This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2017.2733448, IEEE Transactions on Industrial Electronics

Deep Learning of Semi-supervised Process Data with Hierarchical Extreme Learning Machine and Soft Sensor Application

Le Yao, Zhiqiang Ge, *Senior Member, IEEE*

***Abstract*—Data-driven soft sensors have been widely utilized in industrial processes to estimate the critical quality variables which are intractable to directly measure online through physical devices. Due to the low sampling-rate of quality variables, most of the soft sensors are developed on small number of labeled samples and the large number of unlabeled process data are discarded. The loss of information greatly limits the improvement of quality prediction accuracy. One of the main issues of data-driven soft sensor is to furthest exploit the information contained in all available process data. This paper proposes a semi-supervised deep learning model for soft sensor development based on the hierarchical extreme learning machine (HELM). Firstly, the deep network structure of auto-encoders (AE) is implemented for unsupervised feature extraction with all the process samples. Then extreme learning machine (ELM) is utilized for regression through appending the quality variable. Meanwhile, the manifold regularization method is introduced for semi-supervised model training. The new method can not only deeply extract the information that the data contains, but learn more from the extra unlabeled samples as well. The proposed semi-supervised HELM method is applied in a High-low Transformer to estimate the CO content, which shows a significant improvement of the prediction accuracy, compared to traditional methods.**

***Index Terms*—Deep Learning, Extreme Learning Machine, Manifold Regularization, Semi-supervised Learning, Soft Sensor.**

I. Introduction

In modern industrial processes, pivotal quality variables such as the product quality, the content of process gas and the melt indices should be accurately measured to effectively implement the control strategies in a product unit [1]. However, the physical devices for measuring these variables are quite expensive and usually subject to large delays and extreme working environments. Soft sensor, a kind of virtual sensing technique which estimates the hard-to-measure quality variables based on other easy-to-measure process variables to offer reliable and economical alternatives to these expensive physical measuring sensors, have been widely utilized for both process control and process monitoring purposes [2], [3]. It is a combination of data processing, data-driven modeling and software building techniques. Besides, the improvement of Distributed Control System (DCS) drives the modern industry moving towards the big data era. Therefore, data-driven soft sensors have attracted more and more interests of researchers [4], [5]. Compared to traditional first principle modeling methods which typically synthesized prior knowledge or experiences, data-based methods are more flexible. Among the data-driven modeling methods, the models based on neural network (NN) like artificial neural network (ANN) and back propagation neural network (BPNN), which are skilful in dealing with nonlinear relationship between the input and output variables, has been widely applied to soft sensor development [6], [7]. ANN and BPNN are good at approximating nonlinear functions, and the generalization of NN-based models are prominent. Moreover, advanced control capabilities for complex processes have been significantly improved based on NN-based soft sensors.

In last decade, a novel single-hidden layer feed-forward neural network (SLFN) named extreme learning machine (ELM) was proposed [8]. In ELM, the input weights and biases of hidden layers are randomly generated and the output weights are obtained through solving a regularized least squares. Also, it has been proved that SLFNs with randomly generated hidden neurons and tunable output weight maintains it universal approximation capability [9]. It also has been proved that ELM are much more efficient than both ANN and BPNN in network training [10]. In recent years, ELM has attracted more and more attention and has also been successfully applied for soft sensor modeling in industrial processes [11].

Recently, deep learning (DL) has become a hot topic for process modeling, soft sensor models based on deep neural network have also been developed with a significantly improved prediction accuracy. The deep learning methods are derived from the development of single-hidden layer networks. Comparatively, deep learning with multi-layer architecture usually has excellent representation performance than the shallow networks. For soft sensor development, the deep learning method like Deep Belief Network (DBN) [12] and Denoising Auto-encoders (DAE) [13] have achieved outstanding performance. However, those deep learning methods are mostly trained by BP algorithm and all the hidden parameters need to be fine-tuned multiple times for the entire system. Thus, the training of deep learning architectures are commonly time-consuming. In view of this, a hierarchical ELM (HELM) has been proposed by Huang [14], which shown a much efficient performance than the traditional deep learning methods in classification field. The HELM contains two main components: the unsupervised feature extraction and the supervised feature classification or regression. In

This work was supported in part by the National Natural Science Foundation of China (NSFC) (61673337).

Corresponding author: Zhiqiang Ge
The authors are with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China.
E-mail: gezhiqiang@zju.edu.cn .

unsupervised feature extraction, the ELM-based auto-encoder (ELM-AE) is utilized and each layer in the stack architecture can be considered as a sub-module; and for supervised feature classification or regression, the deep level features are first scattered by a random matrix, and then the original ELM is applied for final decision making or prediction. The random feature mapping theory of ELM is still applied in HELM and parameters tuning is not required for the entire system, which provides a much quicker training speed than the traditional BP-based deep neural networks. However, the HELM method has not yet been utilized for soft sensor modeling in the process industry.

In practical processes, input variables used for soft sensor modeling are those fast-sampling process variables like flow rates, temperatures and pressures, while output variables are often the key/quality variables which are difficult to acquire due to lower sampling-rate, expensive labeling cost, complicated and time consuming chemical analyses and so on [15]. Data samples that have both input and output parts are often called labeled data, while those only with input part are referred as unlabeled data. Therefore, the collected dataset from practical industrial processes are commonly partially labeled. Besides, the number of unlabeled data is usually much larger than that of the labeled data, and it is far not enough to build soft sensor model merely based on the labeled samples. However, traditional soft sensor modeling method are those supervised algorithms using only labeled process samples with its corresponding quality variable, and the large number of unlabeled process data are discarded. The loss of information greatly limited the improvement of prediction accuracy for soft sensors. According to this, several semi-supervised soft sensing methods, such as the Semi-supervised Probabilistic Principal Component Regression (SSPPCR) [16] and the Co-training Partial Least Square (Co-PLS) [17] have been proposed through utilizing labeled samples along with the unlabeled process samples. Thus, the information contained in the large number of unsupervised data can be exploited and the prediction performance has been further improved. The semi-supervised algorithms assume that the input data follows the same cluster structure or distribution in input space, and it can incorporate the labeled and unlabeled data into the learning process. Besides, a commonly used semi-supervised method is the manifold regularization, which assumes that the conditional probabilities of two similar samples should also be similar as well [18]. It can be easily embedded in a traditional method to improve a supervised method into a semi-supervised form. The manifold regularization framework have been utilized in several algorithms to form the semi-supervised learning and gained superior performance [19].

