Engineering Journal issn: 0125-8281

Article

Development of a Data-Driven Soft Sensor for Multivariate Chemical Processes Using Concordance Correlation Coefficient Subsets Integrated with Parallel Inverse-Free Extreme Learning Machine

Thirasit Kusolsongtawee[1,a], **Soorathep Kheawhom**[2,b], **Sorin Olaru**[3,c], **and Pornchai Bumroongsri**[1,d,*]

1 Department of Chemical Engineering, Faculty of Engineering, Mahidol University, Nakhon Pathom 73170, Thailand
2 Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330, Thailand
3 Laboratoire des signaux et systèmes, Université Paris-Saclay, CNRS, CentraleSupélec, Gif-sur-Yvette 91190, France
E-mail: [a]thirasit.kos@student.mahidol.ac.th, [b]soorathep.k@chula.ac.th, [c]sorin.olaru@centralesupelec.fr, [d,*]pornchai.bum@mahidol.ac.th (Corresponding author)

Abstract. Nonlinearity, complexity, and technological limitations are causes of troublesome measurements in multivariate chemical processes. In order to deal with these problems, a soft sensor based on concordance correlation coefficient subsets integrated with parallel inverse-free extreme learning machine (CCCS-PIFELM) is proposed for multivariate chemical processes. In comparison to the forward propagation architecture of neural network with a single hidden layer, i.e., a traditional extreme learning machine (ELM), the CCCS-PIFELM approach has two notable points. Firstly, there are two subsets obtained through the concordance correlation coefficient (CCC) values between input and output variables. Hence, impacts of input variables on output variables can be assessed. Secondly, an inverse-free algorithm is used to reduce the computational load. In the evaluation of the prediction performance, the Tennessee Eastman (TE) benchmark process is employed as a case study to develop the CCCS-PIFELM approach for predicting product compositions. According to the simulation results, the proposed CCCS-PIFELM approach can obtain higher prediction accuracy compared to traditional approaches.

Keywords: Data-driven soft sensor, concordance correlation coefficient, extreme learning machine, multivariate chemical process.

ENGINEERING JOURNAL Volume 27 Issue 6
Received 25 March 2023
Accepted 19 June 2023
Published 30 June 2023
Online at https://engj.org/
DOI:10.4186/ej.2023.27.6.25

Nomenclature

A Input matrix
a_i The $i-$th sample in the input vector
a_{ix} $O_{\max_j}$ Maximum value of the data sample
$\overline{a}_x$ $O_{\min_j}$ Minimum value of the data sample
A^{Neg} Input matrix with negative CCC values
A^{Pos} Input matrix with positive CCC values
A^T Transpose of the input matrix
b_l Bias of the $l-$th hidden neuron
B_L Biases of IFELM for L hidden neurons
b_{l+1} Bias of the $l+1-$th hidden neuron
B^{Neg} Biases of the hidden layer with the negative subset for CCCS-PIFELM
B^{Pos} Biases of the hidden layer with the positive subset for CCCS-PIFELM
B^0 Biases of the output layer that is set at zero for CCCS-PIFELM
B^1 Biases among the input layer and the first part of the hidden layer for PELM
B^2 Biases among the input layer and the second part of the hidden layer for PELM
c Extra hidden layer output vector
$c_{L^{Neg}}^{Neg}$ Extra hidden layer output vector with negative CCC values for L^{Neg}
$c_{L^{Pos}}^{Pos}$ Extra hidden layer output vector with positive CCC values for L^{Pos}
c^T Transpose of matrix c
$f(\cdot)$ Transfer function
G_i The second part of hidden layer of the $i-$th sample for PELM
G^T Transpose of matrix G
G^+ Moore-Penrose pseudoinverse of the matrix G
H Hidden layer output matrix
H_i The first part of hidden layer of the $i-$th sample for PELM
H^{Neg} Hidden layer output matrix with a negative subset for CCCS-PIFELM
H^{Pos} Hidden layer output matrix with a positive subset for CCCS-PIFELM
H^T Transpose of matrix H
H^+ Moore-Penrose pseudoinverse of the matrix H
J_{ELM} Objective function for ELM
L Size of hidden neurons
L^{Neg} Size of hidden neurons in which extra hidden neurons are added to the subset with negative CCC values
L^{Pos} Size of hidden neurons in which extra hidden neurons are added to the subset with positive CCC values
m Size of the elements in the input vector
m_1 Size of the elements in the input vector with positive CCC values
m_2 Size of the elements in the input vector with negative CCC values
n Size of the elements in the output vector
N Number of training samples
N_{tst} Test sample size used in performance indicators
O_{ij} Original value of the data sample
O_{ij} Original value of the data sample
$O_{\max_j}$ Maximum value of the data sample
$O_{\mathrm{norm}_{ij}}$ Normalized value of the data sample
q_l Input weight of the $l-$th hidden neuron
Q_L Input weights of IFELM for L hidden neurons
q_{l+1} Input weight of the $l+1-$th hidden neuron
Q_{L+1} Input weights of IFELM for $L+1$ hidden neurons
Q^{Neg} Input weights of CCCS-PIFELM for the subset with negative CCC values
Q^{Pos} Input weights of CCCS-PIFELM for the subset with positive CCC values
Q^1 Input weights of PELM for the first part of the hidden layer
Q^2 Input weights of PELM for the second part of the hidden layer
r_{xt} Concordance correlation coefficient between the $x-$th input feature and the $t-$th output feature
S Training sample dataset
U Activation value of the hidden layer neurons for CCCS-PIFELM
U_L Activation value of the hidden layer neurons for IFELM
U_{L+1} Inverse-free recursive of the matrix H for IFELM
U_{L+1}^1 The first-part formulation for U_{L+1} in IFELM
U_{L+1}^2 The second-part formulation for U_{L+1} in IFELM
U^L Inverse-free recursive of the matrix H for CCCS-PIFELM
U_1^L The first-part formulation for U^L in CCCS-PIFELM
U_2^L The second-part formulation for U^L in CCCS-PIFELM
Z Target output
z_i The $i-$th sample in the output vector
z_{it} The n-dimensional vector of the $i-$th sample for the CCC calculation
$\overline{z}_t$ Mean value of the $t-$th output feature
z_p Observed value used in performance indicators
z'_p Predicted value used in performance indicators
$\hat{Z}$ Predicted output

Greek letters

β Output weights
β_{L+1} Output weights of IFELM with $L+1$ hidden neurons
β_t Weight between the hidden neurons and the $t-$th output neuron
β^1 Output weights of PELM for the first part of the hidden layer
β^2 Output weights of PELM for the second part of the hidden layer
ω Output weights with the biases of the output layer set to zero

1. Introduction

Due to the complicated behavior of multivariate chemical processes, advanced monitoring and control methods are required to obtain high product quality [1, 2]. The increasing complexity of industrial processes makes the development of process models time-consuming and difficult to achieve high accuracy [3, 4]. Some important process parameters such as efficiency and product composition are difficult to estimate and predict. Therefore, a soft sensor with high accuracy is a crucial element in industrial processes. With the emergence of statistical and neural network methods [5], various neural network methods, such as feedforward neural networks [6], functional link neural networks [7], and recurrent neural networks [8], are applied to deal with nonlinear relationships among input and output parameters [9].

An architecture of neural network using forward propagation with a single hidden layer has been generally used to predict process parameters because it has an explicit structure and good generalization performance [10]. Extreme learning machine (ELM) is a neural network with a single hidden layer in which the weights between neurons in input layer and neurons in hidden layer are assigned using a random approach. The weights between neurons in hidden layer and neurons in output layer are obtained by the Moore-Penrose (MP) pseudoinverse method [11, 12]. In terms of computations, ELM has a remarkably fast computational speed compared to networks based on the backpropagation (BP) method [11, 13]. In addition, it has been demonstrated that the ELM can surpass many neural networks in terms of generalization performance [14]. The ELM has been widely applied as a powerful method in a variety of fields, including regression [15-17], classification [18, 19], modeling [20-23], prediction [24-26], and control [27, 28]. Due to its outstanding features, ELM can be used to develop a data-driven soft sensor with good generalization performance and fast computational speed. Shao et al. [29] developed a probabilistic mixture of ELM with semi-supervised learning as a soft sensing approach to enhance representation capabilities and avoid overfitting problems. Zhang et al. [30] proposed integrated methods between the evolutionary algorithm and ELM to predict the melt index of products from propylene polymerization processes. They used the modified gravitational search algorithm to obtain suitable biases and weights for ELM.

