

Article

Pseudo-Labeling Optimization Based Ensemble Semi-Supervised Soft Sensor in the Process Industry

Youwei Li ^{1,2}, Huaiping Jin ^{1,2,*} , Shoulong Dong ³, Biao Yang ^{1,2} and Xiangguang Chen ³

¹ Yunnan Key Laboratory of Computer Technologies Application, Kunming 650500, China; 20192104045@stu.kust.edu.cn (Y.L.); biaoymust@kust.edu.cn (B.Y.)

² Department of Automation, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

³ Department of Chemical Engineering, School of Chemistry and Chemical Engineering, Beijing Institute of Technology, Beijing 100081, China; sldong@bit.edu.cn (S.D.); xgc1@bit.edu.cn (X.C.)

* Correspondence: jinhuaiping@kust.edu.cn; Tel.: +86-15877986943

Abstract: Nowadays, soft sensor techniques have become promising solutions for enabling real-time estimation of difficult-to-measure quality variables in industrial processes. However, labeled data are often scarce in many real-world applications, which poses a significant challenge when building accurate soft sensor models. Therefore, this paper proposes a novel semi-supervised soft sensor method, referred to as ensemble semi-supervised negative correlation learning extreme learning machine (EnSSNCLELM), for industrial processes with limited labeled data. First, an improved supervised regression algorithm called NCLELM is developed, by integrating the philosophy of negative correlation learning into extreme learning machine (ELM). Then, with NCLELM as the base learning technique, a multi-learner pseudo-labeling optimization approach is proposed, by converting the estimation of pseudo labels as an explicit optimization problem, in order to obtain high-confidence pseudo-labeled data. Furthermore, a set of diverse semi-supervised NCLELM models (SSNCLELM) are developed from different enlarged labeled sets, which are obtained by combining the labeled and pseudo-labeled training data. Finally, those SSNCLELM models whose prediction accuracies were not worse than their supervised counterparts were combined using a stacking strategy. The proposed method can not only exploit both labeled and unlabeled data, but also combine the merits of semi-supervised and ensemble learning paradigms, thereby providing superior predictions over traditional supervised and semi-supervised soft sensor methods. The effectiveness and superiority of the proposed method were demonstrated through two chemical applications.

Keywords: soft sensor; unlabeled data; label scarcity; semi-supervised learning; ensemble learning; pseudo labeling; evolutionary optimization; negative correlation learning; extreme learning machine



Citation: Li, Y.; Jin, H.; Dong, S.; Yang, B.; Chen, X. Pseudo-Labeling Optimization Based Ensemble Semi-Supervised Soft Sensor in the Process Industry. *Sensors* **2021**, *21*, 8471. <https://doi.org/10.3390/s21248471>

Academic Editor: János Abonyi

Received: 16 November 2021

Accepted: 16 December 2021

Published: 19 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern industrial processes are equipped with a large number of measurement devices, in order to allow the implementation of advanced monitoring, optimization, and control of the production process. However, many crucial quality variables in industrial process are difficult to measure online, due to the lack of reliable hardware sensors or the high investment in the purchase and maintenance of apparatuses. To tackle this problem, soft sensor technology, as a promising indirect measurement tool, has been proposed, to enable real-time estimations of difficult-to-measure process variables [1,2]. The basis of a soft sensor is to build a mathematical model describing the relationship between the difficult-to-measure target variable and the easy-to-measure secondary variables, and then perform online estimation for the query data, based on the built predictive model. Generally, soft sensors can be divided into two categories: first principle, and data-driven methods. The former method type requires deep physical and chemical knowledge, which is often impossible in many real-world applications. Alternatively, data-driven methods,

only relying on historical process information and data analytics techniques, have gained great popularity in soft sensor applications over the past two decades [3–8].

In the context of data-driven soft sensor modeling, having sufficient process data is of great importance to guarantee the generalization capability of the model. Typically, a sample for model development can be split into two parts, i.e., attributes (also termed as inputs) and label (also termed as output), which correspond to the easily obtained process variables and the target variable to be estimated, respectively. Thus, the modeling samples with labels are called labeled data, while those lacking labels are called unlabeled data. However, the insufficiency of data information is very common in practical applications, which makes it difficult to establish an accurate data-driven model. Roughly speaking, there are two common data deficiency issues: (i) both input and output information are insufficient, and (ii) unlabeled data are sufficient, but labeled data are scarce. One popular approach to deal with the first scenario is virtual sample generation (VSG) [9,10] methods, which are applied to create virtual samples, using the original small data set. Another solution to this problem is transfer learning [11–13], which enables improving the model performance by sharing the relevant information from other domains with rich data. As for the second scenario, active learning [14,15] has been employed to provide the labels for the most informative unlabeled data, using an oracle such as a human annotator; thus, enlarging the labeled modeling data. Alternatively, semi-supervised learning (SSL) [16–20] has been widely used to enhance model prediction performance, by leveraging both labeled and unlabeled data. In practice, there are usually abundant unlabeled data, but limited labeled data in industrial processes, which has been the main bottleneck of building accurate soft sensor models. Thus, this paper focuses on semi-supervised soft sensor development.

As an important branch of machine learning, SSL, has undergone a long history and has become a hot topic [16–20]. SSL algorithms can effectively exploit information from both labeled and unlabeled data to build models whose accuracies are usually better than those obtained from only using the limited labeled data. So far, five typical classes of SSL algorithms have emerged [16]: generative models, semi-supervised support vector machines (S3VM), graph-based methods, self-training, and co-training. However, many of the current SSL methods mainly focus on the prediction of discrete variables in classification problems, such as imaging analysis and processing, natural language processing, and speech recognition [17]. Unfortunately, it is difficult or impossible to transfer these methods to the regression problems directly, as is often the case in soft sensor applications, which mainly concerns predictions of continuous target variables. Hence, research on semi-supervised soft sensors has gained considerable attention in recent years. A detailed review and discussions on this topic can be found in reference [18]. Here, we only outline some representative research:

- (1) Probabilistic generative models: This type of method aims to improve the estimation accuracy of the underlying distribution of the input space by introducing abundant unlabeled data. These methods assume that all samples, labeled and unlabeled, are generated from the same underlying model. The main difference among the different generative methods lies in the underlying assumptions. For example, Ge and Song [21] proposed a semi-supervised principal component regression (PCR) model, using a probabilistic method. The method first formulates a generative model structure according to the traditional PCR model. Then, by assuming that both probability density functions of the principal component and the process noise are Gaussian, the optimal parameters with respect to the data distribution are optimized by maximizing a likelihood function using an expectation-maximum algorithm, where supervised and unsupervised objectives are involved. Similarly, a semi-supervised probabilistic partial least squares regression model was developed for soft sensor modeling [22]. However, the assumption of Gaussian distribution is often ill-conditioned for industrial processes with multiple modes, where the process data usually exhibit non-Gaussian characteristics. To tackle this issue, non-Gaussian distribution based

assumptions have been introduced to build generative soft sensor models, such as a mixture semi-supervised probabilistic PCR model [23], semi-supervised Gaussian mixture regression [24], semi-supervised Dirichlet process mixture of Gaussians [25], semi-supervised mixture of latent factor analysis models [26], and Student's-t mixture regression [27]. Overall, the key to building an accurate generative model lies in accurate model assumptions, which are often difficult to determine without sufficient reliable domain knowledge.

- (2) Graph based methods: Methods of this type are based on manifold assumptions and require constructing a semi-labeled graph to assure the label smoothness over the graph. To this end, one needs to define a graph where the nodes denote labeled and unlabeled samples, and where the edges connect two nodes if their corresponding samples are highly similar. One typical example of such method is the label propagation method [28], originally proposed for addressing classification problems. As for graph based SSL soft sensors, it is a common practice to embed the graph based regularization into the cost function based on traditional supervised regression techniques. For example, two semi-supervised soft sensors were developed, by integrating the extreme learning machine (ELM) method and the graph Laplacian regularization into a unified modeling framework for industrial Mooney viscosity prediction [29,30]. Similarly, Yan et al. [31] developed a semi-supervised Gaussian process regression (GPR) for quality prediction, by using a semi-supervised covariance function, which was defined by introducing the manifold information into the traditional covariance function. Moreover, to enhance the model performance for handling complex process characteristics, as well as exploiting unlabeled process data, Yao and Ge [32] proposed a semi-supervised deep learning model for soft sensor development. First, it implements unsupervised feature extraction through an autoencoder with a deep network structure. Then, ELM is utilized for regression, by introducing manifold regularization. In addition, Yan et al. [33] proposed a semi-supervised deep neural regression network with manifold embedding for soft sensor modeling.
- (3) Representation learning based methods: A general strategy for these methods is to use unlabeled data to assist in extracting abstract latent features of the input data. The most common techniques for this purpose are deep learning methods [34], such as convolutional neural networks, deep belief networks (DBN), long/short-term memory neural networks, and a large variety of autoencoders. Besides its strong representation ability, deep learning is inherently semi-supervised and, thus, can effectively exploit all available process data. As an early attempt, Shang et al. [35] employed a DBN to build soft sensors for estimating the heavy diesel 95% cut-point of a crude distillation unit (CDU). However, traditional representation techniques are mainly implemented in an unsupervised manner, where the output variable information is ignored. To address this issue, several research works have focused on exploring semi-supervised representation learning techniques. For instance, Yan et al. [36] proposed a deep relevant representation learning approach based on a stacked autoencoder, which conducts a mutual information analysis between the representations and the output variable in each layer. Similar research has also been reported in references [37–39].
- (4) Self-labeled methods [40]: The core of such methods is extending the labeled training set by adding high-confidence pseudo-labeled data. In such a modeling framework, one or more predictive models are first trained with labeled data only, and then refined using the extended labeled set through iterative learning. Two representative examples are self-training [41], and co-training [42], based on which some variants have also been proposed, such as COREG [43], Tri-training [44], Multi-train [45], and CoForest [46]. As an instantiation of self-training, a semi-supervised support vector regression model was proposed and verified using 30 regression datasets and an industrial semiconductor manufacturing dataset [47]. In the method, the label distribution of the unlabeled data is estimated with two probabilistic local reconstruction

models, thus providing the labeling confidence. Differently, based on the co-training paradigm, Bao et al. [48] proposed a co-training partial least squares (PLS) method for semi-supervised soft sensor development, by splitting the total process variables into two different parts, serving as two views. Instead of using single-output regression techniques, four semi-supervised multiple-output learning soft sensor models [49] were developed. In addition, by applying a spatial view, a temporal view, and a transformed view together, a multi-view transfer semi-supervised regression was developed, by combining transfer learning and co-training for air quality prediction [50]. Another example is a co-training style semi-supervised artificial neural network model for thermal conductivity prediction based on a disagreement-based semi-supervised learning principle [51]. The core of this method is constructing two artificial neural networks learners with different architectures, to label the unlabeled samples. Despite the simplicity and flexibility of implementation, the success of self-labeled SSL soft sensors heavily depends on the reliable confidence estimation of pseudo-labeled data.