In this paper, a semi-supervised deep learning model based on HELM is proposed for soft sensor development. All the collected process data are utilized for modeling. The entire modeling algorithm contains two main components: the unsupervised feature extraction and the semi-supervised regression for prediction. Firstly, the deep network structure of auto-encoders is implemented for unsupervised feature extraction with inputs of all the process samples. Then, in high-level of the network, the original ELM is utilized for regression through appending the quality variable (output variable of the soft sensor). Meanwhile, the manifold regularization method is embedded in the ELM algorithm for semi-supervised learning. Thus, the modeling method not only deeply extracts the information that the data contains through the deep network, but takes advantage of the extra unlabeled process samples as well. The effectiveness and feasibility of the proposed semi-supervised HELM model for soft sensor development are demonstrated through a real industrial case.

The layout of this paper is given as follows. In Sec. II, the basic ELM algorithm is briefly reviewed. Then, the semi-supervised HELM method is proposed under the manifold regularization framework in Sec. III. Then, the procedures of soft sensor modeling based on semi-supervised HELM is presented in Sec. IV. In the next section, the proposed soft sensor is constructed for an industrial application. Finally, conclusions are made.

Notations of some important mathematical definitions employed in the paper are given as follows:

$\mathbf{I}_N$: Identity matrix of dimension N , a square matrix whose diagonal elements are "1"s and other elements are "0"s; $\mathbf{L}$: graph Laplacian matrix, $\mathbf{L}=\mathbf{D}-\mathbf{U}$, $\mathbf{U}=[\mu_{i,j}]$ is the similarity matrix and $\mathbf{D}$ is a diagonal matrix with the elements $D_{ii}=\sum_j^{l+u}\boldsymbol{\mu}_{i,j}$; $\mathbf{J}(\mathbf{x})$ is the output matrix of the hidden layer of ELM; $\mathbf{H}_i$ is the output matrix of the ith hidden layer of HELM; $\mathbf{W}_i$ represents the weight matrix between the $i-1$th and ith hidden layer.

II. Extreme Learning Machine

ELM is a training algorithm which efficiently determines the parameters of SLFN [8]. While training an ELM model, suppose that N training samples with input variables $\mathbf{X}\in R^{N\times m}$ and output variable $\mathbf{Y}\in R^{N\times r}$ are collected, where m and r represent the dimensions. ELM aims to learn an approximate function to estimate the value of $\mathbf{Y}$. In general, the training of ELM consists of two stages: the random mapping and the parameters solving. The random mapping stage is to construct the hidden layer with a fixed number of randomly generated mapping neurons, where the mapping is conducted through the Sigmoid function [20]:

$$J(\mathbf{x};\boldsymbol{\theta})=\left(1+\exp\left(-\left(\boldsymbol{\omega}^T\mathbf{x}+\mathbf{b}\right)\right)\right)^{-1} \tag{1}$$

where $\boldsymbol{\theta}=\{\boldsymbol{\omega},\mathbf{b}\}$ are the parameters of mapping function.

Since the parameters of the hidden mapping functions are randomly generated, the rest parameters, the output weights between hidden neurons and the output nodes, can be obtained through solving a regularized least square problem. We define the output vector of the hidden layer as $\mathbf{J}(\mathbf{x})\in R^{N\times n}$, where n is the number of hidden neurons. With the output weights $\boldsymbol{\gamma}\in R^{n\times r}$, the output of the network can be given as follows:

$$\hat{\mathbf{y}}_i=\mathbf{J}(\mathbf{x}_i)\boldsymbol{\gamma},\ i=1,2,...,N \tag{2}$$

In the parameters solving stage, the output weights are obtained through minimizing the following unconstrained optimization problem:

$$\min_{\boldsymbol{\gamma}\in R^{n\times r}}\ \Gamma_{ELM}=\frac{1}{2}\|\boldsymbol{\gamma}\|^2+\frac{C}{2}\|\mathbf{Y}-\mathbf{J}\boldsymbol{\gamma}\|^2 \tag{3}$$

where $\mathbf{J}=\left[\mathbf{J}(\mathbf{x}_1)^T,...,\mathbf{J}(\mathbf{x}_N)^T\right]^T\in R^{N\times n}$. The first item in (3) is the regularization term against over-fitting, and the second is the error vector, where C is the penalty coefficient. By setting the gradient of Γ_{ELM} with respect to $\boldsymbol{\gamma}$ to zero, we can get the following equation:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2017.2733448, IEEE Transactions on Industrial Electronics

$$\nabla\Gamma_{ELM} = \boldsymbol{\gamma} + C\mathbf{J}^T\left(\mathbf{Y} - \mathbf{J}\boldsymbol{\gamma}\right) = 0 \tag{4}$$

If the hidden matrix $\mathbf{J}$ is full of column rank, which is the usually case where the number of training samples are much bigger than the number of hidden neurons, (4) can be solved as follows:

$$\boldsymbol{\gamma}^* = \left(\mathbf{J}^T\mathbf{J} + \frac{\mathbf{I}_n}{C}\right)^{-1}\mathbf{J}^T\mathbf{Y} \tag{5}$$

where $\mathbf{I}_n$ is the identity matrix of dimension n. On the contrary, if $\mathbf{J}$ is full of row rank, which means that the hidden neurons are more than the training samples, the solution of $\boldsymbol{\gamma}^*$ is given as follows:

$$\boldsymbol{\gamma}^* = \mathbf{J}^T\left(\mathbf{J}\mathbf{J}^T + \frac{\mathbf{I}_N}{C}\right)^{-1}\mathbf{Y} \tag{6}$$

where $\mathbf{I}_N$ is an identity matrix of dimension N.

III. Semi-supervised Hierarchical Extreme Learning Machine

A. Hierarchical Extreme Learning Machine

In order to learn sufficient representation for achieving high generalization performance, deep neural network like Stack Auto-encoder (SAE) is commonly utilized for mining data features [21]. The HELM is a multi-layer feed-forward network whose parameters are learned through multiple layers of ELM-AE [22]. HELM merges the learning efficiency of ELM and the deep structure of AE to obtain a superior prediction performance.