The objectives of developing soft sensors for industrial applications are satisfactory precision and rapid feedback. In fields of process control [4, 31, 38], it is important that measurements from soft sensors are rigorous and reliable. The rapid feedback of soft sensors can result in a better process control performance. Moreover, soft sensors should be able to cope with some specific issues, such as strongly nonlinear relationships and interactions between input and output features. For the development of efficient soft sensors, some adjustments to ELM have to be implemented to meet particular needs and cope with specific issues. In the conventional ELM approach, relationships between input and output features are not included, so the regression accuracy may be reduced [32, 33]. In He et al. [34], the Pearson correlation coefficient was employed in order to deal with various effects of input features on output features. The combination of the Pearson correlation coefficient and ELM was utilized to develop a data-driven soft sensor for the purified terephthalic acid process. The details of input layer neurons could be instantly shifted to neurons in the output layer owing to the implementation of a double parallel structure. Gao et al. [35] used the enhanced online sequential ELM, which was determined by the configuration of the parallel layer network, for predicting the lifespan of integrated interchangeable avionics in modern aircraft systems. Li et al. [36] established the typical and online least squares parallel ELM for creating a model of combustion characteristics in boilers in order to diminish the pollution emissions.

Due to the fact that ELM randomly chooses thresholds and input weights for a hidden layer in the network, the local minimum can be obtained. The determination of suitable parameters in the hidden layer becomes a concerned issue. To address this problem, Guo et al. [37] developed the incremental ELM by adding hidden neurons one by one in order to achieve the desired approximation capability. Although this algorithm could update the output weights of newly added hidden neurons, the computational complexity can significantly increase [39].

In this research, a data-driven soft sensor using concordance correlation coefficient subsets integrated with a parallel inverse-free extreme learning machine (CCCS-PIFELM) is presented. The CCCS-PIFELM approach has two important features. Firstly, it has two subsets utilizing the concordance correlation coefficient (CCC) values between input and output variables. The input variables can be split into two subsets based on their CCC values, one with positive values and the other with negative values. Variables with positive CCC values are used to create a subset, and those with negative CCC values are also used to create another subset. The impacts of different input variables on output variables are considered as the subsets are created. Secondly, the CCCS-PIFELM approach uses an inverse-free algorithm assigned to deal with the limitations of matrix inversion operations. The weights between hidden and output layers are calculated using the inverse-free algorithm, so the computational load is reduced. In this research, the CCCS-PIFELM approach is employed to develop a novel soft sensor for the Tennessee Eastman (TE) benchmark process. This process is selected because there are numerous elements with highly nonlinear characteristics. Furthermore, the parallel extreme learning machine (PELM) and the inverse-free extreme learning machine (IFELM) are used for performance comparison.

The arrangement of this research is as follows: Preliminaries are introduced in Part 2. The specifications

DOI:10.4186/ej.2023.27.6.25

of the proposed CCCS-PIFELM are described in Part 3. The composition prediction of the TE benchmark process is examined through a demonstration in Part 4. In the final part, the conclusions of this research are presented.

2. Preliminaries

2.1. Extreme Learning Machine

Extreme learning machine (ELM) is an architecture of neural network with a single hidden layer. Parameters in the optimization are only the output weights, whereas the weights between input and hidden layers are randomly created in a specified range. The dataset for training with N samples is defined as $S=\{(a_i, z_i) \mid a_i \in \mathrm{R}^m, z_i \in \mathrm{R}^n, i=1,2,...,N\}$ where a_i denotes the $i-$th sample in the input vector, z_i denotes the $i-$th sample in the output vector, m denotes the size of elements in the input vector, and n denotes the size of elements in the output vector. The output can be calculated from the input as follows:

$$z_i = \beta_t \cdot f\left(q_l \cdot a_i + b_l\right) \tag{1}$$

where β_t denotes the output weight among the hidden neurons and the $t-$th output neuron, $f(\cdot)$ denotes the transfer function, q_l denotes the input weight among the input neurons and the $l-$th hidden neuron, and b_l denotes the bias of the $l-$th hidden neuron. The transfer function $f(\cdot)$ in the hidden layer can be calculated as follows:

$$H = \begin{bmatrix} f\left(q_1 \cdot a_1 + b_1\right) & \cdots & f\left(q_L \cdot a_1 + b_L\right) \\ \vdots & \ddots & \vdots \\ f\left(q_1 \cdot a_N + b_1\right) & \cdots & f\left(q_L \cdot a_N + b_L\right) \end{bmatrix}_{N\times L} \tag{2}$$

where L denotes the size of the hidden neurons. The cost function can be formulated by

$$minimize\ J_{ELM} = \frac{1}{2}\|\beta\|^2 + \frac{1}{2}\left\|Z - \hat{Z}\right\|^2 \tag{3}$$

where $\beta = \left[\beta_1^T, \beta_2^T, ..., \beta_L^T\right]^T \in \mathrm{R}^{L\times n}$ denotes the output weights, $Z = \left[z_1^T, z_2^T, ..., z_N^T\right]^T \in \mathrm{R}^{N\times n}$ denotes the target output and $\hat{Z}$ is the predicted output. The output weight can be calculated from

$$\beta = H^{+}Z \tag{4}$$

where $H^{+} \in \mathrm{R}^{L\times N}$ denotes the Moore-Penrose pseudoinverse of the matrix H that can be formulated by

$$H^{+} = H^T\left(HH^T\right)^{-1}. \tag{5}$$

2.2. Parallel Extreme Learning Machine

Parallel extreme learning machine (PELM) is developed on the principle of ELM, in which the input layer is projected onto two separate hidden layers, including $H_i \in \mathrm{R}^L$ and $G_i \in \mathrm{R}^L$, as follows:

$$H_i = f\left(Q^1 \cdot a_i + B^1\right) \tag{6}$$

$$G_i = f\left(Q^2 \cdot a_i + B^2\right). \tag{7}$$

In the PELM, the cost function is identical to that of the ELM, so the output can be calculated from

$$z_i = \begin{bmatrix}\beta^1 & \beta^2\end{bmatrix}\begin{bmatrix}H_i \\ G_i\end{bmatrix}. \tag{8}$$

The output weights $\beta = \begin{bmatrix}\beta^1 & \beta^2\end{bmatrix}$ can be calculated by the least possible sum of squares of residuals. They can be written as follows:

$$\begin{aligned}\beta &= \begin{bmatrix}H \\ G\end{bmatrix}^{+} Z \\ &= \begin{bmatrix}H \\ G\end{bmatrix}^{T}\left(\begin{bmatrix}H \\ G\end{bmatrix}\begin{bmatrix}H \\ G\end{bmatrix}^{T}\right)^{-1} Z.\end{aligned} \tag{9}$$

2.3. Inverse-free Extreme Learning Machine

Inverse-free extreme learning machine (IFELM) is developed for the purpose of reducing the computational expense. In practice, an appropriate size of hidden neurons is preferable because the computational complexity can increase with the size of hidden neurons. Hence, the computational complexity is traded off with the prediction accuracy. The required accuracy can be achieved by increasing the size of hidden neurons. The objective function for updating weights of $L+1$ hidden neurons based on weights of L hidden neurons can be formulated by

$$\min_{\beta_{L+1}}\left\|Z - \beta_{L+1} f\left(Q_{L+1}A + B_{L+1}\right)\right\|_F^2 \tag{10}$$

where $Q_{L+1} \in \mathrm{R}^{m\times(L+1)}$ is the input weights for $L+1$ hidden neurons, B_{L+1} is the biases for $L+1$ hidden neurons, and $\|\cdot\|_F$ is the Euclidean norm. The operation of matrix inversion may lead to prohibitive computational expense. With the aim of overcoming this problem, IFELM is employed to update the output weight as the size of hidden neurons increases. The output weight $\beta_{L+1} \in \mathrm{R}^{(L+1)\times n}$ of IFELM can be calculated from

$$\beta_{L+1} = U_{L+1} Z \tag{11}$$

$$U_{L+1} = \begin{bmatrix} U_{L+1}^1 & U_{L+1}^2 \end{bmatrix}_{(L+1)\times N} \tag{12}$$

$$U_{L+1}^1 = \frac{c^T cI - cc^T}{c^T c}\left(\frac{U_L Hcc^T U_L}{c^T c - c^T U_L Hc} + U_L\right) \tag{13}$$

$$U_{L+1}^2 = -\frac{U_{L+1}^1 Hc}{c^T c} + \frac{c}{c^T c} \tag{14}$$

where $H = f\left(Q_L A + B_L\right)$ and $c = f\left(A^T q_{l+1} + b_{l+1}\right)$.