According to the type of unlabeled data utilized, the above-mentioned semi-supervised modeling methods can be categorized into two groups: i.e., regularization embedding and pseudo labeling based approaches. The former methods aim to improve the model training, while the latter ones seek to enhance the model generalization capability by enlarging the labeled training data using high-confidence pseudo-labeled data. Compared to the regularization based SSL methods, pseudo labeling based ones have the merits of easy implementation and flexible wrapping with any base learning technique. Thus, in this work, we focus on pseudo-labeling semi-supervised soft sensor development. However, traditional pseudo-labeling SSL methods often encounter several drawbacks. One particular problem is the difficulty in defining confidence evaluation criterion. The most commonly used strategy is to evaluate the improvement rate of prediction performance after introducing the pseudo-labeled data [43]. Unfortunately, using such criterion cannot effectively characterize the complex hypothesis behind the labeled and unlabeled data. Another problem is that traditional pseudo-labeling strategies, such as self-training and co-training are prone to resulting in error propagation and accumulation along with the iterative learning. Thus, it is appealing to investigate the new paradigm of pseudo-labeling, to achieve reliable pseudo-label estimation.

In addition to augmenting the labeled training data, ensemble learning also plays a crucial role in enhancing the prediction performance of soft sensors [52–55]. Usually, ensemble methods can produce a strong ensemble model that is significantly more accurate than a single learner [56]. Although semi-supervised and ensemble learning approaches are two distinctive learning paradigms, they can be complementary to each other with efficient combination. On the one hand, the introduction of ensemble learning into SSL allows avoiding the difficulty in selecting model parameters and overcoming the drawback of single predictive model; thus, reducing the modeling uncertainty and improving the prediction reliability. On the other hand, the utilization of unlabeled data is helpful for improving the accuracy and diversity of base models, which is of great importance for constructing high-performance ensemble models. Therefore, the effective combination of semi-supervised and ensemble learning has attracted much attention from researchers on soft sensor development. For example, some research attempted to combine multiple homogeneous or heterogeneous base learners to construct semi-supervised ensemble models within the self-labeled learning frameworks (e.g., self-training and co-training) [50,51,57]. In addition, some scholars used unlabeled data to enhance the diversity of base models, thus improving the ensemble performance [29,58,59]. Moreover, Shao and Tian [60] proposed a soft sensor method based on a semi-supervised selective ensemble learning strategy. This method uses abundant unlabeled samples to help achieve reliable process state partition. These examples show that the combination of semi-supervised and ensemble learning is expected to improve the performance of a soft sensor model.

In light of the above-mentioned problems, in this work, a new semi-supervised soft sensor modeling method namely EnSSNCLELM was developed for high-quality prediction of industrial processes where the labeled data are limited but the unlabeled data are rich. The feasibility and superiority of the proposed method have been validated through application to a simulated fed-batch penicillin fermentation process and an industrial chlortetracycline fermentation process. Overall, the main contributions of this paper are threefold:

- (1) An improved supervised regression model NCLELM was developed and serves as the base learning technique for the proposed EnSSNCLELM modeling framework. Despite the fast training speed, ELM is prone to deliver unstable predictions, due to the random assignments of input weights and biases. By introducing the ensemble strategy of negative correlation learning into ELM, the NCLELM algorithm allows explicitly increasing the diversity among the base ELM models and, thus, enhancing the prediction accuracy and reliability.
- (2) A multi-learner pseudo-labeling optimization (MLPLO) approach is proposed to achieve pseudo-label estimation. Differently from traditional self-labeling techniques such as self-training and co-training, the MLPLO method attempts to build an explicit optimization problem with the unknown labels of the unlabeled data as the decision variables. Meanwhile, by exploring the inherent connections between labeled and unlabeled data, the individual and collaborative prediction performance of multiple learners are defined and integrated as the optimization objective. Then, an evolutionary optimization approach is adopted to solve the formulated pseudo-labeling optimization problem (PLOP), so as to obtain high-confidence pseudo-labeled samples for expanding the labeled set. A significant advantage of MLPLO is its strong capability for avoiding the error propagation and accumulation found with commonly used iterative learning.
- (3) By effectively combining a MLPLO strategy with ensemble modeling, the proposed EnSSNCLELM soft sensor method allows achieving the complementary advantages of semi-supervised and ensemble learning. On the one hand, semi-supervised learning is helpful for enhancing the accuracy and diversity of ensemble members by providing different high-confidence pseudo-labeled sets. On the other hand, combining multiple semi-supervised models using ensemble methods makes it possible to fully utilize the information of unlabeled data and reduce the modeling uncertainty caused by sub-optimal parameter settings and data selection.

The rest of the paper is organized as follows. Section 2 provides a brief introduction of ELM modeling and negative correlation learning (NCL). The proposed NCLELM and EnSSNCLELM soft sensor methods are elaborated in Section 3. Section 4 reports two case studies, to demonstrate the feasibility and efficiency of the proposed approach. Finally, conclusions are drawn in Section 5.

2. Preliminaries

2.1. Extreme Learning Machine

ELM [61] is a computationally efficient learning method for training single-hidden layer feedforward networks (SLFNs). Unlike traditional gradient-based iterative learning, the input weights and hidden node biases of a ELM model are randomly assigned, and then the output weights of SLFNs can be obtained analytically, which enables extremely fast training of ELM.

Given a training set $D = \{\mathbf{X}, \mathbf{y}\}$ with $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^L$ and $\mathbf{y} = \{y_i\}_{i=1}^L$ representing input and output data, respectively, where L is the number of training samples, and $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}^1$. Then, a standard ELM model with N_{node} hidden nodes and the activation function $g(\cdot)$ can be mathematically modeled as

$$\sum_{i=1}^{N_{\text{node}}} \beta_i g_i(\mathbf{x}_j) = \sum_{i=1}^{N_{\text{node}}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = o_j, \quad j = 1, \dots, L \quad (1)$$

where $\mathbf{w}_i = [w_{i,1}, w_{i,2}, \dots, w_{i,d}]^T$ is the input weight vector that connects the i th hidden node and the input nodes, β_i denotes the output weight that connects the i th hidden node and the output node, and b_i represents the bias of the i th hidden node. While numerous activation functions have been defined, a simple sigmoidal function $g(x) = 1/(1 + \exp(-x))$ is adopted in this study.

To train an ELM model, the fitting errors on the training data are expected to be zeros. In other words, the parameters, i.e., β_i , \mathbf{w}_i , and b_i , must satisfy the following equation:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{y} \quad (2)$$

where

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{N_{\text{node}}} \cdot \mathbf{x}_1 + b_{N_{\text{node}}}) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_L + b_1) & \cdots & g(\mathbf{w}_{N_{\text{node}}} \cdot \mathbf{x}_L + b_{N_{\text{node}}}) \end{bmatrix}_{L \times N_{\text{node}}} \quad (3)$$

$$\boldsymbol{\beta} = [\beta_1, \dots, \beta_{N_{\text{node}}}]^T, \mathbf{y} = [y_1, \dots, y_L]^T \quad (4)$$

Since the input weights and the hidden layer biases have been randomly generated, training an ELM model is equivalent to finding a least-squares solution $\hat{\boldsymbol{\beta}}$ for the linear system $\mathbf{H}\boldsymbol{\beta} = \mathbf{y}$:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{y} \quad (5)$$

where \mathbf{H}^\dagger is the Moore–Penrose generalized inverse of the hidden layer output matrix \mathbf{H} .

2.2. Negative Correlation Learning

NCL [62] is a learning approach originally proposed for neural network ensembles. It introduces a correlation penalty term into the cost function of each individual network in the ensemble, thus allowing explicitly maximizing the diversity among the base models. In this way, NCL can produce individual networks interactively on the same training set and the training errors of these models tend to be negatively correlated. Given a training set $\{\mathbf{x}_i, y_i\}_{i=1}^L$, NCL combines N neural networks $f_n(\mathbf{x}_i)$ to form an ensemble model:

$$f_{\text{ens}}(\mathbf{x}_i) = \frac{1}{N} \sum_{n=1}^N f_n(\mathbf{x}_i) \quad (6)$$

To train network f_n , the cost function e_n for network n is defined by:

$$e_n = \frac{1}{L} \sum_{i=1}^L (f_n(\mathbf{x}_i) - y_i)^2 + \lambda \frac{1}{L} p_n \quad (7)$$

where $\lambda \geq 0$ is a weighting parameter on the penalty term p_n :

$$p_n = \sum_{i=1}^L \left\{ (f_n(\mathbf{x}_i) - f_{\text{ens}}(\mathbf{x}_i)) \cdot \sum_{m \neq n} (f_m(\mathbf{x}_i) - f_{\text{ens}}(\mathbf{x}_i)) \right\} \\ = - \sum_{i=1}^L (f_n(\mathbf{x}_i) - f_{\text{ens}}(\mathbf{x}_i))^2 \quad (8)$$

where the first term is the empirical training error of the network, while the second is a correlation penalty term.

During the training process, as shown in Equations (7) and (8), all the individual networks interact with each other through their penalty terms in the cost function. Each network aims to minimize the difference between $f_n(\mathbf{x}_i)$ and y_i , whereas the difference between $f_{\text{ens}}(\mathbf{x}_i)$ and $f_n(\mathbf{x}_i)$ is maximized. In other words, NCL considers training errors from all other networks while training a network. The parameter λ controls the trade-off between the training error term and the penalty term. In the extreme case of $\lambda = 0$, all

individual networks are trained independently. With the increase of λ , more and more attention is paid to minimizing the correlation-based penalty.

3. Proposed NCLELM and EnSSNCLELM Soft Sensor Methods

To handle the issue of label scarcity for quality prediction in the industrial process, this work aims to leverage both labeled and unlabeled data, to improve the inferential performance of soft sensors, by combining semi-supervised and ensemble learning. For this purpose, we propose NCLELM and EnSSNCLELM soft sensor methods, which are described in detail in the following sections.

3.1. NCLELM

In recent years, ELM has gained growing popularity in soft sensor applications, due to its fast learning speed and good generalization performance [63–66]. However, ELM often produces unstable predictions, due to the uncertainties caused by the random assignments of input weights and biases in the learning process. A popular approach to this problem is to utilize an ensemble strategy to improve the stability and accuracy of the ELM model.

Since the NCL learning approach has been proven effective for enhancing the ensemble performance by explicitly maximizing the diversity among base models [67–69], we derived the NCLELM algorithm by introducing NCL into ELM learning. NCLELM aims to encourage the diversity among component ELM models, by reducing the correlation among their outputs, and, thus, gaining an overall good ensemble accuracy. Thanks to the fast training speed of ELM, NCLELM can also be trained very efficiently, using the multiple ELM models that are included in the learning. Subsequently, we present the proposed NCLELM method in detail.

