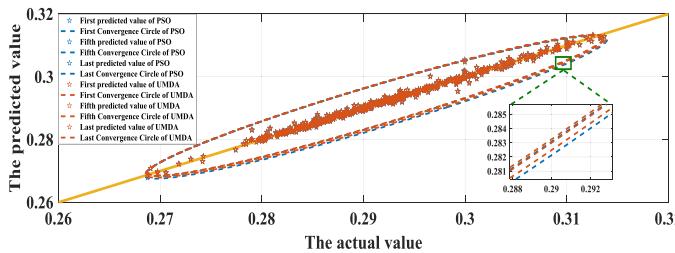


Fig. 19. Convergence process in testing set of UMDA and PSO.

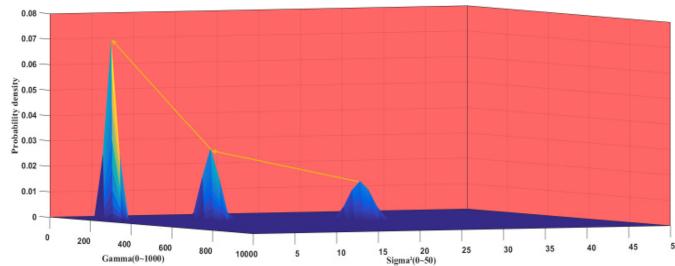


Fig. 20. Variety of Gauss distributions for hyper-parameters in UMDA.

belongs to ATS where the three enveloping lines are outspread, which means that the effect of optimization is irregular.

From the experiments above, UMDA-LOS-LSSVM can implement the most effective hyper-parameters optimization aiming at each testing sample.

#### E. Advantages of Optimization Algorithm UMDA for UMDA-LOS-LSSVM

In this section, the standard particle swarm optimization (PSO) is utilized for comparing with UMDA under the same conditions which is shown in Table III. The two algorithms used to optimize LOS-LSSVM are run eighty times and the results with the lowest final fitness are extracted for contrast respectively. Fig. 18 shows the convergent curves of UMDA and PSO, which indicates that the optimization performances of UMDA and PSO are similar, however, the final fitness of UMDA is lower. Fig. 19 illustrates convergent process in testing set of UMDA and PSO which shows that predicted circles of UMDA are smaller than the circles of PSO and are closer to the actual value. Meanwhile, in UMDA optimization process, the variation of Gaussian distributions that hyper-parameters obey has been shown in Fig. 20. Right to left denotes the first, fifth, and last iterations, respectively. The peak of Gaussian distribution be-

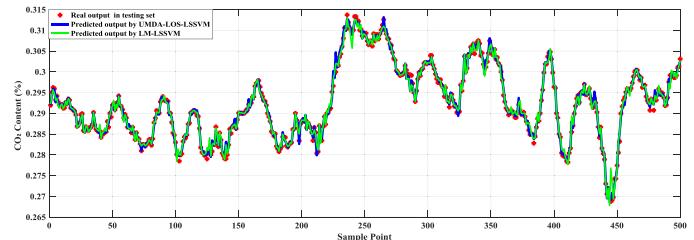


Fig. 21. Compared curves of UMDA-LOS/LM-LSSVM.

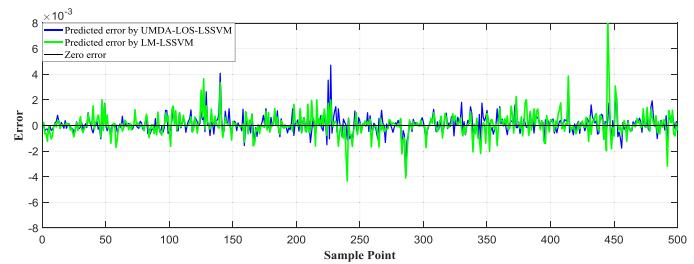


Fig. 22. Compared error of UMDA-LOS/LM-LSSVM.

**TABLE VI**  
EVALUATION INDICATORS OF EXPERIMENTS

Evaluation Indicator	UMDA-LOS-LSSVM	Local-Model-LSSVM
Testing RMSE	0.00072347	0.00093091
Testing Max Error	0.0047	0.0080

comes “thin” gradually, which indicates the optimal parameters are more concentrated.

#### F. Advantages of Local Modeling Compared With local LSSVM in UMDA-LOS-LSSVM

In this part, the implement process of local modeling LSSVM (LM-LSSVM) can be described as follows. First, utilize UMDA and five-folds cross validation to select hyper-parameters in LSSVM. Then, aiming at each testing sample, we select 120 samples closest to the testing sample (Euclidean Distance Metric) from training set. Finally, utilize the samples selected for training a local LSSVM model and complete prediction of the testing sample [25]. The prediction curves and prediction error of two methods have been shown in Figs. 21 and 22. The evaluation indicators are shown in Table VI. Due to all samples for training and on-line local hyper-parameters adjustment for each testing sample, performance of proposed method is better than local modeling.