2.4. Concordance Correlation Coefficient

The concordance correlation coefficient (CCC) is used to determine the relationship between two features. Consider a training dataset with N explicit instances $S = \left\{\left(a_i, z_i\right) | a_i \in \mathrm{R}^m, z_i \in \mathrm{R}^n, i = 1,2,...,N\right\}$ is available, where a_i, which includes m elements, is the vector of the $i-$th input sample, and z_i, which includes n elements, is the vector of the $i-$th output sample; the concordance correlation coefficient r_{xt} can be calculated from

$$r_{xt} = \frac{2\left(\dfrac{1}{N}\sum_{i=1}^{N}\left(a_{ix} - \overline{a}_x\right)\left(z_{it} - \overline{z}_t\right)\right)}{\dfrac{1}{N}\sum_{i=1}^{N}\left(a_{ix} - \overline{a}_x\right)^2 + \dfrac{1}{N}\sum_{i=1}^{N}\left(z_{it} - \overline{z}_t\right)^2 + \left(\overline{a}_x - \overline{z}_t\right)^2}$$
$$x = 1,2,...,m,\ t = 1,2,...,n \tag{15}$$

where $\overline{a}_x$ is the mean values of the $x-$th input feature, $\overline{z}_t$ is the mean values of the $t-$th output feature, and r_{xt} is the concordance correlation coefficient between the $x-$th input feature and the $t-$th output feature.

The CCC values can be used to determine the relationship between input and output variables. In cases where $r_{xt} > 0$, the input variable has a positive effect on the output variable. An increase in the input variable results in an increase in the output variable. In comparison, the output variable increases as the input variable decreases in the case of $r_{xt} < 0$.

Input variables can be classified into two subsets based on their positive and negative CCC values. A subset can be created by grouping together variables with positive CCC values, and another subset can be created by grouping together variables with negative CCC values. The influence of input variables on output variables can be considered.

3. Proposed Soft Sensor Based on Concordance Correlation Coefficient Subsets Integrated with Parallel Inverse-Free Extreme Learning Machine

In this section, a soft sensor based on concordance correlation coefficient subsets integrated with parallel inverse-free extreme learning machine (CCCS-PIFELM) is proposed for multivariate chemical processes. The concept of the proposed CCCS-PIFELM approach is shown in Fig. 1. The CCCS-PIFELM approach has two notable points. Firstly, there are two subsets obtained through the concordance correlation coefficient (CCC) values between input and output variables. Positive CCC values indicate that output variables are positively affected by input variables. Negative CCC values indicate that output variables are adversely affected by input variables. Secondly, an inverse-free algorithm is used to deal with the limitation of matrix inversion operations so the computational load can be reduced.

Suppose that N instances of the training dataset $S = \left\{\left(a_i, z_i\right) | a_i \in \mathrm{R}^m, z_i \in \mathrm{R}^n, i = 1,2,...,N\right\}$ are available, where a_i is the $i-$th input vector comprising m elements and z_i is the $i-$th output vector comprising n elements. The CCC values are calculated using Eq. (15). According to the CCC values between input and output variables, $A^{Pos} \in R^{N\times m_1}$ and $A^{Neg} \in R^{N\times m_2}$ comprise m_1 input variables with positive CCC values and m_2 input variables with negative CCC values, respectively, where $m_1 + m_2 = m$. The input weight matrices of subsets with positive and negative CCC values are $Q^{Pos} \in \mathrm{R}^{m_1\times L^{Pos}}$ and $Q^{Neg} \in \mathrm{R}^{m_2\times L^{Neg}}$, respectively, where L^{Pos} is the size of hidden neurons in which extra hidden neurons are added to the subset with positive CCC values and L^{Neg} is the size of hidden neurons in which extra hidden neurons are added to the subset with negative CCC values $\left(L = L^{Pos} + L^{Neg}\right)$. The biases of hidden layer with the positive and negative subsets for CCC values are $B^{Pos} = \left[b_1^{Pos}, b_2^{Pos}, ..., b_{L^{Pos}}^{Pos}\right]$ and $B^{Neg} = \left[b_1^{Neg}, b_2^{Neg}, ..., b_{L^{Neg}}^{Neg}\right]$, respectively.

The algorithm for prediction in the CCCS-PIFELM approach is described as follows: Instances $O = \left[O_{ij}\right] \in \mathrm{R}^{N\times(m+n)}$ are normalized into the interval [0, 1] using the following equation

$$O_{norm_{ij}} = \frac{O_{ij} - O_{\min_j}}{O_{\max_j} - O_{\min_j}} \tag{16}$$

DOI:10.4186/ej.2023.27.6.25

where $i = 1, 2, ..., N$ and $j = 1, 2, ..., m + n$.

For inputs a_i, $i = 1, 2, ..., N$, hidden layer outputs can be determined by

$$H^{Pos} = f\left(A^{Pos} Q^{Pos} + B^{Pos}\right) \quad (17)$$

$$H^{Neg} = f\left(A^{Neg} Q^{Neg} + B^{Neg}\right) \quad (18)$$

where $f(\cdot)$ is the transfer function, $H^{Pos} \in R^{N \times L^{Pos}}$ is the matrix of hidden layer output for a positive subset, and $H^{Neg} \in R^{N \times L^{Pos}}$ is the matrix of hidden layer output for a negative subset. The weight matrices Q^{Pos} and Q^{Neg} are randomly generated by utilizing a rectangular distribution within the interval of [−0.5, 0.5]. The matrix of hidden layer outputs $H \in R^{N \times L}$ can be written as

$$H = \begin{bmatrix} H^{Pos} & H^{Neg} \end{bmatrix} = \begin{bmatrix} H_{11}^{Pos} & H_{12}^{Pos} & \cdots & H_{1L^{Pos}}^{Pos} & H_{11}^{Neg} & \cdots & H_{1L^{Neg}}^{Neg} \\ H_{21}^{Pos} & H_{22}^{Pos} & \ddots & H_{2L^{Pos}}^{Pos} & H_{21}^{Neg} & \ddots & H_{2L^{Neg}}^{Neg} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_{N1}^{Pos} & H_{N2}^{Pos} & \ddots & H_{NL^{Pos}}^{Pos} & H_{N1}^{Neg} & \ddots & H_{NL^{Neg}}^{Neg} \end{bmatrix}_{N \times L} \quad (19)$$

The output of the CCCS-PIFELM approach can be determined by

$$Z = f\left(H\beta + B^{o}I\right) = f\left(\begin{bmatrix} H \\ I \end{bmatrix}\begin{bmatrix} \beta & B^{o} \end{bmatrix}\right) = f\left(\begin{bmatrix} H \\ I \end{bmatrix}\omega\right) \quad (20)$$

where $I = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}_{N \times 1}$ and $\omega = \begin{bmatrix} \beta & B^{o} \end{bmatrix}_{(L+1) \times n}$.