Given a training set $D = \{\mathbf{x}_i, y_i\}_{i=1}^L$ and an ensemble of ELM models containing N_{ELM} individuals, the error function of the n th individual is expressed as follows:

$$e_n(\boldsymbol{\beta}_n) = \frac{1}{2} \left\{ \|\mathbf{H}_n \boldsymbol{\beta}_n - \mathbf{y}\|^2 - \lambda \|\mathbf{H}_n \boldsymbol{\beta}_n - f_{\text{ens}}\|^2 \right\} \quad (9)$$

where \mathbf{H}_n and $\boldsymbol{\beta}_n$ denote the hidden-layer output matrix and output-layer weight vector for the n th individual ELM model, respectively, and $\mathbf{y} = [y_1, y_2, \dots, y_L]^T$ is a column vector. λ is a trade-off parameter on the correlation penalty term. f_{ens} denotes the simple average of outputs from N_{ELM} individuals:

$$f_{\text{ens}} = \frac{1}{N_{\text{ELM}}} \sum_{n=1}^{N_{\text{ELM}}} \mathbf{H}_n \boldsymbol{\beta}_n \quad (10)$$

Then, the solution to the quadratic optimization problem in Equation (9) can be obtained by setting $(\partial e_n / \partial \boldsymbol{\beta}_n) = 0$, that is

$$\frac{\partial e_n}{\partial \boldsymbol{\beta}_n} = (\mathbf{H}_n \boldsymbol{\beta}_n - \mathbf{y})^T \mathbf{H}_n - \lambda (\mathbf{H}_n \boldsymbol{\beta}_n - f_{\text{ens}}) \left(1 - \frac{1}{N_{\text{ELM}}} \right) \mathbf{H}_n = 0 \quad (11)$$

By substituting Equation (10) into Equation (11), we can get

$$\frac{\partial e_n}{\partial \boldsymbol{\beta}_n} = (\mathbf{H}_n \boldsymbol{\beta}_n - \mathbf{y})^T \mathbf{H}_n - \lambda \left(\mathbf{H}_n \boldsymbol{\beta}_n - \frac{1}{N_{\text{ELM}}} \sum_{n=1}^{N_{\text{ELM}}} \mathbf{H}_n \boldsymbol{\beta}_n \right) \left(1 - \frac{1}{N_{\text{ELM}}} \right) \mathbf{H}_n = 0 \quad (12)$$

Furthermore, Equation (12) is transformed as

$$\left[1 - \left(\frac{N_{\text{ELM}} - 1}{N_{\text{ELM}}} \right)^2 \right] \lambda \mathbf{H}_n^T \mathbf{H}_n \boldsymbol{\beta}_n + \frac{N_{\text{ELM}} - 1}{N_{\text{ELM}}^2} \lambda \sum_{m \neq n}^{N_{\text{ELM}}} \mathbf{H}_n^T \mathbf{H}_m \boldsymbol{\beta}_m = \mathbf{H}_n^T \mathbf{y} \quad (13)$$

By applying Equation (13) to all individual errors e_n and all output weights β_n , and let

$$\mathbf{G}_{n,n} = [1 - \left(\frac{N_{\text{ELM}} - 1}{N_{\text{ELM}}}\right)^2 \lambda] \mathbf{H}_n^T \mathbf{H}_n \quad (14)$$

and

$$\mathbf{G}_{n,m} = \frac{N_{\text{ELM}} - 1}{N_{\text{ELM}}^2} \lambda \sum_{m \neq n}^{N_{\text{ELM}}} \mathbf{H}_n^T \mathbf{H}_m \quad (15)$$

Then the overall optimization of all individual ELM models is achieved by solving a linear system, as follows:

$$\mathbf{G} \beta_{\text{ens}} = \mathbf{T} \quad (16)$$

$$\text{where } \mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \mathbf{G}_{1,N_{\text{ELM}}} \\ \vdots & \ddots & \vdots \\ \mathbf{G}_{N_{\text{ELM}},1} & \cdots & \mathbf{G}_{N_{\text{ELM}},N_{\text{ELM}}} \end{bmatrix}, \beta_{\text{ens}} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{N_{\text{ELM}}} \end{bmatrix}, \text{ and } \mathbf{T} = \begin{bmatrix} \mathbf{H}_1^T \mathbf{y} \\ \vdots \\ \mathbf{H}_{N_{\text{ELM}}}^T \mathbf{y} \end{bmatrix}.$$

Finally, the solution to β_{ens} can be analytically obtained as

$$\hat{\beta}_{\text{ens}} = \mathbf{G}^{-1} \mathbf{T} \quad (17)$$

3.2. EnSSNCLELM

The proposed EnSSNCLELM method can be split into three steps: (i) obtaining high-confidence pseudo-labeled data, (ii) building diverse semi-supervised models, and (iii) combining diverse semi-supervised models.

3.2.1. Formulating the Pseudo-Labeling Optimization Problem

Traditionally, the pseudo labels for the unlabeled data are estimated through self-labeling techniques, such as self-training and co-training. However, such strategies are essentially an implicit optimization, and they often encounter error accumulation and propagation. In addition, it is difficult to evaluate the confidence of pseudo labels effectively. Thus, in our recent study [70], a single-learner pseudo-labeling optimization (SLPLO) scheme was proposed, to achieve reliable estimations of pseudo labels in an explicit way. Nevertheless, one particular drawback of this approach is that only utilizing one single learner to evaluate the goodness of pseudo labels may lead to a high risk. By contrast, multiple learners can provide diverse views and insights, which are preferable to obtain a reliable confidence evaluation.

Therefore, in this work, we propose a multi-learner pseudo-labeling optimization (MLPLO) approach, aiming to produce more reliable pseudo labels than the SLPLO scheme. The task of MLPLO is to explicitly approximate the unknown labels for given unlabeled data, by relying on a group of diverse learners to explore the available information behind the labeled and unlabeled data.

As preliminary steps, the decision variables, optimization objectives, and constraints should be determined in advance; thus, formulating the pseudo-labeling optimization problem. Given a labeled training set $D^l = \{\mathbf{X}^l, \mathbf{y}^l\}$, where $\mathbf{X}^l = \{\mathbf{x}_i^l\}_{i=1}^L$ and $\mathbf{y}^l = \{\mathbf{y}_i^l\}_{i=1}^L$ are the input and output data, respectively, and L represents the number of labeled samples. Meanwhile, denote $\mathbf{X}^u = \{\mathbf{x}_i^u\}_{i=1}^U$ as the input data for the unlabeled dataset D^u , where U is the number of unlabeled samples. In addition, let $\mathbf{y}^u = \{\mathbf{y}_i^u\}_{i=1}^U$ be the pseudo labels of \mathbf{X}^u and $D^{pl} = \{\mathbf{X}^u, \mathbf{y}^u\}$ be the pseudo-labeled set, where \mathbf{y}^u are actually unknown in practical applications and remain to be estimated. Hence, \mathbf{y}^u is determined as the decision vector for the proposed MLPLO problem (MLPLOP).

Subsequently, the objective function should be defined to evaluate the quality of the candidate solutions. For the MLPLO approach, it is necessary to build multiple NCLELM models with high diversity, when considering NCLELM as the base learning technique. This can be achieved by repeating the random generation of input weights and biases,

thereby producing diverse NCLELM models $H = \{\text{NCLELM}_1, \dots, \text{NCLELM}_M\}$. It is worth noting that, during the MLPLO process, the initial configurations of input weights and biases will remain unchanged, though the output weights will be updated with the change of training set.

Within the multi-learner PLO scenario, the key to defining appropriate optimization objective lies in analyzing the individual and collaborative prediction capability of base models on the labeled and pseudo-labeled data. Specifically, four evaluation criteria are defined; including the individual accuracy using the pseudo-labeled data, individual accuracy improvement after including the pseudo-labeled data, smoothness of labeled and pseudo-labeled data, and ensemble accuracy of multiple learners using the pseudo-labeled data.

Suppose there is a pseudo-labeled set $D^{pl} = \{\mathbf{X}^u, \mathbf{y}^u\}$ that needs to be optimized, we randomly divide D^{pl} into M equal-sized training subsets $D^{pl} = \{D_1^{pl}, \dots, D_M^{pl}\}$ and then feed these to the built NCLELM models, respectively, where $D_i^{pl} = \{\mathbf{X}_i^u, \mathbf{y}_i^u\} = \{(x_{i,1}^u, y_{i,1}^u), \dots, (x_{i,P}^u, y_{i,P}^u)\}$ with P denoting the number of samples in the i th subset. The goal of such data partition is to promote the diversity of NCLELM models. Next, the defined evaluation criteria are discussed as follows.

- (1) Individual accuracy using the pseudo-labeled data. It is well known that successful data-driven modeling greatly relies on the assumption that modeling data are independent and identically distributed. That is to say, good prediction performance on unseen samples can be attained only when the test and training data come from the same distribution. This assumption usually applies to developing data-based models for industrial processes. Thus, in the context of semi-supervised modeling, we assume that the labeled and unlabeled data are drawn from the same distribution. Intuitively, this implies that a NCLELM model trained with the pseudo-labeled data can also provide accurate predictions on the labeled set, if the pseudo labels are estimated well enough. Specifically, suppose $H^{pl} = \{h_1^{pl}, h_2^{pl}, \dots, h_M^{pl}\}$ denote the diverse NCLELM models learned from the pseudo-labeled subsets $D_1^{pl}, D_2^{pl}, \dots, D_M^{pl}$, respectively. It is obvious that we expect the performance of h_i^{pl} on D^l to be good if the acquired D_i^{pl} has high quality. Furthermore, by considering all individual accuracies simultaneously, the overall accuracy is minimized:

$$\min_{\mathbf{y}^u} F(\mathbf{y}^u, \mathbf{X}^u, \mathbf{y}^l, \mathbf{X}^l, H) = \sum_{i=1}^M \frac{1}{L} \sum_{j=1}^L (\hat{y}_{i,j}^{pl} - y_j^l)^2 \quad (18)$$

where $\hat{y}_{i,j}^{pl}$ represents the predicted label of the j th labeled sample using h_i^{pl} , and y_j^l is the j th actual label.

- (2) Individual accuracy improvement after including the pseudo-labeled data. In many self-labeling semi-supervised learning algorithms [48–50], the pseudo labels are usually estimated from the already built predictive models. Then, the confidence of these pseudo-labeled data is evaluated according to the prediction accuracy enhancement of the model after adding the target pseudo-labeled data to the original training set. The larger the performance improvement, the higher the confidence of the pseudo-labeled data. Similarly, we also employ this way to evaluate the confidence of the optimized pseudo labels. Suppose $H^{l+pl} = \{h_1^{l+pl}, h_2^{l+pl}, \dots, h_M^{l+pl}\}$ are the NCLELM models learned from $D^l \cup D_1^{pl}, D^l \cup D_2^{pl}, \dots, D^l \cup D_M^{pl}$, respectively. Then, it is desirable to minimize the prediction errors of H^{l+pl} on the labeled training set:

$$\min_{\mathbf{y}^u} F(\mathbf{y}^u, \mathbf{X}^u, \mathbf{y}^l, \mathbf{X}^l, H) = \sum_{i=1}^M \frac{1}{L} \sum_{j=1}^L (\hat{y}_{i,j}^{l+pl} - y_j^l)^2 \quad (19)$$

where $\mathbf{y}^u = [y_{1,1}^u, y_{1,2}^u, \dots, y_{1,P}^u, \dots, y_{M,1}^u, y_{M,2}^u, \dots, y_{M,P}^u]$ denotes the decision vector, $\gamma_i \geq 0$ ($i = 1, 2, 3$) are trade-off parameters. $[y_{\min}, y_{\max}]$ are the lower and upper bounds with each pair of elements for one decision variable being denoted as $[y_{\min,i,j}, y_{\max,i,j}]$, where $i = 1, 2, \dots, M$ and $j = 1, 2, \dots, P$. Note that the upper and lower settings can severely affect the efficiency of the MLPLO approach, so they should be determined carefully. To this end, the prediction uncertainty of the pseudo labels is first estimated through a probabilistic modeling technique and then utilized to aid in determining the boundary of the decision variables. Specifically, in this work, a GPR modeling technique [72] is applied to provide the confidence intervals of the pseudo labels, which includes three steps, i.e., first, building a GPR model from the labeled data D^l , then, obtaining the prediction variances of the unlabeled data using the GPR model, and, finally, determining the search ranges based on the confidence intervals.