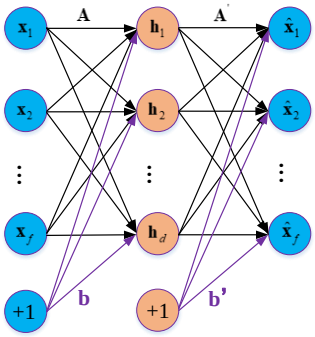


Fig 1. The structure of Auto-encoder

Essentially, the ELM-AE is built on the basis of ELM theory that randomly mapping can effectively ensure the capability of approximating continuous functions. The auto-encoder acts as a feature extractor in a multi-layer network shown as Fig 1. It uses the encoded outputs to approximate the original inputs by minimizing the reconstruction errors. Mathematically, the auto-encoder maps the input data $\mathbf{x}$ to a hidden layer $\mathbf{h}$ through the mapping $\mathbf{h} = g\left(\mathbf{A}\bullet\mathbf{x} + \mathbf{b}\right)$, where $\mathbf{A}$ is the input weight matrix, $\mathbf{b}$ is the bias vector and $g(\bullet)$ is the activation function. Then the representation $\mathbf{h}$ is mapped to a reconstructed input vector $\hat{\mathbf{x}} = g\left(\mathbf{A}'\bullet\mathbf{h} + \mathbf{b}'\right)$ with parameters $\mathbf{A}'$ and $\mathbf{b}'$. In ELM-AE, input weight matrix $\mathbf{A}$ is randomly generated and the mapped output $\hat{\mathbf{x}}$ is taken as $\mathbf{x}$. Then the output weight $\mathbf{A}'$ is calculated as (5) or (6). Similar as the basic ELM, once the auto-encoder is initialized, the fine-tuning of parameters is not required.

Fig 2 shows the process of learning HELM model from the training samples $\mathbf{X}$ and $\mathbf{Y}$. A fully connected multi-layer network is constricted here with h hidden layers. It can be seen that the HELM network is separated into two parts: the unsupervised feature extraction with multi-layer ELM-AEs and the supervised ELM regression. Training samples are firstly transferred to the $h-1$ layers of ELM-AE to extract deep features, and then the features are randomly mapped in the last layer for final prediction $\mathbf{Y}$ with an original ELM. The parameters $\mathbf{W}_i$ $(i = 1,...,h)$ represents the weight between the $i-1$th and ith hidden layer, and $\boldsymbol{\gamma}$ denotes the output weight of the last ELM. It should be mentioned that the biases nodes are incorporated in the input and hidden neurons.

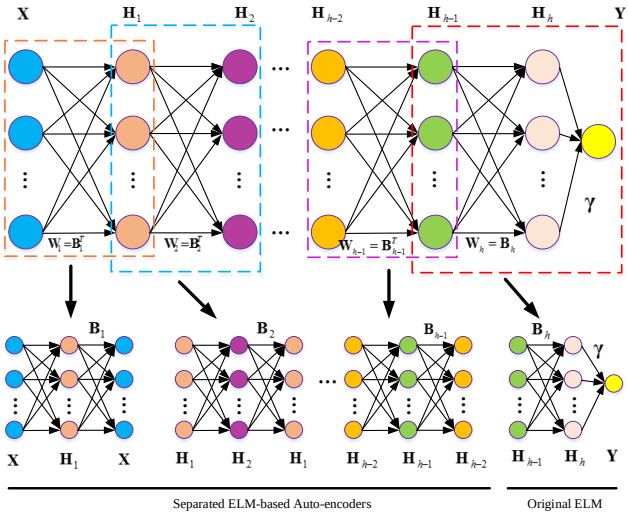


Fig 2. The learning process of Hierarchical ELM

For the $h-1$ layers of ELM-AE, the output of each hidden layer are calculated as follows:

$$\mathbf{H}_i = g\left(\mathbf{H}_{i-1}\bullet\mathbf{W}_i\right) \tag{7}$$

where $\mathbf{H}_i$ is the output of the ith hidden layer, $\mathbf{H}_{i-1}$ is the output of the $i-1$th hidden layer. Each of the hidden layer is decoupled within the network and constructed as an auto-encoder, whose output is the same as its input. Then, the output weight of auto-encoder $\mathbf{B}_i$ $(i = 1,...,h-1)$ is calculated as the basic ELM using (5) or (6) and they are transposed to the weight of hidden layer of HELM. Besides, once the feature of the previous hidden layer is determined, the parameters of the current layer will also be determined without fine-tuning. In contrast to the traditional deep learning methods whose parameters are iteratively trained through BP algorithms and so on, the HELM possesses a much faster training speed.

B. Semi-supervised HELM

According to the structure in Fig 2, HELM can be utilized in data classification and regression. However, it is a supervised algorithm with one input corresponding to one output. As mentioned in the Introduction, the quality variable in real processes are very low sampled, which means that quality samples are much less than the process variables. Commonly, soft sensors are built on small number of labeled samples and large number of unlabeled samples are discarded. Therefore, to exploit the process information contained in the large unlabeled dataset, a semi-supervised HELM (SS-HELM) method for soft sensor modeling is proposed in this section.

Generally, predictive models trained by small number of labeled samples are not that accurate. However, if an extra unsupervised learning mechanism is introduced to determine the distribution of the samples, the prediction performance of the model would be improved. Manifold regularization is an unsupervised learning restriction which derives from the manifold learning to obtain the data distribution with all

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2017.2733448, IEEE Transactions on Industrial Electronics

available samples [18]. It is built on the assumption that the similar samples in both labeled samples $\mathbf{S}_l = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{l}$ and unlabeled samples $\mathbf{S}_u = \{\mathbf{x}_i\}_{i=1}^{u}$ are close enough. In this case, the most similar data are incorporated in the same distribution, which intends to minimize the following equation:

$$\hat{\Phi} = \frac{1}{2}\sum_{i,j}\mu_{ij}\left\|\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j\right\|^2 \tag{8}$$

where $\hat{\mathbf{y}}_i$ and $\hat{\mathbf{y}}_j$ denote the predicted outputs with respect to the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ and μ_{ij} is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, calculated by the following Gaussian function:

$$\mu_{i,j} = \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / 2\sigma^2\right) \tag{9}$$