The output transfer function of the CCCS-PIFELM approach is linear, so the biases of output layer B^{o} is given a value of zero. The output of the CCCS-PIFELM approach can be written as

$$Z = H\beta . \quad (21)$$

In order to reduce the computational load, the output weight β is updated using an inverse-free algorithm as the size of hidden neurons increases. The output weight calculated from the inverse-free algorithm is equivalent to the solution of the benchmark ELM algorithm using the inverse operation. From the concepts in [39], U^{L} is the inverse-free recursive of the matrix H for the CCCS-PIFELM approach. Thus, the output weight of the CCCS-PIFELM approach can be calculated by

$$\beta = U^{L} Z \quad (22)$$

$$U^{L} = \begin{bmatrix} U_1^{L} & U_2^{L} \end{bmatrix}_{L \times N} \quad (23)$$

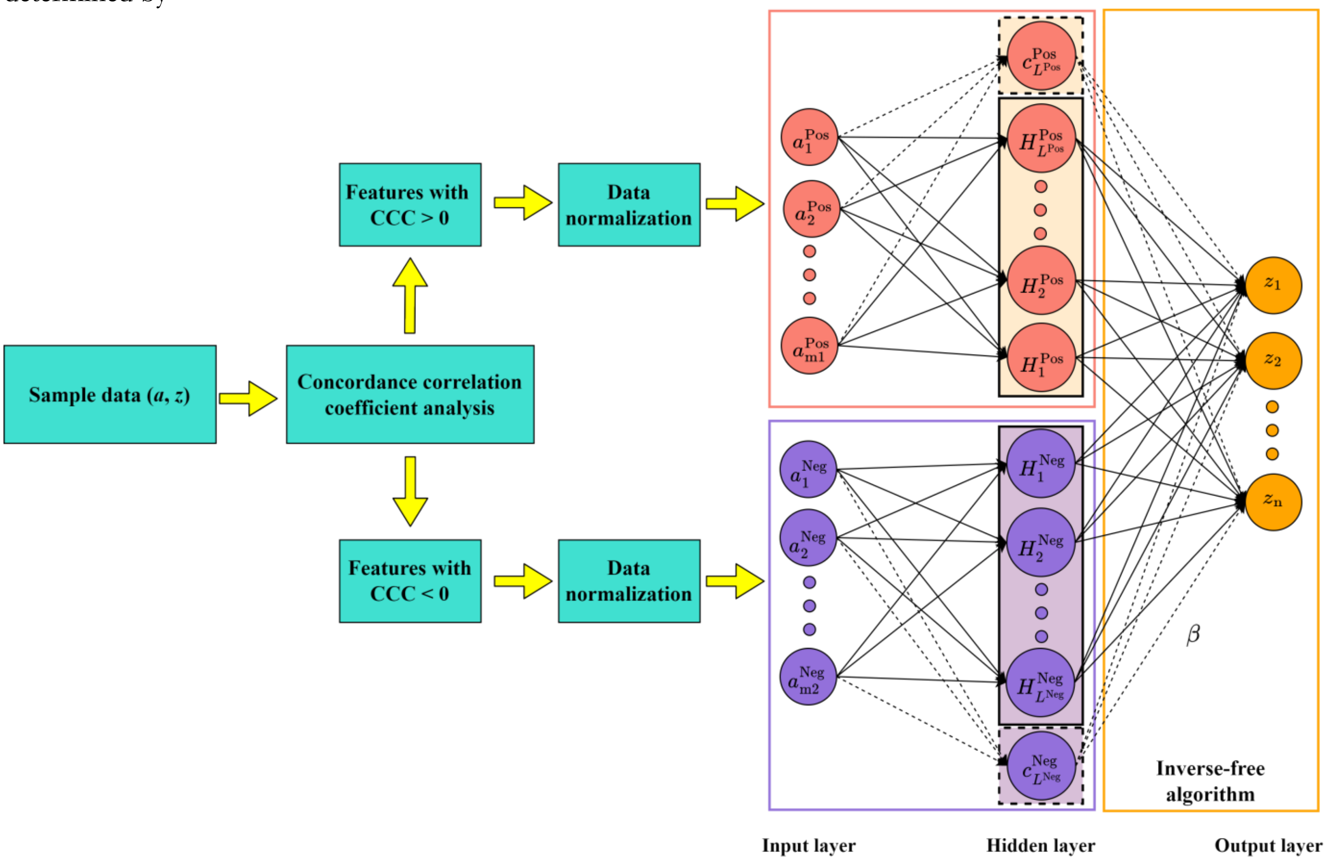

Fig. 1. Soft sensor based on concordance correlation coefficient subsets integrated with parallel inverse-free extreme learning machine (CCCS-PIFELM).

$$U_1^L = \frac{c^T cI - cc^T}{c^T c}\left(\frac{UHcc^T U}{c^T c - c^T UHc} + U\right) \quad (24)$$

$$U_2^L = -\frac{U_1^L Hc}{c^T c} + \frac{c}{c^T c}. \quad (25)$$

where $c \in R^L = \left[c_{L^{Pos}}^{Pos} \ c_{L^{Neg}}^{Neg}\right]$ is the extra hidden layer output vector.

The CCCS-PIFELM approach is compared to the PELM and IFELM approaches in terms of prediction performance. As performance indicators, the mean absolute error (MAE) and root mean squared error (RMSE) are used as follows:

$$\text{MAE} = \frac{1}{N_{tst}}\sum_{1}^{N_{tst}}\left|z_p - z'_p\right| \quad (26)$$

$$\text{RMSE} = \sqrt{\frac{1}{N_{tst}}\sum_{p=1}^{N_{tst}}\left(z_p - z'_p\right)^2} \quad (27)$$

where N_{tst} denotes the test sample size, z_p denotes the observed value and z'_p denotes the predicted value.

4. Case Study

The performance of the CCCS-PIFELM approach is evaluated in this section. Simulations are performed on a computer with an AMD Ryzen 5 2500U (2.0 GHz) processor using MATLAB 2022.

4.1. TE Benchmark Process

Tennessee Eastman (TE) benchmark process has been proposed by the Eastman Chemical Company to test the effectiveness of various algorithms [34]. The graphical representation of TE benchmark process is presented in Fig. 2. In this process, the reactor, stripping column, condenser, compressor, and separating column are the main operating units. Reactants A, C, D, and E are fed into the process to produce the liquid products G and H, as well as the by-product F. As shown in Table 1, 22 variables obtained from continuous process measurements are used as inputs in order to predict the product compositions (mol%) of D, E, F, G, and H. Datasets of TE benchmark process with 393 samples are collected from the database [40].

4.2. Developing the CCCS-PIFELM Approach for Predicting Outputs of TE Benchmark Process

As shown in Fig. 3, the implementation of the proposed CCCS-PIFELM approach can be summarized as follows:

1) Analysis of effects of input variables using CCC

For each input variable, the CCC value is calculated using Eq. (15). In accordance with the CCC values, input variables are classified into two subsets which are variables with positive and negative CCC values. The determination of CCC values allows for the consideration of the impacts of input variables on output variables.

2) Normalization and preparation of sample data

All sample data are normalized using Eq. (16).