3.2.2. Solving the Pseudo-Labeling Optimization Problem

For the formulated MLPLO problem shown in Equation (24), it can be seen that NCLELM training and prediction are performed repeatedly during the calculation of the objective function, which means the objective function fails to satisfy continuity, differentiability, convexity, etc. Obviously, the classical optimization techniques are not suitable for solving such an optimization problem. Fortunately, evolutionary algorithms, inspired by biological phenomena, have earned great success in both machine learning and engineering applications for many tasks, such as feature selection, model selection, and optimal scheduling [73–77]. Unlike the classical optimization methods, evolutionary approaches conduct a parallel search, and they have the advantages of adapting to complex problems, where derivability and convexity are not required. Despite the availability of numerous evolutionary algorithms, for the sake of simplicity, one of the most well-known and commonly used approaches, genetic algorithms (GA), is chosen as an example to illustrate how to solve the MLPLO problem through an evolutionary approach.

The strategy of producing high-confidence pseudo labels using GA optimization is illustrated in Figure 1. The details of individual representation, fitness evaluation, and evolutionary operations are presented, as follows:

(1) Individual representation. In most cases, the quality variables in the industrial process are real numbers, thus the decision variables $\mathbf{y}^u = \{y_{1,1}^u, y_{1,2}^u, \dots, y_{1,P}^u, \dots, y_{M,1}^u, y_{M,2}^u, \dots, y_{M,P}^u\}$ are encoded as chromosomes by real-number coding, as illustrated in Fig. 2. In addition, an initial population with N_{pop} individuals is generated randomly within the range of $[y_{\min}, y_{\max}]$.

(2) Fitness evaluation. The goal of this step is to determine the quality of candidate solutions in the populations. In this case, we compute the value of the objective function defined in Equation (24) and use its reciprocal as the fitness value to evaluate the goodness of the chromosomes.

(3) Evolutionary operations. If the stopping condition is not satisfied, the GA needs to generate an offspring population by executing evolutionary operations; i.e., selection, crossover, and mutation. The selection operator can find the good solutions with the largest fitness values in a population. Crossover creates new individuals, by combining the genes of one individual with those of another. In mutation, the genes in individuals will make small random changes to create mutation children. Such an operation provides genetic diversity and allows enlarging the search space.

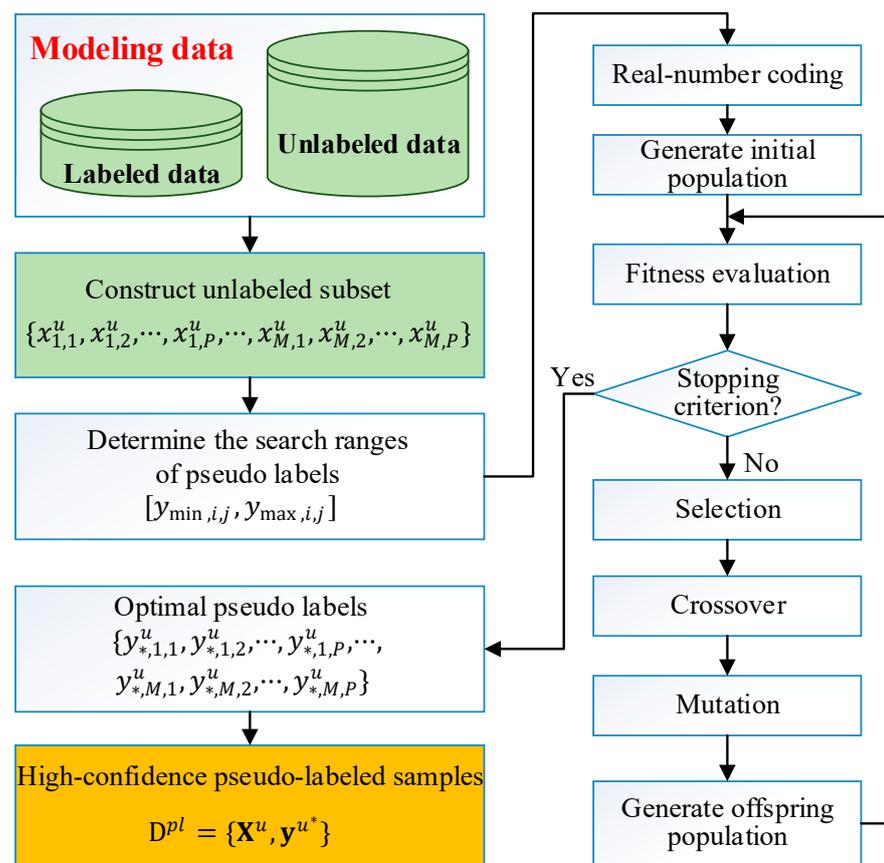


Figure 1. Schematic diagram of solving the MLPLO problem using GA.

After performing the GA based MLPLO optimization, the optimal solution encoding the best pseudo labels is obtained, i.e., $y_*^u = [y_{*,1,1}^u, y_{*,1,2}^u, \dots, y_{*,1,P}^u, \dots, y_{*,M,1}^u, y_{*,M,2}^u, \dots, y_{*,M,P}^u]$. Then, by combining y_*^u with its corresponding X^u , we get the optimized pseudo-labeled set $D^{pl} = \{X^u, y^{u*}\}$.

3.2.3. Building Diverse SSNCLEM Base Models

In order to obtain a good ensemble model, we need to construct accurate and diverse base models. Once the MLPLO process is completed, a group of pseudo-labeled samples can be obtained and further combined with the labeled set to build a set of SSNCLELM models. It is fairly easy to see that the performance of the SSNCLELM model is highly dependent on the quality of the optimized pseudo-labeled samples. Hence, to ensure the reliability of the MLPLO process, the following two problems should be handled properly:

(1) Selection of the unlabeled samples. Though there are a large number of unlabeled data available for SSL learning in practical industrial processes, the number of unlabeled samples for the MLPLO approach cannot be too large. As indicated in Figure 2, the dimension of the decision variables is equal to the size of the selected unlabeled data; hence, inclusion of too many unlabeled samples will inevitably make the MLPLO problem become a large-scale optimization problem [78]. In this case, it is very difficult to obtain good a search performance, efficiency, and effectiveness. In addition, introducing too many pseudo-labeled samples into the SSNCLELM model construction will also weaken the influence of the labeled samples, thus leading to high modeling risk. Therefore, it is more realistic to generate diverse small-scale pseudo-labeled sets and then construct diverse SSNCLELM base models.

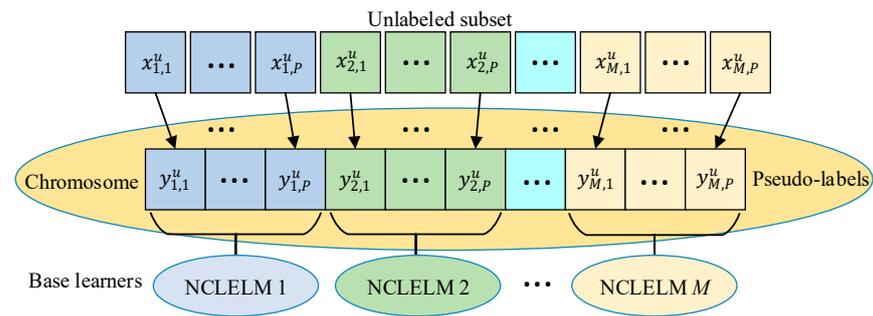


Figure 2. Individual structure for GA-based pseudo-labeling optimization.

(2) Setting of the trade-off parameters $\{\gamma_1, \gamma_2, \gamma_3\}$. Besides the unlabeled data selection, another important factor affecting the optimization performance of the MLPLO approach, is determining an appropriate combination of trade-off parameters. In practice, it is a difficult task to find a globally optimal combination of the three parameters, and there is usually more than one combination that can meet the requirements. Considering this, a natural method is to apply diverse settings of trade-off parameters for the MLPLO process, thus avoiding the difficulties in parameter selection, as well generating diverse pseudo-labeled samples. This is very helpful for enhancing the diversity of SSNCLELM base models.

In light of the above problems, we aim to generate diverse SSNCLELM models through a multi-model perturbation mechanism, which combines random assignment of NCLELM model parameters, random selection of small-scale unlabeled samples for MLPLO optimization, and diverse settings of trade-off parameters $\{\gamma_1, \gamma_2, \gamma_3\}$.

Let $\Phi_k = \left\{ \{\text{NCLELM}_{\text{init},i}\}_{i=1}^M, D^u, \{\gamma_i\}_{i=1}^3 \right\}_k$ denote the specialized setting for the k th run of diverse SSNCLELM model construction, where $\{\text{NCLELM}_{\text{init},i}\}_{i=1}^M$ are generated by randomly assigning input weights and biases, the selected unlabeled data D^u are randomly resampled from D^u , and $\{\gamma_i\}_{i=1}^3$ is a combination from $\gamma_1 \in \{\gamma_{1,i}\}_{i=1}^{k_1}$, $\gamma_2 \in \{\gamma_{2,i}\}_{i=1}^{k_2}$, and $\gamma_3 \in \{\gamma_{3,i}\}_{i=1}^{k_3}$. By applying Φ_k to the proposed MLPLO approach, an enlarged labeled set $D^{l+pl} = D^l \cup D_k^{pl}$ can be obtained and further used to update the initial models $\{\text{NCLELM}_{\text{init},i}\}_{i=1}^M$, thus resulting in a set of diverse SSNCLELM models $\{\text{SSNCLELM}_{k,1}, \text{SSNCLELM}_{k,2}, \dots, \text{SSNCLELM}_{k,M}\}$. By repeating the above procedure $K = k_1 \times k_2 \times k_3$ times, we can obtain a total of $K \times M$ diverse SSNCLELM models:

$$\begin{cases} \Phi_1 : \{\text{SSNCLELM}_{1,1}, \text{SSNCLELM}_{1,2}, \dots, \text{SSNCLELM}_{1,M}\} \\ \vdots \\ \Phi_K : \{\text{SSNCLELM}_{K,1}, \text{SSNCLELM}_{K,2}, \dots, \text{SSNCLELM}_{K,M}\} \end{cases} \quad (25)$$

3.2.4. Combining Diverse SSNCLELM Base Models

After obtaining diverse SSNCLELM models, we need to combine these models in an appropriate manner. It is a common practice to utilize all base models for the ensemble construction. However, according to Zhou's finding of 'many could be better than all' [71], it may be better to combine only part rather than all of the base models. The underlying reason is that there may exist poor base models, which can hurt the ensemble performance instead of improving it. Thus, ensemble pruning has been strongly emphasized for ensemble model construction [60,79,80]. Regarding the area of semi-supervised learning, the introduction of unlabeled and pseudo-labeled information may lead to unsafe semi-supervised learning [81]. That is, the prediction capability of a model cannot be improved, but encounters deterioration after including the unlabeled and pseudo-labeled data.