#### G. Advantages of Fitting Ability Compared With Strong Regression Model GBDT in UMDA-LOS-LSSVM

Gradient Boosting Decision Tree (GBDT) which is one of the most effective algorithms for classification and regression applications [35]. In this experiment, the number of boost trees is 200, the learning rate is 0.3, and the max depth of boost tree is 30. Prediction curves and prediction error of two methods have been shown in Figs. 23 and 24. Evaluation indicators are

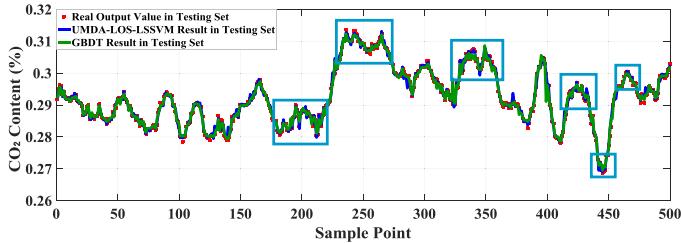


Fig. 23. Compared curves of UMDA-LOS-LSSVM/GBDT.

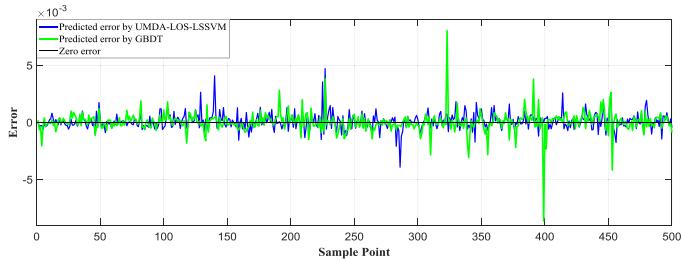


Fig. 24. Compared error of UMDA-LOS-LSSVM/GBDT.

TABLE VII  
EVALUATION INDICATORS OF EXPERIMENTS

Evaluation Indicator	UMDA-LOS-LSSVM	GBDT
Testing RMSE	0.00072347	0.00088973
Testing Max Error	0.0047	0.0086

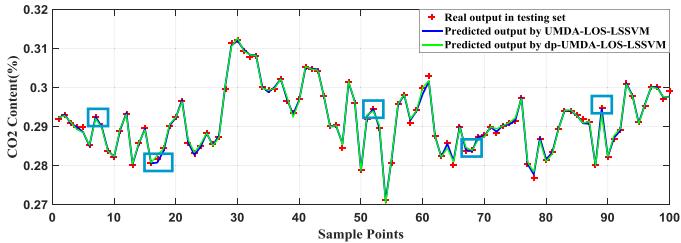


Fig. 25. Predicted result in testing set by dp-algorithm.

shown in Table VII. From results, the proposed method owns the advantages of local modeling, which can obtain better prediction effect at spikes. Parameters optimization in the proposed method can adjust nonlinear unit of model. It owns stronger nonlinear processing ability.

#### H. Prediction of Residual CO<sub>2</sub> Content Based on Distributed and Parallel UMDA-LOS-LSSVM

In this part, 10 000 samples have been collected from the process, 5000 of them are utilized for training models, 100 of them are regarded as the testing samples and 4900 of them are used to construct the candidate set. Meanwhile, the number ratio  $\alpha$  is set as 20. The distributed parameters in the Section III-A can be set: K1 is 2, and K2 is 3. The whole experiment can be completed in the platform in the Section III-B.

The predicted results of (distributed and parallel) dp-UMDA-LOS-LSSVM and it without distribution have been shown in Fig. 25, and evaluation indicators has been shown in Table VIII

TABLE VIII  
EVALUATION INDICATORS OF EXPERIMENTS

Methods	Optimized Parameters	RMSE	MAX_ERROR	Time
dp method	$\gamma=225.42485$ $\sigma^2=1.55869$	0.0005591	0.0018	880s
	$\gamma=432.91333$ $\sigma^2=1.22639$			
non-dp method	$\gamma=963.75085$ $\sigma^2=0.99906$	0.0005084	0.0017	6868s

in this paper. From the predicted results, dp and non-dp methods are similar as a whole. However, the number of training samples in submodels is less than non-dp model, which will reduce the accuracy of model more or less. It will be reflected in prediction of spikes, which has been marked by blue frameworks shown in Fig. 25. But from RMSE in Table VIII, the predictive loss of dp-model is small, and it is acceptable completely. More importantly, the small predictive loss can obtain large improvement in modeling time. Compared with non-dp model, the time of dp-modeling is almost 1/8 of non-dp modeling, which means that distributed and parallel UMDA-LOS-LSSVM proposed can be applied to large scale data under the condition of less precision loss.