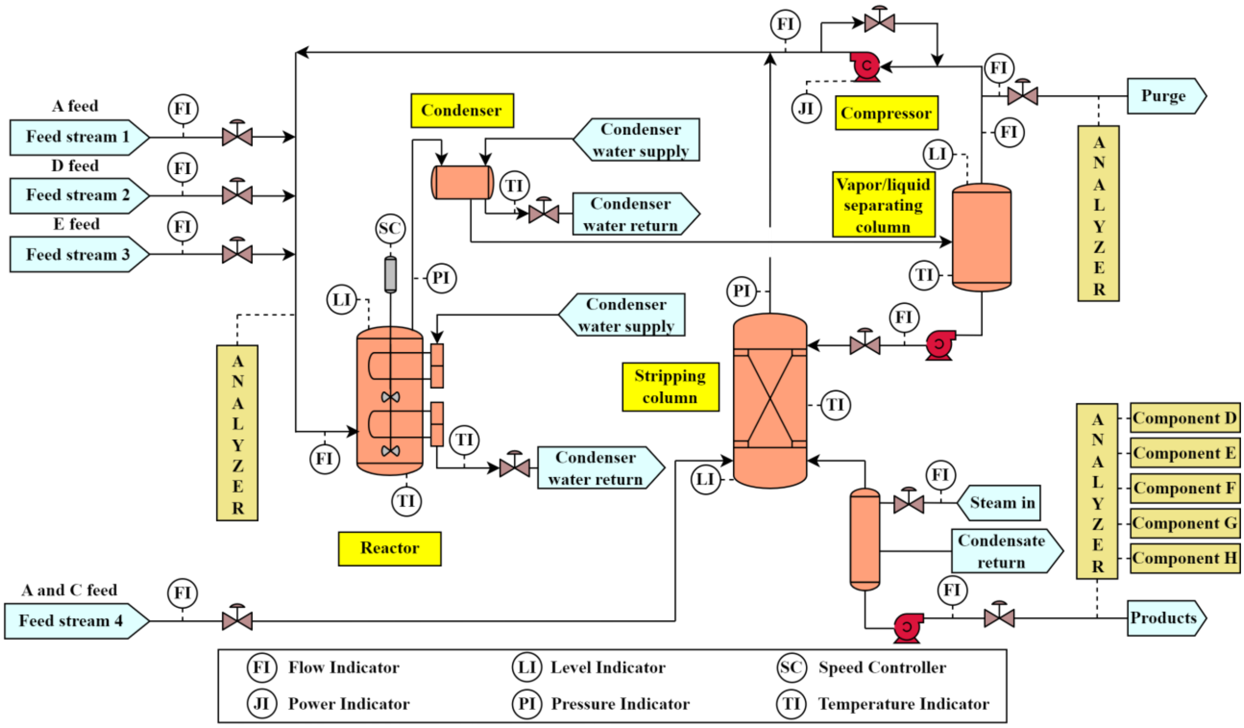

Fig. 2. Graphical representation of TE benchmark process.

DOI:10.4186/ej.2023.27.6.25

Table 1. Input variables and their corresponding ranges of TE benchmark process.

Descriptions of input variables [min; max; mean]			
1	A feed flow rate (kscmh) [0.004; 0.373; 0.229]	12	Percentage of product separating column level (%) [24.653; 72.488; 53.475]
2	D feed flow rate (kg/h) [3,025; 3,888; 3,452]	13	Pressure in product separating column (kPa gauge) [2,333; 2,693; 2,531]
3	E feed flow rate (kg/h) [3,936; 5430; 4,648]	14	Outflow of product separating column (m^3/h) [20.699; 31.118; 25.467]
4	A and C feed flow rate (kscmh) [8.103; 10.242; 9.110]	15	Percentage of stripping column level (%) [0; 107.951; 50.1153]
5	Recycle flow rate (kscmh) [24.914; 30.992; 27.435]	16	Pressure in stripping column (kPa gauge) [2,877; 3,330; 3,116]
6	Reactor feed flow rate (kscmh) [39.566; 46.912; 42.564]	17	Outflow of stripping column (m^3/h) [19.801; 24.964; 22.319]
7	Pressure in reactor (kPa gauge) [2,418; 2,789; 2,621]	18	Temperature of stripping column (^oC) [43.345; 78.644; 64.795]
8	Percentage of reactor level (%) [58.669; 76.225; 67.575]	19	Stripping column steam flow rate (kg/h) [0; 39.425; 15.734]
9	Temperature of reactor (^oC) [117; 128; 123]	20	Compressor power (kW) [214.200; 295.572; 247.235]
10	Purge flow rate (kscmh) [0; 0.801; 0.414]	21	Outlet temperature of reactor cooling water (^oC) [96.516; 110.258; 103.563]
11	Temperature of product separating column (^oC) [69.836; 106.597; 90.829]	22	Outlet temperature of separating column cooling water (^oC) [61.693; 100.798; 83.856]

Training, validation, and testing sets are classified in the proportions of 70%, 15%, and 15%, respectively. Training data are used for training the model and determining optimal output weights. The validation data are employed to prevent the over-fitting problem. Testing data are used to evaluate the generalization error in the model.

3) Preparation of two subsets

Two subsets of positive and negative CCC values are determined in accordance with two sets of input variables. Sigmoid functions are used as transfer functions in hidden neurons.

4) Training of the CCCS-PIFELM approach

Hidden layer outputs can be calculated using Eqs. (17), (18), and (19) after two subsets of positive and negative CCC values are created. The matrix of optimal output weight is determined using the inverse-free algorithm.

5) Performance testing of the CCCS-PIFELM approach

The outputs can be calculated using Eqs. (20) and (21). Testing data are used in the CCCS-PIFELM approach for evaluating the prediction performance by RMSE and MAE values.

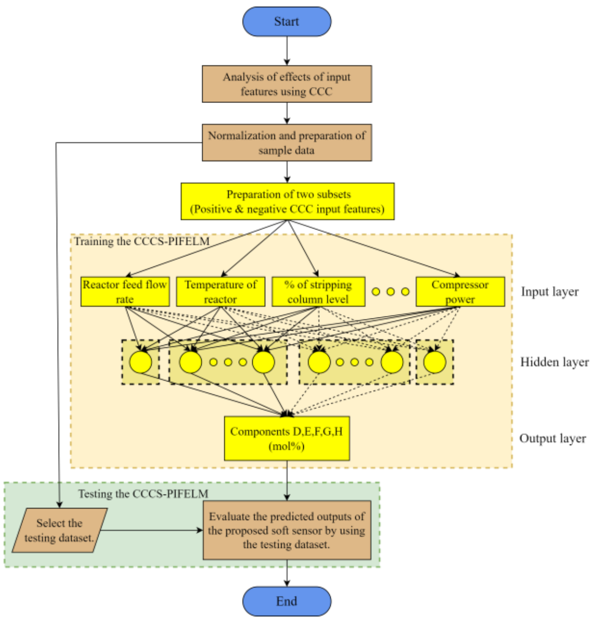

Fig. 3. Implementation of the CCCS-PIFELM approach for predicting outputs of TE benchmark process.

4.3. Concordance Correlation Coefficient Analysis of Input Features

The values of concordance correlation coefficient (CCC) can be determined using Eq. (15) for all input variables. If the CCC values are greater than zero, output variables are positively affected by input variables. If the CCC values are less than zero, output variables are adversely affected by input variables.

Results for the CCC analysis of input variables in TE benchmark process are shown in Table 2, in which input variables with positive and negative CCC effects are given.

4.4. Results and Discussion

In this part, the prediction accuracy of different approaches, including PELM, IFELM, and CCCS-PIFELM, is compared. The values of RMSE and MAE are used in the evaluation. The optimal size of neurons in hidden layer is adjusted by incrementally increasing the number of neurons with an interval of five [11]. The optimal parameters and response times of the PELM, IFELM, and CCCS-PIFELM approaches for the TE benchmark process are presented in Table 3.

According to Table 3, the CCCS-PIFELM approach requires the lowest number of weights and biases. In addition, it has the lowest response time compared to other approaches. Figure 4 shows the predicted product compositions using the PELM, IFELM, and CCCS-PIFELM approaches. It can be observed that the product compositions predicted by the proposed CCCS-PIFELM approach agree well with the actual data. The proposed CCCS-PIFELM approach has high prediction accuracy and a low response time. These features are important for efficient operations in chemical processes.