Thus, we propose an ensemble pruning strategy based on performance improvement evaluation. First, the supervised prediction performance of a NCLELM model learnt from the labeled data is evaluated on an independent validation set $D_{\text{val}} = \{\mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}}\}$. Then, a

SSNCLELM model is trained from the augmented labeled set containing the pseudo-labeled data, and its performance is further evaluated using the same validation set. Finally, the performance improvement ratio is calculated, to decide whether the candidate SSNCLELM model is retained or not. Specifically, for the i th SSNCLELM, the performance improvement ratio (PIR) is calculated as follows:

$$RMSE_i^{init} = \text{NCLELM}_i.\text{evaluate}(D_{val}) \quad (26)$$

$$RMSE_i = \text{SSNCLELM}_i.\text{evaluate}(D_{val}) \quad (27)$$

$$PIR_i = (RMSE_i^{init} - RMSE_i) / RMSE_i^{init} \quad (28)$$

where $RMSE_i^{init}$ represents the RMSE of the i th NCLELM on D_{val} and $RMSE_i$ is the RMSE of the i th SSNCLELM on D_{val} . Then, a threshold PIR_{th} is required to select SSNCLELM models whose PIR values exceed PIR_{th} . In general, PIR_{th} can be set to 0, as used in our case studies, which means that those SSNCLELM models without performance improvement on the validation set are discarded.

After the ensemble pruning, suppose that a total of $S \leq K \times M$ SSNCLELM models are retained for ensemble construction. Since the simple average rule cannot always function well, it is appealing to integrate the base models using a weighting scheme. Hence, in this work, a stacking strategy is employed for model combination. The basic idea of this approach is to use the built SSNCLELM models as the first-level learners, and then train the second-level learner by using the outputs of the first-level learners as inputs, while the original output is still regarded as the output of the model. That is,

$$\text{EnSSNCLELM} = \text{Stacking}(\text{SSNCLELM}_{sel,1}, \dots, \text{SSNCLELM}_{sel,S}) \quad (29)$$

Specifically, given an independent validation set $D_{val} = \{\mathbf{X}_{val}, \mathbf{y}_{val}\}$, the prediction outputs $\hat{\mathbf{y}}_{val} = \{\hat{\mathbf{y}}_{val}^1, \hat{\mathbf{y}}_{val}^2, \dots, \hat{\mathbf{y}}_{val}^S\}$ on D_{val} can be obtained using the selected base models $\{\text{SSNCLELM}_{sel,i}\}_{i=1}^S$. Then, a stacking model can be built from $\{\hat{\mathbf{y}}_{val}, \mathbf{y}_{val}\}$, where $\hat{\mathbf{y}}_{val}$ and \mathbf{y}_{val} are used as input and output data, respectively. To effectively handle the collinearity issue among the SSNCLELM base models, a PLS stacking model is built. Thus, for one query sample \mathbf{x}_q , the ensemble prediction output \hat{y}_q can be obtained as

$$\hat{y}_q = \beta_0 + \sum_{i=1}^S \beta_i \hat{y}_q^i \quad (30)$$

where $\{\beta_0, \beta_1, \dots, \beta_S\}$ denotes the PLS regression coefficients, and \hat{y}_q^i is the predicted value from the i th SSNCLELM model.

3.2.5. Implementation Procedure of the EnSSNCLELM Soft Sensor

The pseudo-code of the proposed EnSSNCLELM soft sensor method is described in Algorithm 1, and its workflow is illustrated in Figure 3.

Algorithm 1 EnSSNCLELM soft sensor method.

INPUT: D^l : labeled set

D^u : unlabeled set

D_{val} : validation set

n_{pop} : population size for GA optimization

n_{gen} : maximum number of iterations

$\{\gamma_1, \gamma_2, \gamma_3\}$: trade-off parameters

$\{N_{ELM}, N_{node}, \lambda\}$: the number of ELM in each NCLELM, the number of hidden node size of ELM base models, the tradeoff parameter of NCLELM method, respectively

U : the number of unlabeled data D^u that requires pseudo labeling

M : the number of NCLELM models for the MLPLO approach

Algorithm 1 *Cont.*

PROCESS:

- 1: Generate diverse combinations of parameters $\{\gamma_{1,k}, \gamma_{2,k}, \gamma_{3,k}\}_{k=1}^K$ for the MLPLO optimization, where $K = k_1 \times k_2 \times k_3$, $\gamma_1 \in \{\gamma_{1,i}\}_{i=1}^{k_1}$, $\gamma_2 \in \{\gamma_{2,i}\}_{i=1}^{k_2}$ and $\gamma_3 \in \{\gamma_{3,i}\}_{i=1}^{k_3}$;
%% Building diverse SSNCLELM models
- 2: for $k = 1$ to K do
 %% Estimating the pseudo labels through the MLPLO approach
- 3: Generate M new initial NCLELM models $\{\text{NCLELM}_{\text{init},i}\}_{i=1}^M$ with the hyperparameters $\{N_{\text{ELM}}, N_{\text{node}}, \lambda\}$ by randomly setting the input weights and biases;
- 4: Select a small-scale unlabeled set $D^{u'}$ by randomly resampling from D^u ;
- 5: Determine the pseudo labels of $D^{u'}$ as the decision variables
 $\mathbf{y}_k^u = \{y_{1,1}^u, y_{1,2}^u, \dots, y_{1,P}^u, \dots, y_{M,1}^u, y_{M,2}^u, \dots, y_{M,P}^u\}$ with $P = U' / M$ and set the lower and upper bounds $[\mathbf{y}_{\min}, \mathbf{y}_{\max}]$ for \mathbf{y}_k^u based on GPR regression analysis;
- 6: Encode the decision variables as real-valued chromosome s , and randomly generate an initial population $Pop = \{s_i\}_{i=1}^{n_{pop}}$ with n_{pop} individuals within the decision boundary;
- 7: Repeat n_{gen} times:
8: Decode the pseudo labels of \mathbf{y}_k^u from each individual chromosome in the population;
- 9: Evaluate the fitness of each individual s_i in Pop according to Equation (24);
- 10: Generate an offspring population by performing selection, crossover, and mutation operations;
 mutation operations;
- 11: end of Repeat
- 12: Select the best individual s^* from the final population;
- 13: Obtain the best pseudo labels \mathbf{y}^{u*} by decoding the chromosome s^* and form the pseudo-labeled set D_k^{pl} ;
 %% Building diverse SSNCLELM models for each run
- 14: Update the initial models $\{\text{NCLELM}_{\text{init},i}\}_{i=1}^M$ using the optimized enlarged labeled set $D^l \cup D_k^{pl}$ to obtain the semi-supervised models $\{\text{SSNCLELM}_{k,i}\}_{i=1}^M$;
- 15: end of for
- %% Ensemble pruning based on performance improvement evaluation
- 16: Let $\Omega = \{\text{SSNCLELM}_{1,1}, \dots, \text{SSNCLELM}_{1,M}, \dots, \text{SSNCLELM}_{K,1}, \dots, \text{SSNCLELM}_{K,M}\}$;
- 17: for $i = 1$ to K do
18: for $j = 1$ to M do
19: $\text{RMSE}_{i,j} = \text{SSNCLELM}_{i,j}.\text{evaluate}(D_{\text{val}})$;
- 20: $\text{PIR}_{i,j} = (\text{RMSE}_{i,j}^{\text{init}} - \text{RMSE}_{i,j}) / \text{RMSE}_{i,j}^{\text{init}}$;
- 21: if $\text{PIR}_{i,j} \geq \text{PIR}_{\text{th}}$;
- 22: Add $\text{SSNCLELM}_{i,j}$ to the selected model pool Ω_{sel} ;
- 23: end of if
24: end of for
25: end of for
- 26: Let $\Omega_{\text{sel}} = \{\text{SSNCLELM}_{\text{sel},1}, \text{SSNCLELM}_{\text{sel},2}, \dots, \text{SSNCLELM}_{\text{sel},S}\}$;
 %% Constructing PLS stacking ensemble model
- 27: Let $D_{\text{val}} = \{X_{\text{val}}, y_{\text{val}}\}$;
- 28: $\{\hat{y}_{\text{val}}^1, \hat{y}_{\text{val}}^2, \dots, \hat{y}_{\text{val}}^S\} = \Omega.\text{predict}(X_{\text{val}})$;
- 29: Let $\hat{\mathbf{y}}_{\text{val}} = \{\hat{y}_{\text{val}}^1, \hat{y}_{\text{val}}^2, \dots, \hat{y}_{\text{val}}^S\}$;
- 30: $\{\beta_m\}_{m=0}^S = \text{PLS.fit}(\hat{\mathbf{y}}_{\text{val}}, y_{\text{val}})$, where β is the regression coefficient of PLS;
- %% Online prediction
- 31: Given a query sample x_q ;
- 32: $\{\hat{y}_{q,m}\}_{m=1}^S = \Omega.\text{predict}(x_q)$;
- 33: The final prediction output \hat{y}_q of x_q are obtained by Equation (30).

OUTPUT: \hat{y}_q

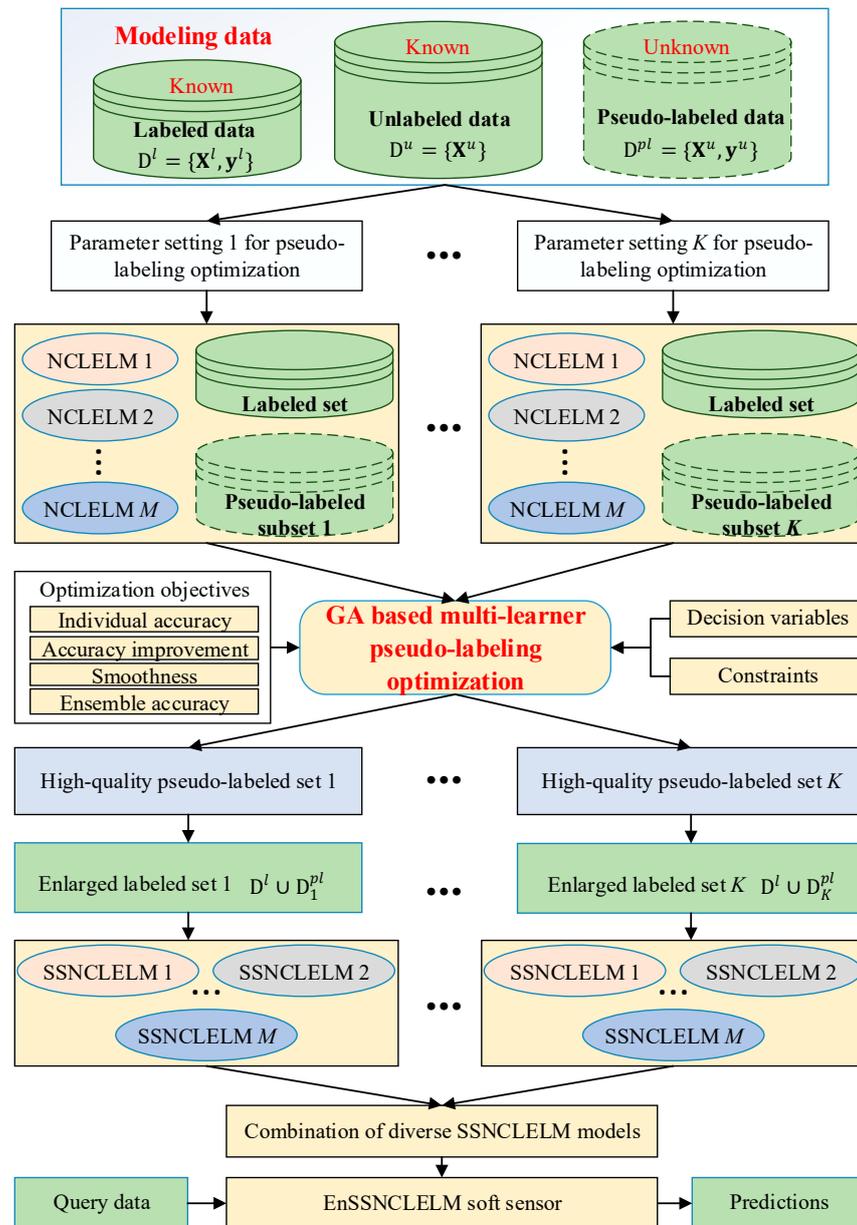


Figure 3. Workflow of the proposed EnSSNCLELM soft sensor.