Then (8) can be transformed into the following equation:

$$\hat{\Phi} = Tr\left(\hat{\mathbf{Y}}^T\mathbf{L}\hat{\mathbf{Y}}\right) \tag{10}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{U}$ represents the graph Laplacian matrix [18]. $\mathbf{U} = [\mu_{i,j}]$ is the similarity matrix and $\mathbf{D}$ is a diagonal matrix with the elements $D_{ii} = \sum_j^{l+u}\boldsymbol{\mu}_{i,j}$. In order to construct the semi-supervised learning algorithm, the regularized item (10) should be added to (3) to improve the prediction performance, which is given as follows:

$$\begin{aligned}\Gamma_{SS-ELM} &= \min_{\boldsymbol{\gamma}\in R^{nxr}} \frac{1}{2}\|\boldsymbol{\gamma}\|^2 + \frac{C}{2}\|\mathbf{e}_i\|^2 + \frac{\lambda}{2}Tr\left(\hat{\mathbf{Y}}^T\mathbf{L}\hat{\mathbf{Y}}\right)\\ s.t. \quad & \mathbf{e}_i = \mathbf{y}_i^T - \mathbf{J}(\mathbf{x}_i)\boldsymbol{\gamma}, \quad i = 1,...,l\\ & \hat{\mathbf{Y}}_i = \mathbf{J}(\mathbf{x}_i)\boldsymbol{\gamma}, \quad i = 1,...,l+u\end{aligned} \tag{11}$$

where $\mathbf{L} \in R^{(l+u)\times(l+u)}$ is the graph Laplacian matrix, and $\hat{\mathbf{Y}}$ is the assumed output matrix mapped by all the inputs, and λ is the trade-off parameter. Furthermore, (11) can be written as follows:

$$\Gamma_{SS-ELM} = \min_{\boldsymbol{\gamma}\in R^{nxr}} \frac{1}{2}\|\boldsymbol{\gamma}\|^2 + \frac{C}{2}\|\mathbf{Y} - \mathbf{J}\boldsymbol{\gamma}\|^2 + \frac{\lambda}{2}Tr\left(\boldsymbol{\gamma}^T\mathbf{H}^T\mathbf{L}\mathbf{H}\boldsymbol{\gamma}\right) \tag{12}$$

where $\mathbf{Y}$ is the real output of the labeled samples, $\mathbf{H} \in R^{(l+u)\times n}$ is the hidden neurons with respect to the input of all the available samples, given as follows:

$$\mathbf{H} = \begin{bmatrix} J\left(\mathbf{w}_1^T\mathbf{x}_1 + b_1\right) & J\left(\mathbf{w}_2^T\mathbf{x}_1 + b_2\right) & \cdots & J\left(\mathbf{w}_n^T\mathbf{x}_1 + b_n\right) \\ J\left(\mathbf{w}_1^T\mathbf{x}_2 + b_1\right) & J\left(\mathbf{w}_2^T\mathbf{x}_2 + b_2\right) & \cdots & J\left(\mathbf{w}_n^T\mathbf{x}_2 + b_n\right) \\ \vdots & \vdots & & \vdots \\ J\left(\mathbf{w}_1^T\mathbf{x}_{l+u} + b_1\right) & J\left(\mathbf{w}_2^T\mathbf{x}_{l+u} + b_2\right) & \cdots & J\left(\mathbf{w}_n^T\mathbf{x}_{l+u} + b_n\right) \end{bmatrix} \tag{13}$$

By setting the gradient of (12) to 0, the output weight can be obtained as follows:

$$\boldsymbol{\gamma}^* = \left(\mathbf{I}_n + C\mathbf{J}^T\mathbf{J} + \lambda\mathbf{H}^T\mathbf{L}\mathbf{H}\right)^{-1}C\mathbf{J}^T\mathbf{Y} \tag{14}$$

where $\mathbf{I}_n$ is an identity matrix of dimension n. Similar as (6), when the number of labeled data is fewer than the hidden neurons, the solution is given as follows:

$$\boldsymbol{\gamma}^* = \tilde{\mathbf{J}}^T\left(\mathbf{I}_{l+u} + C\tilde{\mathbf{J}}\tilde{\mathbf{J}}^T + \lambda\mathbf{L}\mathbf{H}\mathbf{H}^T\right)^{-1}C\tilde{\mathbf{Y}} \tag{15}$$

where $\mathbf{I}_{l+u}$ is an identity matrix with dimension $l+u$, $\tilde{\mathbf{J}}$ and $\tilde{\mathbf{Y}}$ are augment matrix with the last u rows equal to 0 and the first l rows equal to $\mathbf{J}$ and $\mathbf{Y}$, respectively.

After the construction of semi-supervised ELM, the last layer of HELM is then transformed into the semi-supervised form. According to Fig. 2, when the output of the penult hidden layer is calculated by (7), it'll be taken as the input ($\mathbf{X}$) of the last ELM, and the partially collected quality samples ($\mathbf{Y}$) are incorporated to build the semi-supervised ELM. Finally, the last output weight is calculated through (14) or (15), and the SS-HELM model is completely constructed.

IV. Soft Sensor Modeling Based on SS-HELM

The procedures of the proposed soft sensor based on SS-HELM algorithm is summarized as the following table:

TABLE I
ALGORITHM 1

Algorithm 1: The SS-HELM algorithm for soft sensor modeling

Input: $\mathbf{X}$, $\mathbf{Y}$: original datasets; C *and* λ : penalty coefficient and trade-off parameter;

Output: $\mathbf{W}_i$ $(i = 1,...,h)$: weights between the $i-1$th and ith hidden layer; $\boldsymbol{\gamma}$: output weights; $\mathbf{Y}_{pre}$: the output prediction of testing sample.

Start:

Step 1: Variable selection according to the theoretical analysis and operator experiences;

Step 2: Determine the training dataset and the validation dataset;

Step 3: Normalize datasets; set C *and* λ ;

Step 4: Determine the number of layers for SS-HELM and the number of neurons for each layer;

Step 5: Randomly initialize the input weights and the weights between hidden neurons;

Step 6: Train the ELM-based multi-layer auto-encoder with all the labeled and unlabeled samples, obtain weight matrices $\mathbf{W}_i$ for each layer;

Step 7: Train the Semi-supervised ELM for the last layer with quality variable, obtain output weight $\boldsymbol{\gamma}$;

Step 8: Build the SS-HELM model for output prediction, test its performance;

Step 9: Predict the quality variable, $\mathbf{Y}_{pre}$, for testing samples.

END.

Intuitively, the procedures are displayed in the flowchart of SS-HELM for soft sensor modeling in Fig 3.