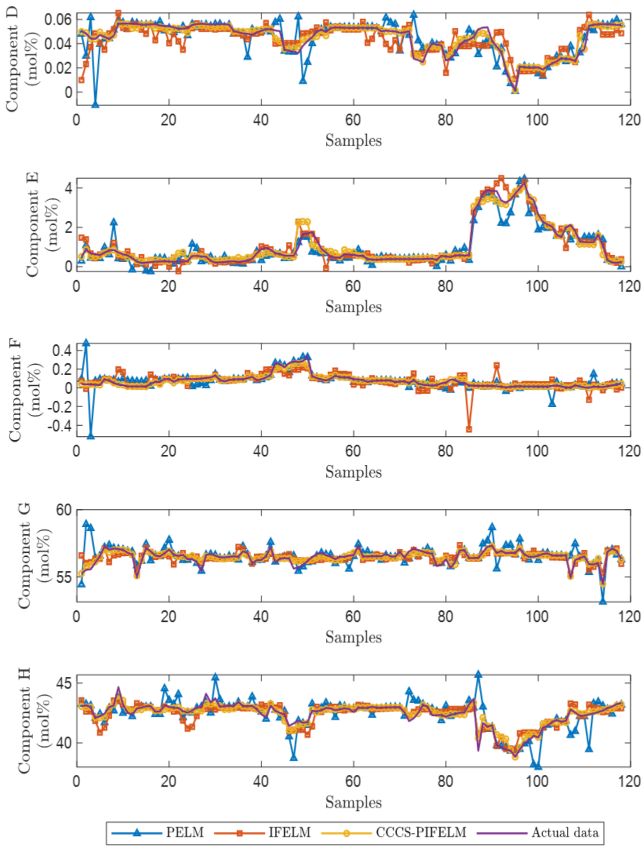

Fig. 4. Predicted product compositions using the PELM, IFELM, and CCCS-PIFELM approaches for TE benchmark process.

Table 2. Results for the CCC analysis of the input variables in TE benchmark process.

Product compositions	Input number with positive effect	Input number with negative effect
D	1,2,4,5,6,7,9,11,13,16,17,18,20,21,22	3,8,10,12,14,15,19
E	2,3,4,10,12,14,15,17,19	1,5,6,7,8,9,11,13,16,18,20,21,22
F	1,2,3,4,6,7,9,11,12,13,14,15,16,17,18,21,22	5,8,10,19,20
G	1,2,3,4,5,6,7,8,10,13,16,19,20	9,11,12,14,15,17,18,21,22
H	1,5,8,9,11,15,16,18,20,21,22	2,3,4,6,7,10,12,13,14,17,19

Table 3. Optimal parameters and response times of the PELM, IFELM, and CCCS-PIFELM approaches.

Components	PELM		IFELM		CCCS-PIFELM	
	Number of weights and biases	Response time (s)	Number of weights and biases	Response time (s)	Number of weights and biases	Response time (s)
D	(1610, 70)	0.0654	(1035, 45)	0.0602	(440, 35)	0.0510
E	(1610, 70)	0.0651	(1035, 45)	0.0605	(410, 35)	0.0504
F	(1725, 75)	0.0743	(1150, 50)	0.0723	(480, 40)	0.0567
G	(1725, 75)	0.0740	(1150, 50)	0.0726	(480, 40)	0.0563
H	(1725, 75)	0.0745	(1150, 50)	0.0722	(480, 40)	0.0567

DOI:10.4186/ej.2023.27.6.25

The comparisons of scatter plots for different approaches are presented in Fig. 5. The prediction accuracy is higher as the predicted data are distributed closer to the diagonal line. It can be observed that the predicted data of the proposed CCCS-PIFELM approach are quite close to the diagonal line, indicating that it has a better prediction performance compared to other approaches.

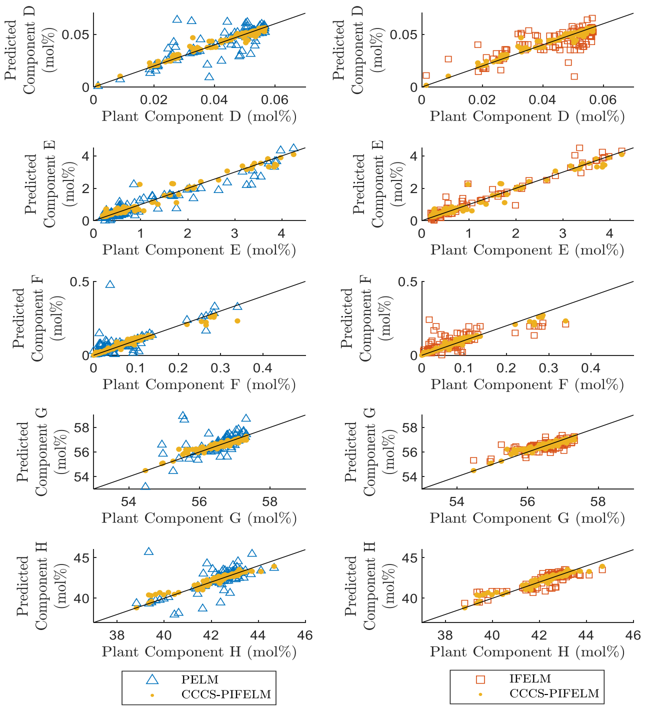

Fig. 5. Scatter plots for predicting product compositions.

Figures 6 and 7 show the values of RMSE and MAE, respectively, for the prediction of product compositions. The RMSE and MAE values are used to evaluate variations of errors in the prediction set. The RMSE values are usually greater than or equal to the MAE values. The variance of the errors increases as the differences between the values of RMSE and MAE increase. The results in Figs. 6 and 7 show that the proposed CCCS-PIFELM approach can give lower values of RMSE and MAE than other approaches.

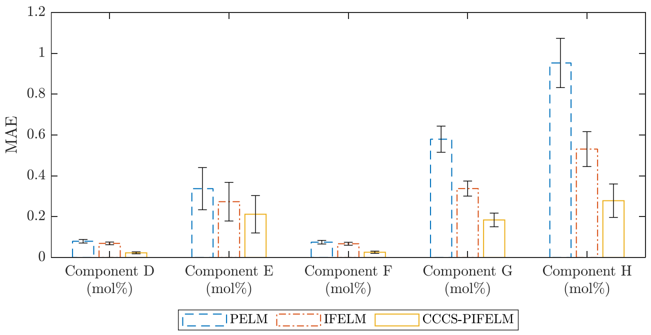

Fig. 6. RMSE values for predicting product compositions.

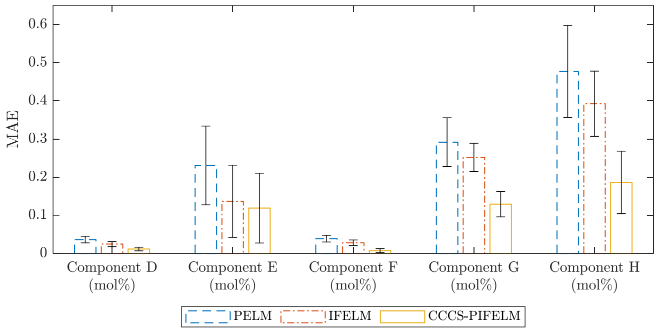

Fig. 7. MAE values for predicting product compositions.

Box plots of relative prediction errors are presented in Fig. 8. The band within each box represents the median value of relative prediction errors. The top and base surfaces of each box are the upper and lower quartiles, respectively. The length between top and base surfaces is the interquartile range (IQR) in which upper and lower quartiles serve as 0.75 and 0.25 quantiles, respectively. Red plus signs above and below the box indicate outliers, whose values are higher than 1.5 times the IQR. Whiskers are lines that lengthen above and below each box. One whisker relates the upper quartile to the highest nonoutlier point, and the other relates the lower quartile to the lowest nonoutlier point. The narrowest box ranges of the CCCS-PIFELM approach, as demonstrated in Fig. 8, indicate that it has been shown to have the highest prediction performance.