4. Case Studies

In this section, the performance of the proposed NACLELM and EnSSNCLELM soft sensor methods are evaluated through a simulated penicillin fermentation process and an industrial fed-batch chlortetracycline (CTC) fermentation process. The methods for comparison are as follows:

- (1) ELM [61]: the supervised single ELM model.
- (2) NACLELM: the proposed supervised ensemble of ELM models using the NCL rule.
- (3) EnNACLELM_{avg}: the supervised ensemble of NACLELM models using the simple averaging rule.
- (4) EnNACLELM_{pls}: the supervised ensemble of NACLELM models using the PLS stacking strategy.
- (5) CoELM: the co-training based semi-supervised ELM model, where the confidence evaluation method proposed by Zhou and Li (2005) [43] is used and the Euclidean distance is used to choose similar samples for confidence evaluation, and two different ELM models are trained using two randomly generated feature subsets.

(6) SSELM [29]: semi-supervised ELM model embedding the graph based Laplacian regularization.

(7) EnSSNCLELM_{avg}: the semi-supervised ensemble of diverse SSNCLELM models using the simple averaging rule.

(8) EnSSNCLELM: the proposed method, the semi-supervised ensemble of SSNCLELM models using PLS stacking strategy.

To quantitatively assess the prediction performance of different soft sensors, the root-mean-square error (RMSE) and coefficient of determination (R^2) are used:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N_{\text{test}}} (y_n - \hat{y}_n)^2} \quad (31)$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N_{\text{test}}} (\hat{y}_n - y_n)^2}{\sum_{n=1}^{N_{\text{test}}} (\hat{y}_n - \bar{y})^2} \quad (32)$$

where N_{test} is the number of testing samples; \hat{y}_n and y_n are the n th predicted and actual values of the testing sample, respectively; and \bar{y} is the mean value of the output variable.

4.1. Application to Penicillin Fermentation Process

4.1.1. Process Description

The penicillin fermentation process has been widely used for investigating the modeling, monitoring, and controlling of batch processes [82,83]. A flow diagram of the process is illustrated in Figure 4. Since formation of secondary metabolites (in this case, penicillin) is not associated with cell growth, the cells generally grow in a batch culture and then achieve synthesis of antibiotics through a fed-batch operation. In general, the penicillin production process lasts for 400 h, during which two cascade controllers are used to maintain the pH and temperature. In addition, the sterile substrate and air are continuously fed into the bioreactor to supply the nutrients for cell growth and product formation, as well as satisfying the oxygen consumption necessary for the microorganisms.

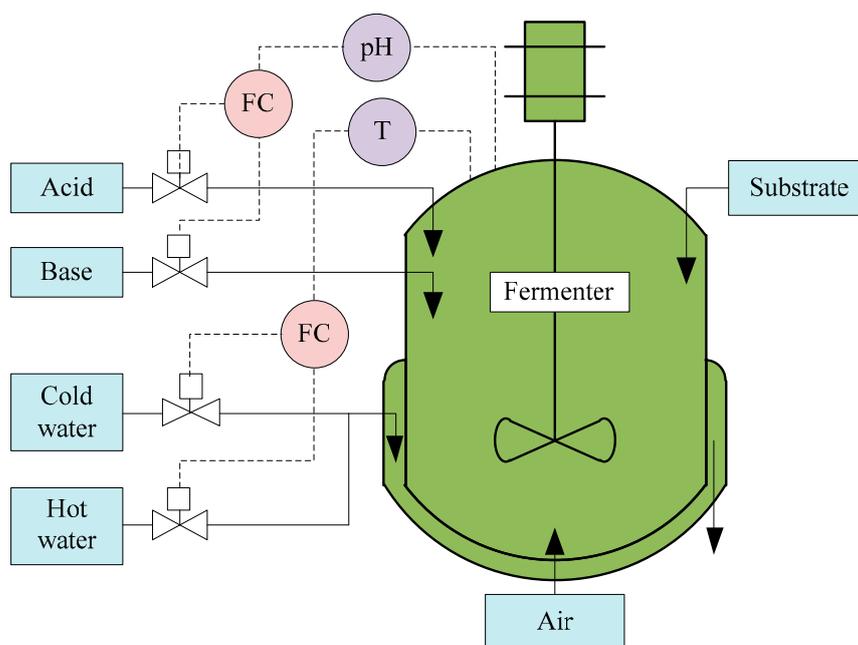


Figure 4. Flow diagram of penicillin fermentation process.

In our experimental study, the process data were collected from the PenSim platform, which was developed by the Process Modeling, Monitoring, and Control Research

Group of Illinois Institute of Technology for simulating the fed-batch penicillin fermentation process [84]. The software PenSim can be downloaded from the website on <http://simulator.iit.edu/web/pensim/index.html> (accessed on 5 August 2019). In this work, penicillin concentration was chosen as the difficult-to-measure variable, while the relevant process variables in Table 1 were used as the input variables of soft sensor models. Moreover, the sampling interval was set as 0.5 h and the default operation settings were considered.

Table 1. Input variables for soft sensor development in the penicillin fermentation process.

No.	Variable Description (Unit)
1	Culture time (h)
2	Aeration rate (L/h)
3	Agitator power (W)
4	Substrate feed rate (L/h)
5	Substrate feed temperature (K)
6	Dissolved oxygen concentration (g/L)
7	Culture volume (L)
8	Carbon dioxide concentration (g/L)
9	pH (–)
10	Fermenter temperature (K)
11	Generated heat (kcal)
12	Cooling water flow rate (L/h)

4.1.2. Prediction Performance and Discussion of NCLELM

To verify the effectiveness and superiority of the proposed NCLELM, we conducted the experiments as follows:

(1) Two application scenarios with different labeled data sizes were considered. The first case used 200 labeled samples collected from five batches for model training, while the second case used 800 labeled samples from the same batches. In addition, 800 labeled samples from another five batches were used for testing.

(2) The prediction performance of NCLELM with respect to different parameter configurations $\{N_{\text{ELM}}, N_{\text{node}}, \lambda\}$ was analyzed, where $N_{\text{ELM}} \in \{5, 10, 15, 20\}$, $N_{\text{node}} \in \{5, 6, \dots, 30\}$, and $\lambda \in \{0, 0.2, \dots, 1\}$. Among these parameters, N_{ELM} and N_{node} affect the model complexity significantly, while λ indicates the degree of concern placed on minimizing the correlation-based penalty.

(3) A comparison between NCLELM and ELM was performed to demonstrate the advantage of the NCL rule. Since both methods exhibit a random nature, due to the random assignments of input weights and biases; for the sake of fairness, their prediction accuracies are provided as the simple average of those from 50 runs.

Figure 5 illustrates the testing performance of the proposed NCLELM approach for different settings of parameters $\{N_{\text{ELM}}, N_{\text{node}}, \lambda\}$, where the prediction performance of ELM is also given for comparison. The effects of hyper-parameters $\{N_{\text{ELM}}, N_{\text{node}}, \lambda\}$ on the NCLELM model performance are analyzed first. It is readily seen that, overall, the prediction errors of NCLELM decreased with the increase of N_{node} . It is also noticeable that, when large N_{node} was used, the performance enhancement resulting from the increase of N_{node} became subtle. This was mainly because a high model complexity tends to cause overfitting when the training data are insufficient. For example, in the application case with 200 labeled training samples, as shown in Figure 5a, the testing accuracy was not improved significantly after N_{node} exceeded 15, due to the scarcity of the labeled training data. In comparison, as shown in Figure 5b, the trend plots from the case with 800 labeled training samples decreased more sharply and the testing RMSE values continuously became smaller when N_{node} increased.

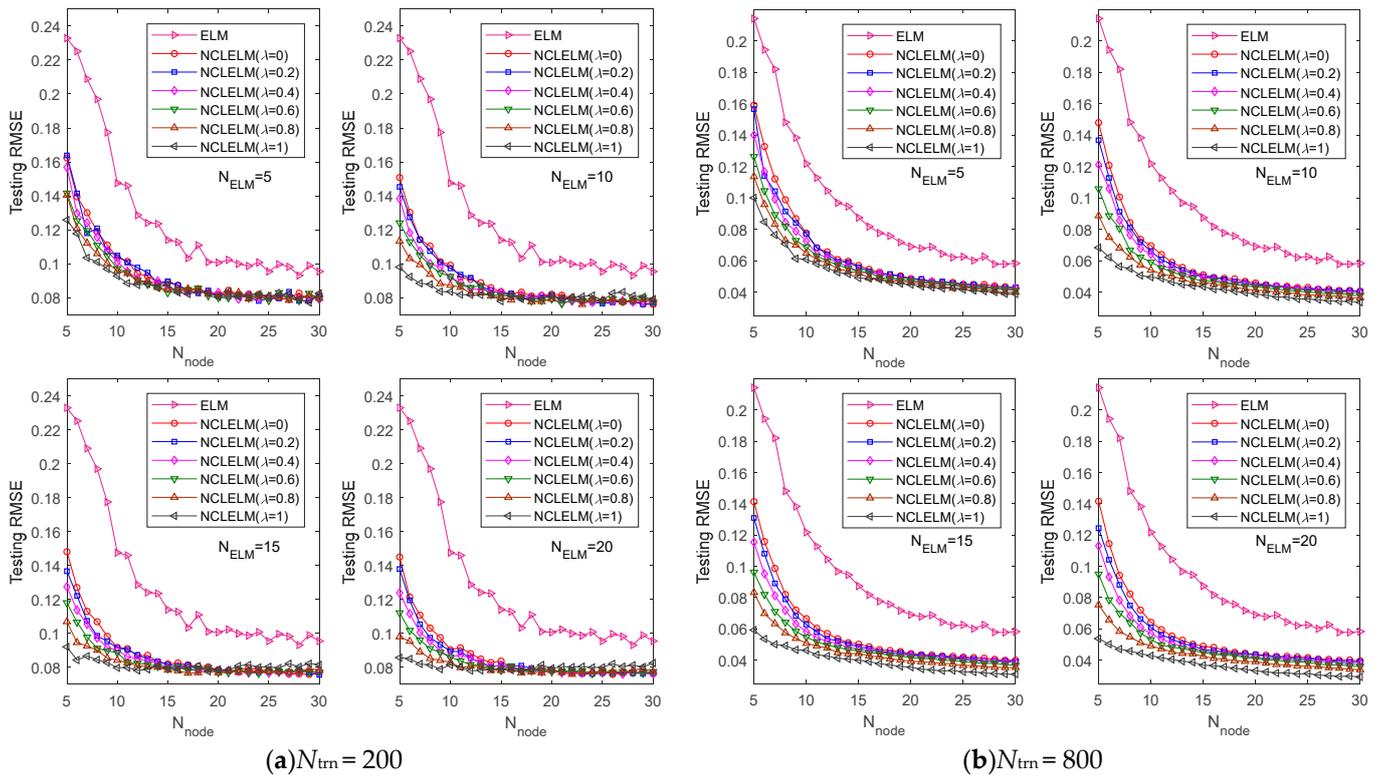


Figure 5. Prediction results of the ELM, EnELM, and NCLELM methods under different parameter settings. In (a,b), the training sample size was $N_{\text{trn}} = 200$ and 800 , respectively.