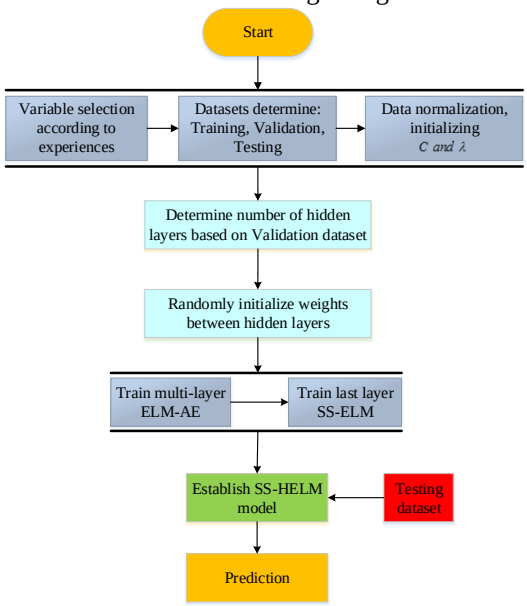


Fig 3. Flowchart of the SS-HELM algorithm for soft sensor modeling

V. Case Study: Application on CO Content Estimation in High-Low Transformer

The High-Low Transformer is a pivotal production unit derived from a real Ammonia Synthesis process. The main function of the High-Low Transformer is to transform the intractable carbon monoxide (CO) into carbon dioxide (CO_2), which would be absorbed in the subsequent CO_2 absorption column. The overall chemical reaction taken place is de-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2017.2733448, IEEE Transactions on Industrial Electronics

scribed as follows

$$CO + H_2O \xleftrightarrow{\Delta} CO_2 + H_2 \uparrow \quad (16)$$

The goal of this unit is to furthest reduce the content of CO in the process gas and to stably operate the process in an energy-saving style. Thus, the first and foremost procedure is to measure the content of residual CO at the out pipe of the unit as a critical quality variable. In real process, the content of residual CO is measured through offline laboratory analysis with an extremely low sampling rate. According to the process design, a brief flowchart of High-Low Transform unit with all the 27 process instruments is shown in Fig. 4, and the descriptions of the 26 process variables and the devices in the flowchart are presented in Table II and Table III, respectively.

TABLE II
DESCRIPTIONS OF THE 26 PROCESS VARIABLES IN THE FLOWCHART

No.	Tags (Type-SN)	Descriptions
1	AI-04001A	The flowrate to 04R001
2	AI-04001A-Ar	The content of Ar to 04R001
3	AI-04001A-CO	The content of CO to 04R001
4	AI-04001A-CH_4	The content of CH_4 to 04R001
5	AI-04001A-H_2	The content of H_2 to 04R001
6	AI-04001B	The flowrate to 04R002
7	AI-04001B-Ar	The content of Ar to 04R002
8	AI-04001B-CO_2	The content of CO_2 to 04R001
9	AI-04001B-CH_4	The content of CH_4 to 04R001
10	AI-04001B-H_2	The content of H_2 to 04R001
11	AI-04001B-N_2	The content of N_2 to 04R001
12	TI-04001	Temp. of 04R001's up level
13	TI-04002	Temp. of 04R001's middle level
14	PC-04003	Press. at the exit of 04R002
15	TI-04003	Temp. of 04R001's down level
16	TR-04004	Exit process gas temp. of 04R001
17	TI-04005	Temp. of BFW at 04E002
18	TC-04006	Exit process gas temp. of 04E002
19	TI-04008	Temp. of 04R002's up level
20	TI-04009	Temp. of 04R002's middle level
21	TI-04010	Temp. of 04R002's down level
22	LC-04011	The level of 04E003
23	PC-04011	Press. of process gas to 05 unit
24	TR-04011	Exit process gas temp. of 04R002
25	TI-04012	Temp. of recycled N_2 at 04K101
26	TI-04013	Entrance process gas temp. of 04R002

TABLE III
DESCRIPTION OF THE DEVICES IN THE FLOWCHART

Tags	Descriptions	Tags	Descriptions
04R001	High temp. transform column	04E001	Heat exchanger
04R002	Low temp. transform column	04E002	Heat exchanger
04F001	Gas-liquid separator	04E003	Heat exchanger

In order to demonstrate the effectiveness of the proposed SS-HELM for soft sensor modeling, 15 process variables have been selected from the 26 ones in Fig 4 according to the process design and the operation experiences. They are the most relevant variables to the CO content and the High-Low column operating condition. Most of the variables form the Analyzers are not that close to the final residual CO content except the input CO content of process gas and the flow-rate of process gas to the High-Low columns. The temperatures and the pressures on the columns are related to the operating condition in this unit, then they would contribute most in the modeling. The number of the chosen variables are 1, 3, 6, 12~16, 19~21, 23~26. To build the soft sensor model, 1000 labeled samples have been collected from the DCS database, 400 of them are utilized for training models, 200 of them are used for model parameter validation and the rest 400 are regarded as the testing samples. Additionally, other 600 unlabeled samples are also collected to demonstrate the proposed semi-supervised method.

TABLE IV
PARAMETER SETTINGS FOR ELM, SS-ELM, HELM AND SS-HELM

Parameters\Methods	C	λ	N_{elm}	N_{ae}
ELM\SS-ELM	5	0.2	20	-
HELM\SS-HELM	25	0.2	30	10

TABLE V
PREDICTION RESULTS WITH DIFFERENT NUMBER OF HIDDEN LAYERS ON THE VALIDATION DATASET (A=TRAININGRMSE, B=VALIDATINGRMSE)

Layers	HELM		SS-HELM	
	A	B	A	B
2	0.0035	0.0040	0.0033	0.0038
3	**0.0033**	**0.0039**	**0.0029**	**0.0033**
4	0.0037	0.0040	0.0034	0.0037
5	0.0041	0.0045	0.0043	0.0046
6	0.0044	0.0047	0.0037	0.0045
7	0.0041	0.0044	0.0038	0.0046

Before conducting the model training, some parameters are set based on the experienced values of the previous ELM researches [14], [19]. The penalty coefficient C and trade-off parameter λ are not that sensitive in an empirical region. Commonly, the value of C for ELM is determined in the interval $[1, 10]$. For HELM, the value of C is determined in the interval $[5, 30]$. For both ELM and HELM, λ is set between $[0.1, 0.5]$. In this paper, the two parameters are set through trial and error in the empirical intervals with the validating dataset. For ELM, the number of hidden neurons N_{elm} is usually set close to the number of input neurons. While for HELM, the number of the hidden neurons of the auto-encoders N_{ae} is commonly less than the number of input neurons. The detailed parameter values are set as Table IV. Then, to construct the SS-HELM model, the number of hidden layers should be determined. The models will be respectively trained on different number of hidden layers, and they have been utilized to predict the output of the validation dataset. Table V displays the prediction results. Both the HELM and SS-HELM have been validated on 2 to 7 layers of hidden neurons, where A and B represent the training and validating Root Mean Square Error (RMSE) [23], respectively. When the number of hidden layers equals 3, both the HELM and SS-HELM obtained the best prediction performance. However, if the number of hidden layers increases continuously but data sizes stay constant, over-fitting will occur and the prediction performance will not gain any improvements. Both the number of hidden layers and data sizes should be simultaneously increased to improve the performance of the deep network. It should be mentioned that all the results in this case are mean values of 20 times of training and predictions. Thus, in the following testing works, the number of hidden layers will be set as 3.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2017.2733448, IEEE Transactions on Industrial Electronics