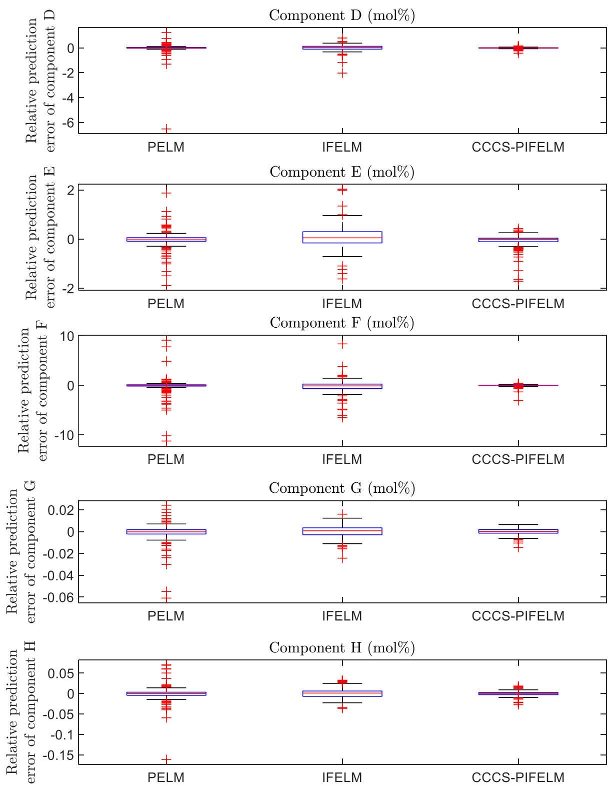

Fig. 8. Box plots of relative prediction errors.

5. Conclusions

For multivariate chemical processes, a soft sensor based on concordance correlation coefficient subsets integrated with parallel inverse-free extreme learning machine (CCCS-PIFELM) is proposed in this study. The analysis of the concordance correlation coefficient is performed to classify input variables into two subsets for considering their impacts on output variables. The inverse-free operation is applied to reduce the computational load. The CCCS-PIFELM approach can obtain better prediction accuracy compared to other approaches. Hence, the CCCS-PIFELM approach can be used as a powerful tool for the prediction of output variables in multivariate chemical processes.

Acknowledgement

This research project is supported by Mahidol University (Fundamental Fund: fiscal year 2023 by National Science Research and Innovation Fund (NSRF)). We also extend our gratitude to Mahidol University for providing the 60th Year Supreme Reign of His Majesty King Bhumibol Adulyadej Scholarship for the Ph.D. thesis of Mr. Thirasit Kusolsongtawee.

References

[1] W. Shao, Z. Ge, and Z. Song, "Semi-supervised mixture of latent factor analysis models with application to online key variable estimation," *Control Eng. Pract.*, vol. 84, pp. 32–47, 2019. [Online]. Available: https://doi.org/10.1016/j.conengprac.2018.11.008

[2] P. Bumroongsri and S. Kheawhom, "An interpolation-based robust MPC algorithm using polyhedral invariant sets," in *The 2013 European Control Conference, ECC 2013*, Zurich, Switzerland, 2013, pp. 3161-3166. [Online]. Available: https://doi.org/10.23919/ECC.2013.6669124

[3] P. Mohagheghi, W. Gilani, A. Stefanescu, and M. A. Fernandez, "An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases," *Empir. Softw. Eng.*, vol. 18, no. 1, pp. 89–116, Jan. 2013. [Online]. Available: https://doi.org/10.1007/s10664-012-9196-x

[4] P. Bumroongsri and S. Kheawhom, "An off-line formulation of tube-based robust MPC using polyhedral invariant sets," *Chem. Eng. Commun.*, vol. 203, no. 6, pp. 736-745, Jun. 2016. https://doi.org/10.1080/00986445.2015.1089402

[5] M. A. Hosen, A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction interval-based neural network modelling of polystyrene polymerization reactor - A new perspective of data-based modelling," *Chem. Eng. Res. Des.*, vol. 92, no. 11, pp. 2041–2051, Feb. 2014. [Online]. Available: https://doi.org/10.1016/j.cherd.2014.02.016

[6] A. Alblawi, "Fault diagnosis of an industrial gas turbine based on the thermodynamic model coupled with a multi feedforward artificial neural networks," *Energy Reports*, vol. 6, pp. 1083–1096, Apr. 2020. [Online]. Available: https://doi.org/10.1016/j.egyr.2020.04.029

[7] F. A. Essa, M. Abd Elaziz, and A. H. Elsheikh, "Prediction of power consumption and water productivity of seawater greenhouse system using random vector functional link network integrated with artificial ecosystem-based optimization," *Process Saf. Environ. Prot.*, vol. 144, pp. 322–329, Jul. 2020. [Online]. Available:https://doi.org/10.1016/j.psep.2020.07.044

[8] H. Wang and L. A. Ricardez-Sandoval, "Dynamic optimization of a pilot-scale entrained-flow gasifier using artificial recurrent neural networks," *Fuel*, vol. 272, Apr. 2020, Art. no. 117731. [Online]. Available: https://doi.org/10.1016/j.fuel.2020.117731

[9] N. Paeedeh, K. Ghiasi-Shirazi, "Improving the backpropagation algorithm with consequentialism weight updates over mini-batches," *Neurocomputing*, vol. 461, pp. 86–98, Oct. 2021. [Online]. Available: https://doi.org/10.1016/j.neucom.2021.07.010

[10] M. A. Mujtaba et al., "Ultrasound-assisted process optimization and tribological characteristics of biodiesel from palm-sesame oil via response surface methodology and extreme learning machine - Cuckoo search," *Renew. Energy*, vol. 158, pp. 202–214, May 2020. [Online]. Available: https://doi.org/10.1016/j.renene.2020.05.158

[11] Y. L. He, Z. Q. Geng, Y. Xu, and Q. X. Zhu, "A hierarchical structure of extreme learning machine (HELM) for high-dimensional datasets with noise," *Neurocomputing*, vol. 128, pp. 407–414, Aug. 2014. https://doi.org/10.1016/j.neucom.2013.08.024

[12] L. Lin, F. Wang, X. Xie, and S. Zhong, "Random forests-based extreme learning machine ensemble for multi-regime time series prediction," *Expert Syst. Appl.*, vol. 83, pp. 164–176, Apr. 2017. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.04.013

[13] K. Samiee, A. Iosifidis, and M. Gabbouj, "On the comparison of random and Hebbian weights for the training of single-hidden layer feedforward neural networks," *Expert Syst. Appl.*, vol. 83, pp. 177–186, Apr. 2017. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.04.025

[14] Y. Lan, Z. Hu, Y. C. Soh, and G. Bin Huang, "An extreme learning machine approach for speaker recognition," *Neural Comput. Appl.*, vol. 22, no. 3–4, pp. 417–425, Apr. 2013. [Online]. Available: https://doi.org/10.1007/s00521-012-0946-x

[15] Y. Lan, Y. C. Soh, and G. Bin Huang, "Constructive hidden nodes selection of extreme learning machine for regression," *Neurocomputing*, vol. 73, pp. 3191–3199, May. 2010. [Online]. Available: https://doi.org/10.1016/j.neucom.2010.05.022

[16] J. Peng, L. Li, and Y. Y. Tang, "Combination of activation functions in extreme learning machines for multivariate calibration," *Chemom. Intell. Lab. Syst.*, vol. 120, pp. 53–58, Nov. 2013. [Online]. Available: https://doi.org/10.1016/j.chemolab.2012.11.004

[17] Q. Yu et al., "Ensemble delta test-extreme learning machine (DT-ELM) for regression," *Neurocomputing*, vol. 129, pp. 153–158, Aug. 2014. [Online]. Available: https://doi.org/10.1016/j.neucom.2013.08.041

[18] W. Zheng, X. Fu, and Y. Ying, "Spectroscopy-based food classification with extreme learning machine," *Chemom. Intell. Lab. Syst.*, vol. 139, pp. 42–47, Sep. 2014. [Online]. Available: https://doi.org/10.1016/j.chemolab.2014.09.015

[19] A. Law and A. Ghosh, "Multi-label classification using a cascade of stacked autoencoder and extreme learning machines," *Neurocomputing*, vol. 358, pp. 222–234, May 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.05.051