Similarly to N_{node} , N_{ELM} also had a great impact on the NCLELM model complexity. It can be seen from the four subplots in Figure 5a, when a small N_{node} was used for NCLELM, a large N_{ELM} was more likely to produce a high prediction accuracy. However, when a large N_{node} was considered, the increase of N_{ELM} only led to a slight enhancement of the testing accuracy. A similar phenomenon can also be found in Figure 5b. Based on the above analysis, we can conclude that, when the training data size is small, a less complex model structure is preferable; while a more complex model, which can be built by increasing N_{node} and/or N_{ELM} , is required when the training data are sufficient.

Unlike N_{node} and N_{ELM} , λ was introduced to control the importance of the correlation based penalty objective in the training loss function. When the individual model complexity was not so high, e.g., in the case with a small N_{node} , a large λ was required to ensure the prediction accuracy of NCLELM. One possible reason for this is that, in such a situation, it is desirable to emphasize the NCL effect of NCLELM, to guarantee the diversity among the ELM base models, which is crucial for building a high-performance ensemble model. As the model complexity grows, the benefits from introducing the NCL rule decline, even though a large λ is employed. One possible answer to this question is that, when a complex model structure is used, the diversity among the base ELM models for the NCLELM construction can be effectively maintained only through the inherent random assignments of the NCLELM parameter configurations. In this case, the NCL rule-based diversity improvement is not obvious. In the case of $\lambda = 0$, NCLELM degenerates as the simple averaging rule based ensemble of ELM models.

Subsequently, we compared the prediction accuracy of the NCLELM and ELM methods. As can be seen from the figure, similarly to NCLELM, the prediction performance of ELM was also enhanced with the increase of N_{node} . When the training data are limited (e.g., $N_{\text{trn}} = 200$), a moderate N_{node} , such as 20 is acceptable, whereas a larger N_{node} , such as 30, is more suitable for a case with a relatively large training data size, such as $N_{\text{trn}} = 800$. Meanwhile, we found that the performance of NCLELM was always much better than that

of ELM, regardless of the parameter settings and training data sizes, which implies the outstanding performance of NCLELM. The above experimental results fully confirm the effectiveness and superiority of the proposed NCLELM over traditional ELM.

4.1.3. Analysis and Comparison of EnSSNCLELM Prediction Results

To evaluate the effectiveness of the proposed EnSSNCLELM soft sensor method, five batches, including 4000 labeled samples, were collected for soft sensor training. To imitate the practical application scenario with scarce labeled data but rich unlabeled data, the obtained training set was further divided into two parts: a small-sized labeled training set with only 200 labeled samples, and a large-sized unlabeled training set with 3800 samples, whose actual labels had been removed. In addition, an independent validation set including 40 labeled samples was collected from two batches for ensemble pruning and stacking. Moreover, another five batches were obtained to assess the online prediction performance of the soft sensors.

To build well-performing soft sensor models, some critical parameters of different soft sensor methods should be determined in advance by cross-validation or trial and error, as follows:

- (1) ELM: The hidden node size N_{node} was set as 20 by 5-fold cross-validation.
- (2) NCLELM: The hyperparameters $\{N_{\text{ELM}}, N_{\text{node}}, \lambda\}$ were set as $\{5, 20, 0.6\}$ by 5-fold cross-validation, respectively.
- (3) CoELM and SSELM: N_{node} for the two methods were selected as 20, the maximum number of iterations for CoELM was set as 300, and the penalty coefficient for SSELM was set as 0.3, by trial and error.
- (4) EnSSNCLELM: The hyperparameters $\{N_{\text{ELM}}, N_{\text{node}}, \lambda\}$ for the SSNCLELM base models were the same as those of NCLELM. The M and U' were set to 3 and 300, respectively, for formulating the MLPLOP problem. In addition, the parameter configurations for the GA based MLPLO optimization of EnSSNCLELM were set as follows: the population size $n_{\text{pop}} = 50$ and the maximum number of iterations $n_{\text{gen}} = 50$ for GA optimization, and diverse combinations of trade-off parameters $\{\gamma_1, \gamma_2, \gamma_3\}$ were generated from the candidate set of $\{0, 0.01, 0.1, 0.5, 1\}$. In addition, the number of principal components of the PLS stacking model was determined by 5-fold cross-validation.

Considering that the EnNCLELM_{avg}, EnSSNCLELM_{avg}, and EnNCLELM_{pls} methods are actually the degraded versions of the proposed EnSSNCLELM approach, their parameter configurations were set to the same as those of EnSSNCLELM. Under the above parameter settings, a total of 375 SSNCLELM models were first built based on the extended labeled data. After ensemble pruning, 350 models were retained for constructing the ensemble models EnSSNCLELM_{avg} and EnSSNCLELM. Meanwhile, the NCLELM base models, corresponding to the SSNCLELM ones, were selected for EnNCLELM_{avg} and EnNCLELM_{pls} modeling.

The prediction results of different soft sensors on the testing set are tabulated in Table 2. It can be seen that ELM had the worst performance as a single model, and NCLELM delivered a better accuracy, due to the introduction of the NCL ensemble strategy. By using ELM as the base learning technique, two representative semi-supervised methods, i.e., CoELM and SSELM, were developed using co-training and embedding graph regularization, respectively. Though CoELM and SSELM include unlabeled data and, thus, achieved an accuracy improvement, their prediction performance was still very poor. Furthermore, by using NCLELM for base model building, two supervised ensemble models, namely EnNCLELM_{avg} and EnNCLELM_{pls} are developed by introducing a simple averaging rule and PLS stacking strategy, respectively. It is readily observed that, compared to NCLELM, both ensemble methods provided a better prediction accuracy, and the comparison between EnNCLELM_{pls} and EnNCLELM_{avg} further implied the superiority of the stacking scheme over to the simple averaging strategy. In addition, compared to EnNCLELM_{avg}, EnSSNCLELM_{avg} obtained a much better prediction accuracy, due to the utilization of unlabeled data information, which can be also revealed by the comparison

between EnNCLELM_{pls} and EnSSNCLELM. Overall, due to the efficient combination of ensemble and semi-supervised learning, the proposed EnSSNCLELM approach provided a much better prediction performance than the other traditional supervised and semi-supervised methods.

Table 2. Prediction results of penicillin concentration using different soft sensor methods in the penicillin fermentation process.

Method	RMSE	R ²
ELM	0.1044	0.9479
NCLELM	0.0812	0.9689
CoELM	0.0772	0.9720
SSELM	0.1019	0.9506
EnNCLELM _{avg}	0.0778	0.9714
EnSSNCLELM _{avg}	0.0599	0.9822
EnNCLELM _{pls}	0.0627	0.9821
EnSSNCLELM	0.0440	0.9908

Figure 6 displays the trend plots of the predicted and actual penicillin concentrations using ELM, EnNCLELM_{pls}, and EnSSNCLELM methods, to provide a more intuitive comparison of model performance. It is evident that the predictions of NCLELM have large deviations from the actual values, especially in the fermentation period of 40–150 h. In comparison, EnNCLELM_{pls} significantly reduced the deviations, due to the introduction of ensemble learning. Furthermore, we can see that EnSSNCLELM produced smaller deviations than EnNCLELM_{pls} in the period of 70–180 h, although their prediction accuracies are comparable in other zones. The superior predictions of EnSSNCLELM were mainly due to the introduction of ensemble learning and the utilization of pseudo-labeled data. On the one hand, ensemble learning allows significantly reducing the prediction uncertainty from the base models, which is especially useful for handling the extreme deviations that often occur for a single model. On the other hand, using an enlarged training set can effectively improve the prediction capability of the base models, which is helpful for constructing a good ensemble.

Subsequently, the individual model performance and ensemble prediction results from the proposed SSNCLELM approach are illustrated in Figure 7. Since, EnSSNCLELM is essentially an ensemble method, both the quality of the SSNCLELM base models and the combination scheme are critical to the ensemble construction. First, the base model performance was investigated under the supervised and semi-supervised learning settings. To this end, once a NCLELM model was built, a corresponding SSNCLELM model with the same input weights and biases was trained by the enlarged labeled set. From the figure, it can be observed that the prediction RMSE values of SSNCLELM models were much smaller than those of the NCLELM models, except for in very few cases. Obviously, such significant performance enhancement is entirely attributed to the incorporation of semi-supervised learning. Then, the ensemble prediction performance using different combination strategies was compared. With the NCLELM and SSNCLELM base models, EnNCLELM_{avg} and EnSSNCLELM_{avg} were built using the simple averaging rule, while EnNCLELM_{pls} and EnSSNCLELM employed PLS stacking. One can observe that, both EnNCLELM_{avg} and EnSSNCLELM_{avg} only obtained a prediction accuracy that was inferior to the best base model, which is mainly because the simple averaging ignores the differences among the base models. By contrast, EnNCLELM_{pls} provided a prediction accuracy that was close to that of the best base model, while our proposed EnSSNCLELM approach was greatly superior to all base models. These results further confirm the effectiveness of the proposed pseudo-labeling technique and the PLS stacking strategy.