The proposed model has been put into practical application in industrial big data distributed analytics cloud platform introduced in the Section III-B, and the effects of off-line and on line modeling have been shown in Fig. 3.

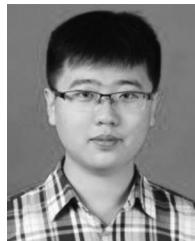
## V. CONCLUSION

In the present paper, the idea of local modeling was utilized for constructing the objective set of LSSVM parameter optimization, which extracts the samples similar to testing samples from the candidate set and forms the objective set. Meanwhile, the UMDA as a kind of estimation of distribution algorithms (EDA) with better convergence effect has been utilized for solving the problem of optimization, which improve the prediction accuracy. In experiments, UMDA-LOS-LSSVM can achieve better performance than common parameter optimization methods, local modeling and strong model. Then, the distributed and parallel UMDA-LOS-LSSVM was proposed based on the data parallel strategy, which can greatly reduce the time cost of model training in case of less prediction precision loss. Finally, all the proposed algorithms were implemented on the industrial big data distributed analytics cloud platform. The real industrial CO<sub>2</sub> absorbing column unit has demonstrated the effectiveness and superiority of the proposed algorithms for soft sensor modeling and application.

## REFERENCES

- [1] Z. Ge, "Distributed predictive modeling framework for prediction and diagnosis of key performance index in plant-wide processes," *J. Process Control*, vol. 65, pp. 107–117, 2018.
- [2] L. Yao and Z. Ge, "Deep learning of semi-supervised process data with hierarchical extreme learning machine and soft sensor application," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1490–1498, Feb. 2018.