[20] X. Wen, "Modeling and performance evaluation of wind turbine based on ant colony optimization-extreme learning machine," *Appl. Soft Comput. J.*, vol. 94, Jun. 2020, Art. no. 106476. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.106476

[21] M. Larrea, A. Porto, E. Irigoyen, A. J. Barragán, and J. M. Andújar, "Extreme learning machine ensemble model for time series forecasting boosted by PSO: Application to an electric consumption problem," *Neurocomputing*, vol. 452, 465-472, Sep. 2021. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.12.140

[22] Q. Li, X. Zhang, T. Ma, C. Jiao, H. Wang, and W. Hu, "A multi-step ahead photovoltaic power prediction model based on similar day, enhanced colliding bodies optimization, variational mode decomposition, and deep extreme learning machine," *Energy*, vol. 224, Jun. 2021, Art. no. 120094. [Online]. Available: https://doi.org/10.1016/j.energy.2021.120094

[23] Y. Lei, H. R. Karimi, L. Cen, X. Chen, and Y. Xie, "Processes soft modeling based on stacked autoencoders and wavelet extreme learning machine for aluminum plant-wide application," *Control Eng. Pract.*, vol. 108, Mar. 2021, Art. no. 104706. [Online]. Available: https://doi.org/10.1016/j.conengprac.2020.104706

[24] Y. Zhou, N. Zhou, L. Gong, and M. Jiang, "Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine," *Energy*, vol. 204, Aug. 2020, Art. no. 117894. [Online]. Available: https://doi.org/10.1016/j.energy.2020.117894

[25] Z. F. Liu, L. L. Li, M. L. Tseng, and M. K. Lim, "Prediction short-term photovoltaic power using improved chicken swarm optimizer - Extreme learning machine model," *J. Clean. Prod.*, vol. 248, Mar. 2020, Art. no. 119272. [Online]. Available: https://doi.org/10.1016/j.jclepro.2019.119272

[26] Y. Wang, J. Liu, and Y. Han, "Production capacity prediction of hydropower industries for energy optimization: Evidence based on novel extreme learning machine integrating Monte Carlo," *J. Clean. Prod.*, vol. 272, Nov. 2020, Art. no. 122824. [Online]. Available: https://doi.org/10.1016/j.jclepro.2020.122824

[27] W. Cao and Q. Yang, "Online sequential extreme learning machine based adaptive control for wastewater treatment plant," *Neurocomputing*, vol. 408, pp. 169–175, Sep. 2020. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.05.109

[28] X. H. Gao, K. I. Wong, P. K. Wong, and C. M. Vong, "Adaptive control of rapidly time-varying discrete-time system using initial-training-free online extreme learning machine," *Neurocomputing*, vol. 194, pp. 117–125, Jun. 2016. [Online]. Available: https://doi.org/10.1016/j.neucom.2016.01.071

[29] W. Shao, Z. Ge, Z. Song, and K. Wang, "Nonlinear industrial soft sensor development based on semi-supervised probabilistic mixture of extreme learning machines," *Control Eng. Pract.,* vol. 91, Oct. 2019, Art. no. 104098. [Online]. Available: https://doi.org/10.1016/j.conengprac.2019.07.016

[30] M. Zhang, X. Liu, and Z. Zhang, "A soft sensor for industrial melt index prediction based on evolutionary extreme learning machine," *Chinese J. Chem. Eng.*, vol. 24, no. 8, pp. 1013–1019, Aug. 2016. https://doi.org/10.1016/j.cjche.2016.05.030

[31] P. Bumroongsri and S. Kheawhom, "The polyhedral off-line robust model predictive control strategy for uncertain polytopic discrete-time systems," *IFAC Proc. Vol.*, vol. 45, pp. 655-660, July 2012. [Online]. Available: https://doi.org/10.3182/20120710-4-SG-2026.00017

[32] Y. L. He, Z. Q. Geng, and Q. X. Zhu, "A data-attribute-space-oriented double parallel (DASODP) structure for enhancing extreme learning machine: Applications to regression datasets," *Eng. Appl. Artif. Intell.*, vol. 41, pp. 65–74, May 2015. [Online]. Available:https://doi.org/10.1016/j.engappai.2015.02.001

[33] Y. L. He, Z. Q. Geng, and Q. X. Zhu, "Positive and negative correlation input attributes oriented subnets based double parallel extreme learning machine (PNIAOS-DPELM) and its application to monitoring chemical processes in steady state," *Neurocomputing*, vol. 165, pp. 171–181, Oct. 2015. [Online]. Available:https://doi.org/10.1016/j.neucom.2015.03.007

[34] Y. L. He, Z. Q. Geng, and Q. X. Zhu, "Data driven soft sensor development for complex chemical processes using extreme learning machine," *Chem. Eng. Res. Des.*, vol. 102, pp. 1–11, Oct. 2015. [Online]. Available: https://doi.org/10.1016/j.cherd.2015.06.009

[35] Z. Gao, C. Ma, J. Zhang, and W. Xu, "Enhanced online sequential parallel extreme learning machine

and its application in remaining useful life prediction of integrated modular avionics," *IEEE Access*, vol. 7, pp. 183479–183488, Dec. 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2960406

[36] X. Li, J. Liu, and P. Niu, "Least square parallel extreme learning machine for modeling NOx emission of a 300MW circulating fluidized bed boiler," *IEEE Access*, vol. 8, pp. 79619–79636, Apr. 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2990440

[37] L. Guo, J. hua Hao, and M. Liu, "An incremental extreme learning machine for online sequential learning problems," *Neurocomputing*, vol. 128, pp. 50–58, Mar. 2014. [Online]. Available: https://doi.org/10.1016/j.neucom.2013.03.055

[38] S. Kheawhom and P. Bumroongsri, "Constrained robust model predictive control based on polyhedral invariant sets by off-line optimization," *Chem. Engineer. Trans.*, vol. 32, pp. 1417-1422, Jun. 2013. [Online]. Available: https://www.aidic.it/cet/13/32/237.pdf https://doi.org/10.3303/CET1332237

[39] S. Li, Z. H. You, H. Guo, X. Luo, and Z. Q. Zhao, "Inverse-free extreme learning machine with optimal information updating," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1229–1241, Jun. 2016. [Online]. Available: https://doi.org/10.1109/TCYB.2015.2434841

[40] X. Chen. (2019). *Tennessee Eastman simulation dataset* Accessed: 1 January 2021. [Online]. Available: https://dx.doi.org/10.21227/4519-z502

Thirasit Kusolsongtawee received the Bachelor of Engineering and Master of Engineering (Chemical Engineering) from Mahidol University in 2014 and 2016, respectively. He is in the process of obtaining his Ph.D. in Integrated Chemical Engineering at Mahidol University. His area of research interest lies in process simulation and data-driven modeling for chemical processes.

Soorathep Kheawhom obtained the Bachelor of Engineering in Chemical Engineering from Chulalongkorn University in 1997. He completed his Master's and Ph.D. Degrees under the mentorship of Professor Masahiko Hirao at the University of Tokyo. He received his Ph.D. in 2004 after spending five years in Japan. He is currently working as an associate professor in the Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University in Bangkok, Thailand.

Sorin Olaru is currently working as a professor of CentraleSupélec and a member of the CNRS Laboratory of Signals and Systems, both part of the Paris-Saclay University in France. He is engaged in researching numerical methods in optimization and control, set-theoretic characterization of constrained dynamical systems and optimization-based control design. His current research projects include network control (time-delay) systems, fault-tolerant control and embedded predictive control.

Pornchai Bumroongsri obtained Bachelor's, Master's and Doctoral degrees in Chemical Engineering from Chulalongkorn University in 2008, 2009 and 2012, respectively. He currently holds the position of assistant professor in the Department of Chemical Engineering, Faculty of Engineering, Mahidol University. His research focus is on developing robust model predictive control, optimization technology and artificial intelligence for industrial chemical processes.