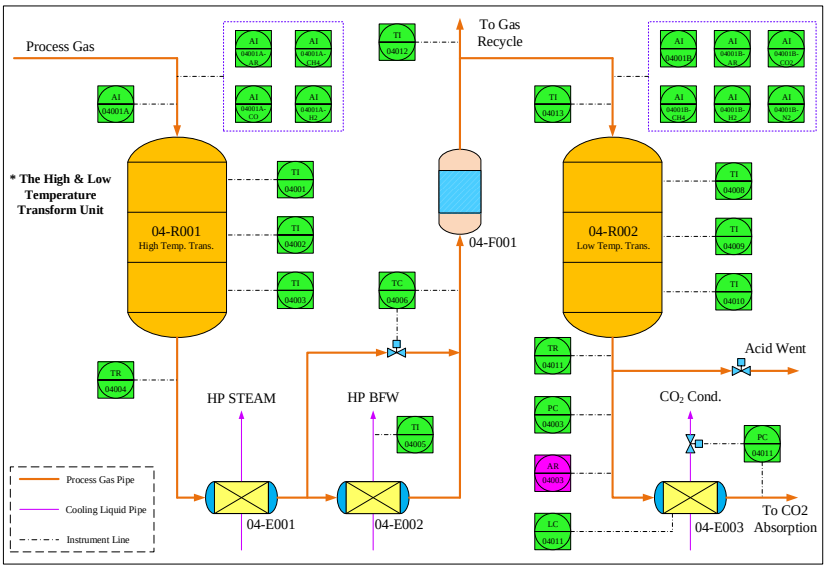


Fig 4. The flowchart of the High-Low Transform unit

TABLE VI

PREDICTION RESULTS OF SOFT SENSORS BASED ON THE FOUR ALGORITHMS WITH DIFFERENT NUMBERS OF UNLABELED SAMPLES (A=TRAININGRMSE, B=TESTINGRMSE, C=TRAININGTIME (S))

N_u	ELM			SS-ELM			HELM			**SS-HELM**		
	A	B	C	A	B	C	A	B	C	A	B	C
0	0.0040	0.0045	0.25	0.0038	0.0042	0.28	0.0033	0.0039	0.95	0.0030	0.0037	1.42
100	-	-	-	0.0037	0.0041	0.70	-	-	-	0.0031	0.0036	1.84
200	-	-	-	0.0036	0.0041	1.27	-	-	-	0.0029	0.0038	2.03
300	-	-	-	0.0036	0.0041	1.93	-	-	-	0.0032	0.0036	2.43
400	-	-	-	0.0033	0.0040	2.39	-	-	-	0.0030	0.0036	3.62
500	-	-	-	0.0034	0.0040	3.90	-	-	-	0.0031	0.0034	5.93
600	-	-	-	0.0032	0.0039	5.27	-	-	-	**0.0029**	**0.0033**	9.29

Secondly, the testing works of SS-HELM are conducted on different scale of unlabeled samples to demonstrate that the extra unlabeled samples would significantly affect the prediction performance. From 0 to 600, the unlabeled samples increased 100 a time to train different SS-HELM models. The 400 testing samples are utilized for the prediction. What's more, the basic ELM and HELM are also employed to train model only on the 400 labeled samples. As another comparison, the Semi-supervised ELM (SS-ELM) is also conducted on the same condition of SS-HELM. Detailed prediction results of CO content based on the four algorithms are presented in Table VI, where N_u denotes the number of unlabeled samples. It can be observed that the performance of SS-HELM is improving along with the increasing number of unlabeled samples. The same results can be seen in the comparison between the basic ELM and the SS-ELM. Besides, the performance of basic HELM is almost the same as the SS-HELM with no unlabeled samples, which can also be seen between the basic ELM and the SS-ELM. Thus, it can be concluded that the semi-supervised algorithms with extra unlabeled samples can effectively improve the prediction performance of the original supervised algorithms. On the other hand, the performance of the HELM denotes that the deep structure can learn more from the data and provide a superior prediction results than the basic ELM.

As an intuitive comparison, the prediction results and the prediction errors of the basic ELM, SS-ELM, HELM and SS-ELM are presented in Fig 5, 6, 7 and 8. For them, 400 labeled samples are utilized in the model training. However, the other 600 unlabeled samples are also participated in the Semi-supervised methods. Intuitively, the improvement of prediction performance can be seen from Fig 5 to Fig 8. For basic ELM, the prediction errors shown in Fig 5 are mostly kept at -0.01 and $+0.01$. The SS-ELM and HELM have improved this to $(-0.01, -0.005)$ and $(+0.005, +0.01)$. Finally, the prediction performance of SS-HELM has been further improved, whose prediction errors have been mostly decreased to the level of -0.005 and $+0.005$. Therefore, we can concluded that the semi-supervised method has improved the performance of basic algorithm, and the deep network structure is beneficial to data feature extraction.

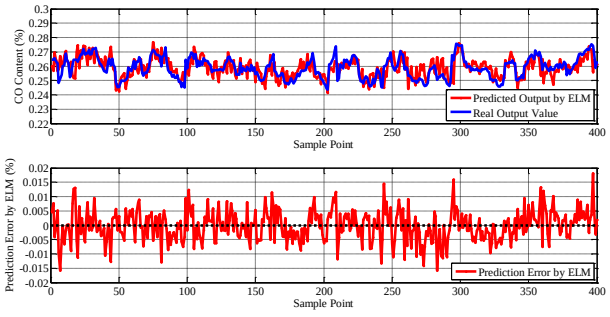


Fig 5. Prediction and Error of ELM (RMSE=0.0045)

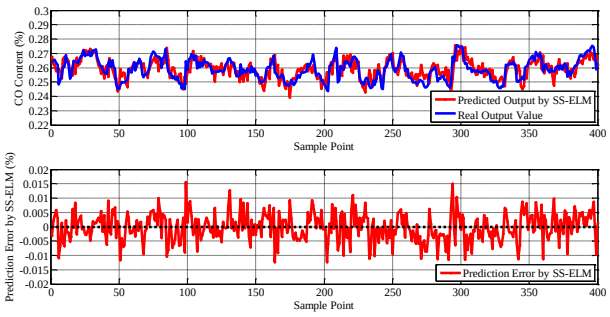


Fig 6. Prediction and Errors of SS-ELM (RMSE=0.0039)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIE.2017.2733448, IEEE Transactions on Industrial Electronics