- [3] Z. Ge, "Process data analytics via probabilistic latent variable models: A tutorial review," *Ind. Eng. Chem. Res.*, vol. 57, pp. 12646–12661, 2018.
- [4] P. Kadlec, R. Grbić, and B. Gabrys, "Review of adaptation mechanisms for data-driven soft sensors," *Comput. Chem. Eng.*, vol. 35, no. 1, pp. 1–24, 2011.
- [5] Z. Ge and X. Chen, "Dynamic probabilistic latent variable model for process data modeling and regression application," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 323–331, Jan. 2019.
- [6] Z. Ge *et al.*, "Data mining and analytics in the process industry: The role of machine learning," *IEEE Access*, vol. 5, pp. 20590–20616, 2017.
- [7] B. Lin and X. Gu, "Soft sensor modeling based on DE-LSSVM," *J. Chem. Ind. Eng.*, vol. 59, no. 7, pp. 1681–1685, 2008.
- [8] J. A. K. Suykens, J. Vandewalle, and M. B. De, "Optimal control by least squares support vector machines," *Neural Netw.*, vol. 14, no. 1, pp. 23–35, 2001.
- [9] R. Liao *et al.*, "Particle swarm optimization-least squares support vector regression based forecasting model on dissolved gases in oil-filled power transformers," *Electr. Power Syst. Res. Neural*, vol. 81, no. 12, pp. 2074–2080, 2010.
- [10] S. Tao, D. Chen, and W. Zhao, "Fast pruning algorithm for multi-output LS-SVM and its application in chemical pattern classification," *Chemometrics intellig. Lab. Syst.*, vol. 96, no. 1, pp. 63–69, 2009.
- [11] G. Rubio *et al.*, "A heuristic method for parameter selection in LS-SVM: Application to time series prediction," *Int. J. Forecasting*, vol. 27, no. 3, pp. 725–739, 2011.
- [12] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
- [13] G. Rubio *et al.*, "Efficient optimization of the parameters of LS-SVM for regression versus cross-validation error," in *Proc. Int. Conf. Artif. Neural Netw.*, 2009, pp. 406–415.
- [14] Z. Su *et al.*, "A new hybrid model optimized by an intelligent optimization algorithm for wind speed forecasting," *Energy Convers. Manage.*, vol. 85, pp. 443–452, 2014.
- [15] H. C. Jung, J. S. Kim, and H. Heo, "Prediction of building energy consumption using an improved real coded genetic algorithm based least squares support vector machine approach," *Pattern Recog.*, vol. 90, pp. 76–84, 2015.
- [16] R. G. Gorjaei *et al.*, "A novel PSO-LSSVM model for predicting liquid rate of two phase flow through wellhead chokes," *J. Natural Gas Sci. Eng.*, vol. 24, pp. 228–237, 2014.
- [17] H. Xu and G. Chen, "An intelligent fault identification method of rolling bearings based on LSSVM optimized by improved PSO," *Mech. Syst. Signal Process.*, vol. 35, no. 1, pp. 167–175, 2013.
- [18] G. Rubio *et al.*, "Efficient optimization of the parameters of LS-SVM for regression versus cross-validation error," in *Proc. Int. Conf. Artif. Neural Netw.*, 2009, pp. 406–415.
- [19] X. Zhu *et al.*, "A WD-GA-LSSVM model for rainfall-triggered landslide displacement prediction," *J. Mountain Sci.*, vol. 15, no. 1, pp. 156–166, 2018.
- [20] Z. Ying and K. C. Keong, "Fast leave-one-out evaluation and improvement on inference for LS-SVMs," in *Proc. IEEE Int. Conf. Pattern Recog.*, 2004, pp. 494–497.
- [21] K. De Brabanter *et al.*, "Fixed-size LS-SVM applied to the Wiener-Hammerstein benchmark," in *Proc. IFAC Proc. Volumes*, vol. 42, no. 10, pp. 826–831, 2009.
- [22] G. Si *et al.*, "Least squares support vector machine parameters optimization based on improved fruit fly optimization algorithm with applications," *J. Xi'an Jiaotong Univ.*, vol. 6, pp. 14–19, 2017.
- [23] S. Yin, X. Xie, and W. Sun, "A nonlinear process monitoring approach with locally weighted learning of available data," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1507–1516, Feb. 2017.
- [24] K. W. Smola and Q. H. Wu, "Local prediction of non-linear time series using support vector regression," *Pattern Recog.*, vol. 41, no. 5, pp. 1539–1547, 2008.
- [25] Y. Liu, H. Wang, and P. Li, "Local least squares support vector regression with application to online modeling for batch processes," *J. Chem. Ind. Eng.-China-*, vol. 58, no. 11, 2008, Art. no. 2846.
- [26] B. Igne *et al.*, "A evaluation of spectral pretreatments, partial least squares, least squares support vector machines and locally weighted regression for quantitative spectroscopic analysis of soils," *J. Near Infrared Spectrosc.*, vol. 18, no. 3, pp. 167–176, 2010.
- [27] J. Zhu, Z. Ge, Z. Song, and F. Gao, "Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data," *Annu. Rev. Control*, vol. 46, pp. 107–133, 2018.
- [28] L. Yao and Z. Ge, "Scalable semi-supervised GMM for big data quality prediction in multimode processes," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3681–3692, 2019.
- [29] S. He *et al.*, "An effective fault diagnosis approach based on optimal weighted least squares support vector machine," *Can. J. Chem. Eng.*, vol. 95, no. 12, pp. 2357–2366, 2017.
- [30] J. A. Lozano, "Estimation of distribution algorithms," *Genetic Algorithms Evol. Comput.*, vol. 64, no. 5, pp. 454–468, 2001.
- [31] I. Inza *et al.*, "Feature subset selection by genetic algorithms and estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with TIPS," *Artif. Intell. Med.*, vol. 23, no. 2, 2001, Art. no. 187.
- [32] Q. Zhang, "On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 8, no. 1, pp. 80–93, Feb. 2004.
- [33] J. Zhu, Z. Ge, and Z. Song, "Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data," *IEEE Trans. Ind. Inform.*, vol. 13, no. 4, pp. 1877–1885, Aug. 2017.
- [34] L. Yao and Z. Ge, "Big data quality prediction in the process industry: A distributed parallel modeling framework," *J. Process Control*, vol. 68, pp. 1–13, 2018.
- [35] L. Wang *et al.*, "Application of relative entropy and gradient boosting decision tree to fault prognosis in electronic circuits," *Symmetry*, vol. 10, no. 10, 2018, Art. no. 495.



**Xinyu Zhang** received the B.Eng. degree in automation from the Northeastern University, Shenyang, China, in 2017. He is currently working toward the M.Eng. degree in automation at the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His research interests include data-based modeling, deep learning, distributed computing, and soft sensor applications.



**Zhiqiang Ge** received the B.Eng. and Ph.D. degrees in automation from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively.

He was a Research Associate with the Department of Chemical and Biomolecular Engineering, Hong Kong University of Science Technology from July 2010 to December 2011, and a Visiting Professor with the Department of Chemical and Materials Engineering, University of Alberta from January 2013 to May 2013. He was an Alexander von Humboldt Research Fellow with the University of Duisburg-Essen during November 2014 to January 2017. He is currently a Full Professor with the College of Control Science and Engineering, Zhejiang University. His research interests include industrial big data, process monitoring, quality prediction, machine learning, and Bayesian methods.