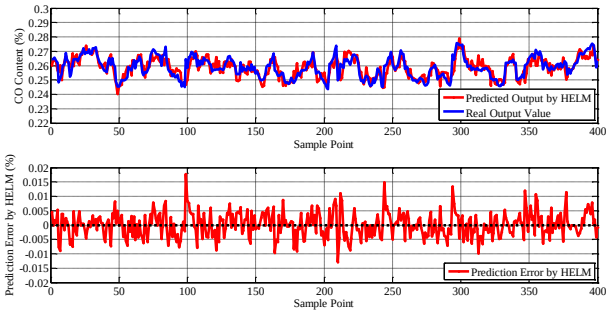


Fig 7. Prediction and Error of HELM (RMSE=0.0038)

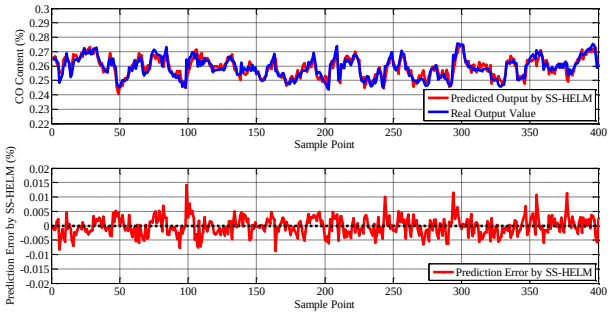


Fig 8. Prediction and Error of SS-HELM (RMSE=0.0033)

Furthermore, the proposed SS-HELM has been compared with 4 existing soft sensing methods, the SS-PPCR [16], Co-PLS [17], DBN [12] and SDAE-NN [13]. With the same number of training, testing and unlabeled samples as the SS-HELM, the soft sensor models are built and prediction results are displayed in Table VII. The SS-PPCR and Co-PLS are semi-supervised methods which sufficiently utilize the extra unlabeled samples for modeling, but their inherent linearity greatly limits the improvement of prediction accuracy. This also reflects on the prediction plots and the prediction errors shown in Fig. 9 and Fig. 10. The other two methods are derived from the deep learning domain, which have been effectively implemented for quality prediction. The DBN is pre-trained through a stacked Restricted Boltzman Machine (RBM) [12], and the obtained parameters are utilized to initialize a BP-NN. For SDAE-NN, the same technique is introduced, the initial weights of NN is obtained through training a stacked Denoising Auto-encoder. For them, the number of hidden layers and the hidden neurons are set the same as SS-HELM. The detailed prediction results are compared in Table 6, which shows that DBN and SDAE-NN both perform well for prediction. Especially, the TestRMSE of SDAE-NN is the same as SS-HELM, where auto-encoder plays an important role in the training. Since the BP algorithm is utilized in DBN and SDAE-NN, their training iterations cost much longer time than the SS-HELM. Finally, the prediction plots and errors of DBN and SDAE-NN are depicted in Fig. 11 and Fig. 12, respectively.

TABLE VII
COMPARED RESULTS WITH FOUR EXISTING METHODS
(A=TRAININGRMSE, B=TESTINGRMSE, C=TRAININGTIME (S))

Method\Indices	A	B	C
SS-PPCR [16]	0.0052	0.0057	16.96
Co-PLS [17]	0.0050	0.0051	14.73
DBN [12]	0.0032	0.0036	12.36
SDAE-NN [13]	0.0030	**0.0033**	22.78
SS-HELM	**0.0029**	**0.0033**	**9.29**

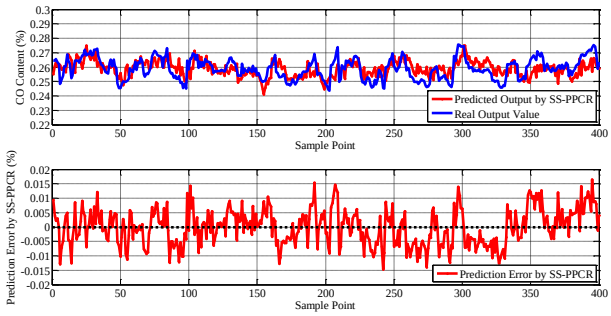


Fig 9. Prediction and Error of SS-PPCR (RMSE=0.0057)

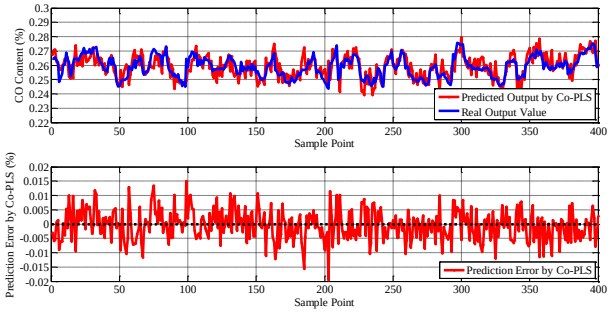


Fig 10. Prediction and Error of Co-PLS (RMSE=0.0051)

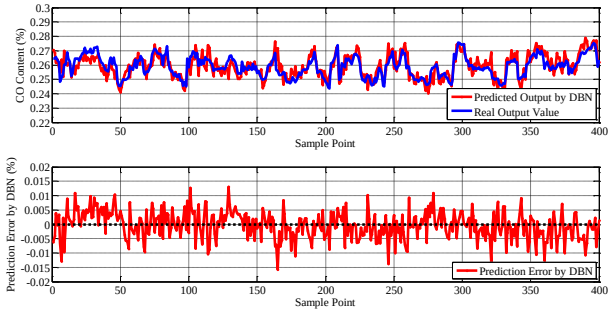


Fig 11. Prediction and Error of DBN (RMSE=0.0036)

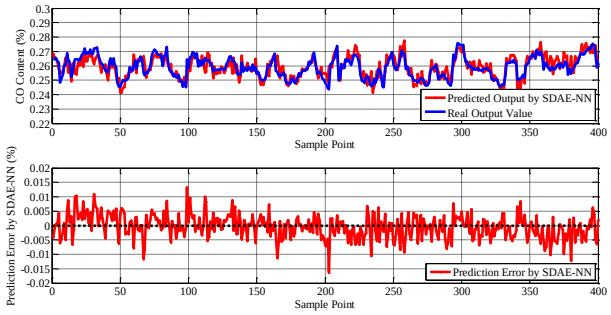


Fig 12. Prediction and Error of SDAE-NN (RMSE=0.0033)

VI. CONCLUSIONS AND OUTLOOK

In the present paper, the manifold regularization method is utilized to transform the basic HELM into a Semi-supervised HELM algorithm, which is further employed for soft sensor modeling. The deep network structure of auto-encoders is firstly implemented for unsupervised feature extraction with all the labeled and unlabeled samples, then the semi-supervised ELM based on manifold regularization is utilized for the regression in the last hidden layer. The combination of unsupervised feature extraction and semi-supervised learning greatly improves the prediction performance from the basic ELM and HELM, which means that the proposed method deeply and widely obtains information from the process data. The real industrial process of the High-low unit has demonstrated the effectiveness and

superiority of the proposed SS-HELM algorithm for soft sensor modeling, which can be further implemented in the process for CO content estimation.

As the sensors and instruments have been abundantly implemented in the industrial processes, the research on soft sensor networks would be a hot topic [24], [25], which can furthest obtain the state of huge distributed industrial systems for monitoring and control. The proposed approach can be further applied. Due to the fast learning speed, the ELM-based algorithms can be further applied in big data area for distributed modeling, which is always seeking for efficient computing [26]. Moreover, multiple process data forms like graphs, videos and voices could be utilized, then more extensive information can be learned through the proposed deep learning network.

REFERENCES

[1] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

[2] P. Kadlec, B. Gabrys, S. Strandt, "Data-driven soft sensors in the process industry," *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 795-814, 2009.

[3] Z. Ge, "Mixture Bayesian regularization of PCR model and soft sensing application," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4336-4343, 2015.

[4] Y. A. W. Shardt, H. Hao, S. X. Ding, "A new soft-sensor-based process monitoring scheme incorporating infrequent KPI measurements," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3843-3851, 2015.

[5] P. Kadlec, R. Grbić, B. Gabrys, "Review of adaptation mechanisms for data-driven soft sensors," *Computers & Chemical Engineering*, vol.35, no.1, pp. 1-24, 2011.

[6] J. C. Gonzaga, L. A. Meleiro, C. Kiang, et al., "ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process," *Computers & Chemical Engineering*, vol.33, no.1, pp. 43-49, 2009.

[7] A. Rani, V. Singh, J. R. Gupta, "Development of soft sensor for neural network based control of distillation column," *ISA Transactions*, vol.52, no.3, pp. 438-449, 2013.

[8] G. B. Huang, Q. Y. Zhu, C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol.70, no.1, pp. 489-501, 2006.

[9] G. B. Huang, L. Chen, C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transaction on Neural Networks*, vol.17, no.4, pp. 879-892, 2006.

[10] G. B. Huang, H. Zhou, X. Ding, et al., "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.42, no.2, pp. 513-529, 2012.

[11] Y. L. He, Z. Q. Geng, Q. X. Zhu, "Data driven soft sensor development for complex chemical processes using extreme learning machine," *Chemical Engineering Research and Design*, vol. 102, pp. 1-11, 2015.

[12] C. Shang, F. Yang, D. Huang, et al., "Data-driven soft sensor development based on deep learning technique," *Journal of Process Control*, vol.24, no.3, pp. 223-233, 2014.

[13] W. Yan, D. Tang, Y. Lin, "A data-driven soft sensor modeling method based on deep learning and its application," *IEEE Transactions on Industrial Electronics*, 2016, DOI: 10.1109/TIE.2016.2622668.

[14] J. Tang, C. Deng, G. B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol.27, no.4, pp. 809-821, 2016.

[15] K. Fujiwara, M. Kano, S. Hasebe, et al., "Soft sensor development using correlation based just in time modeling," *AIChE Journal*, vol.55, no.7, pp. 1754-1765, 2009.

[16] Z. Ge, Z. Song, "Semi-supervised Bayesian method for soft sensor modeling with unlabeled data samples," *AIChE Journal*, vol.57, no.8, pp. 2109-2119, 2011.

[17] L. Bao, X. Yuan, Z. Ge, "Co-training partial least squares model for semi-supervised soft sensor development," *Chemometrics and Intelligent Laboratory Systems*, vol.147, pp. 75-85, 2015.

[18] M. Belkin, P. Niyogi, V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol.7, pp. 2399-2434, 2006.

[19] G. Huang, S. Song, J. Gupta, et al., "Semi-supervised and unsupervised extreme learning machines," *IEEE Transactions on Cybernetics*, vol.44, no.12, pp. 2405-2417, 2014.

[20] M. Uzair, F. Shafait, B. Ghanem, et al., "Representation learning with deep extreme learning machines for efficient image set classification," *Neural Computing and Applications*, 2016, DOI:10.1007/s00521-016-2758-x.

[21] F. J. Huang, Y. L. Boureau, Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pp. 1-8, 2007.

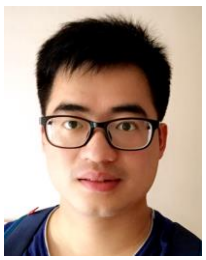
[22] G. E. Hinton, R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol.313, no.5786, pp. 504-507, 2006.

[23] L. Yao, Z. Ge, "Moving window adaptive soft sensor for state shifting process based on weighted supervised latent factor analysis," *Control Engineering Practice*, vol.61, pp. 72-80, 2017.

[24] P. Salvo Rossi, D. Ciuonzo, T. Ekman, "HMM-based decision fusion in wireless sensor networks with noncoherent multiple access," *IEEE Communications Letters*, vol. 19, no. 5, pp. 871-874, 2015.

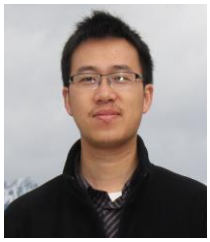
[25] D. Ciuonzo, A. Buonanno, et al., "Distributed classification of multiple moving targets with binary wireless sensor networks," *IEEE Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pp. 1-8, 2011.

[26] J. Zhu, Z. Ge, Z. Song, "Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data," *IEEE Transactions on Industrial Informatics*, 2017, DOI: 10.1109/TII.2017.2658732.



Le Yao received the B. Eng. and M. Eng. degrees from the Jiangnan University, Wuxi, China, in 2012 and 2015, respectively.

He is currently working towards the Ph.D. degree at the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include data-based modeling, distributed computing, process data analysis and soft sensor applications.



Zhiqiang Ge (M'13, SM'17) received the B. Eng. and Ph.D. degrees 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, Edmonton, AB, Canada, from January 2013 to May 2013. From November 2014 to January 2017, he was an AvH Research Fellow in University of Duisburg-Essen, Germany. He is currently a Full Professor with the Department of Control Science and Engineering, Zhejiang University. His research interests include industrial big data modeling and applications, process monitoring and fault diagnosis, predictive modeling and soft sensor, and Bayesian statistical learning and applications.