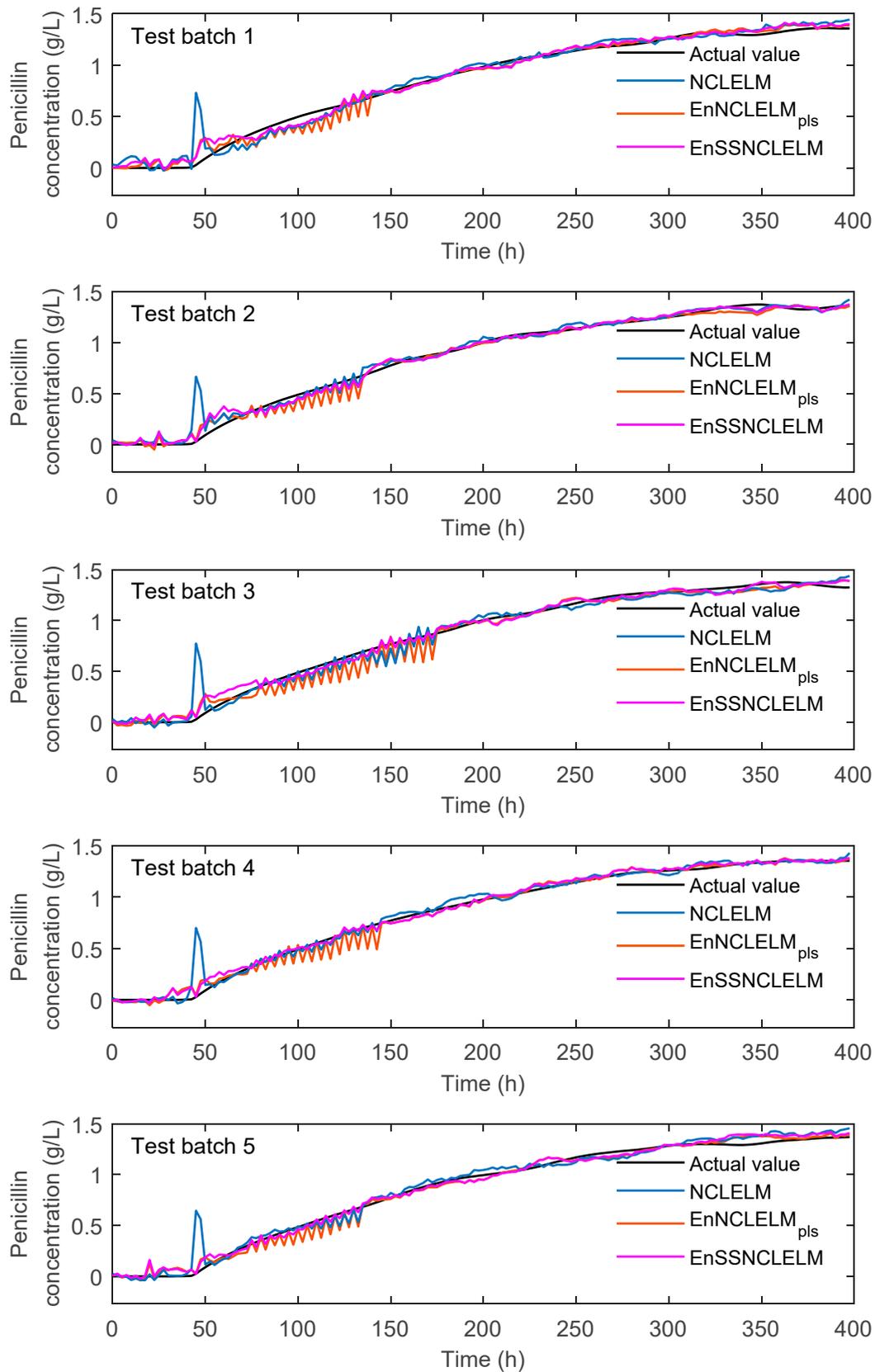


Figure 6. Trend plots of penicillin concentration predictions using ELM, EnNCLELM_{pls}, and EnSSNCLELM methods in penicillin fermentation process.

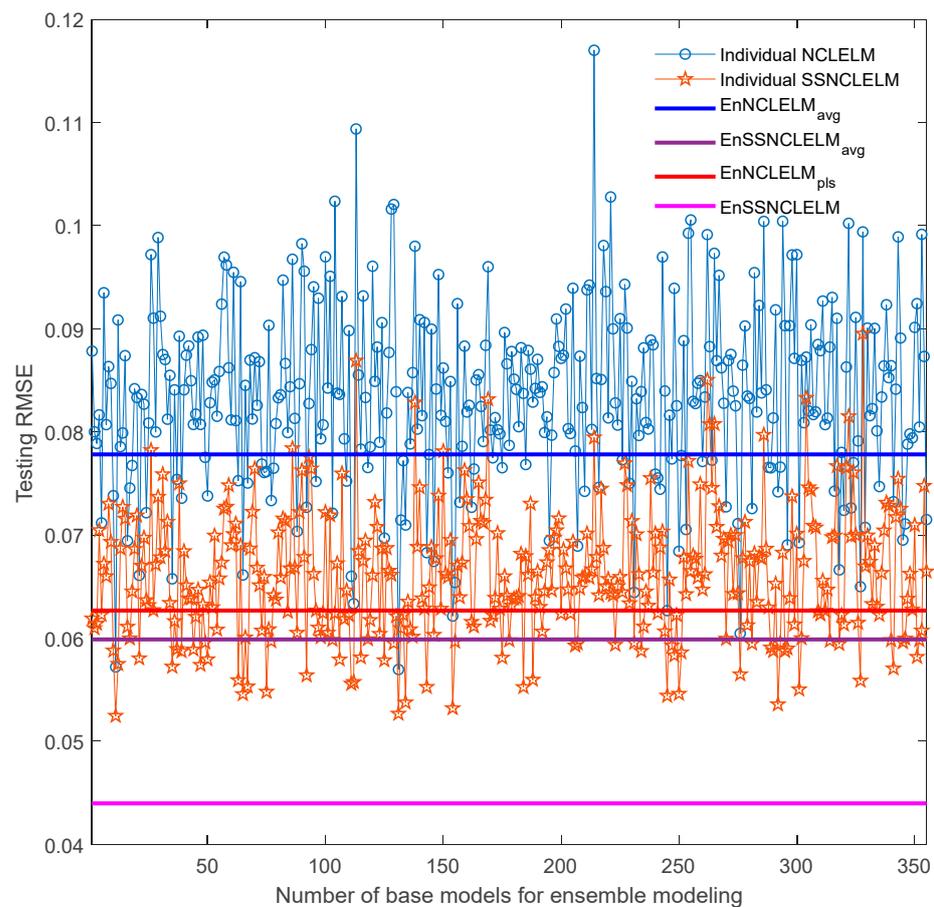


Figure 7. Prediction performance of individual models before and after incorporating semi-supervised learning.

The success of the proposed EnSSNCLELM method depends highly on the quality of the obtained pseudo labels. Thus, one remaining question is how well the pseudo labels can coincide with the actual labels. Although in real-world applications, the actual labels for the unlabeled data are unknown, in this case study, the true labels could be easily acquired from the simulated platform. Hence, as illustrated in Figure 8, we present two types of comparisons: evaluating the fitting degree of the pseudo and true labels, and comparing the testing performance, before and after introducing pseudo-labeled data. As indicated by the prediction results from the two example runs, the pseudo and the actual labels achieved a strong agreement with small prediction RMSE values. Moreover, it is noticeable that $RMSE_{SSNCLELM}$ was much lower than $RMSE_{NCLELM}$, due to the inclusion of pseudo-labeled samples. These observations further imply the effectiveness and reliability of the proposed pseudo-labeling optimization technique.

The above results show that the proposed EnSSNCLELM method can effectively combine semi-supervised learning with ensemble learning, thus allowing resolving the label scarcity issue and improving the estimation reliability of difficult-to-measure variables in an industrial process.

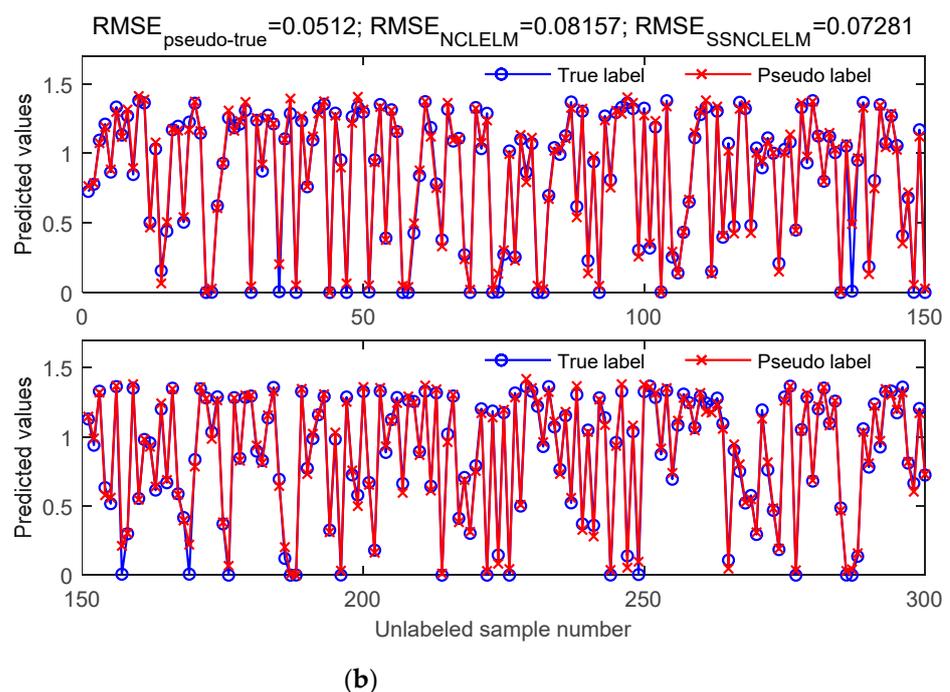
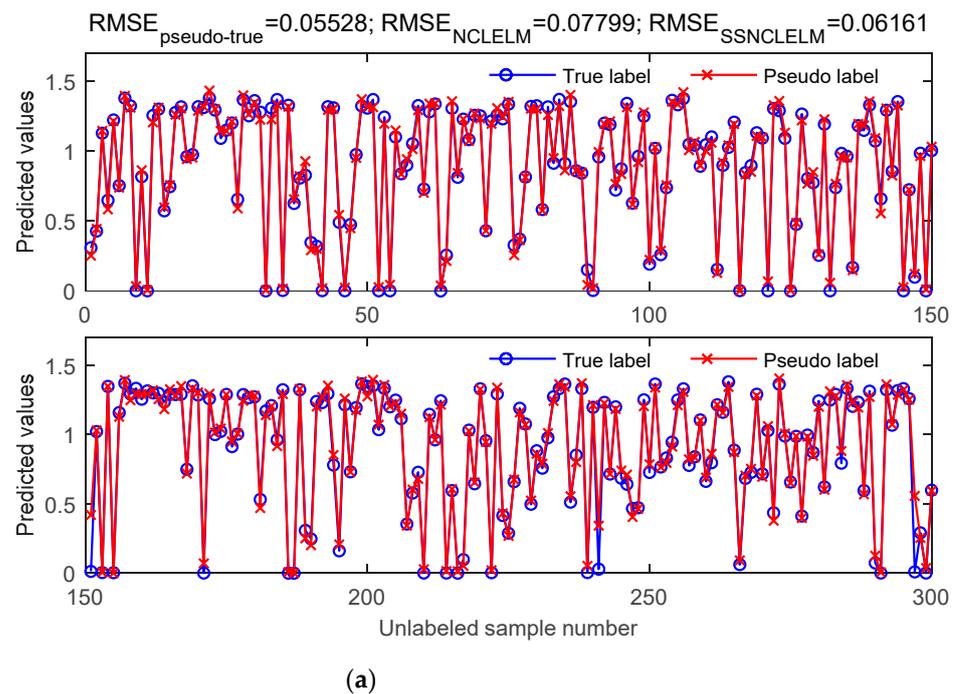


Figure 8. Comparison of the real and pseudo labels estimated by the proposed pseudo-labeling optimization method from (a) Example run 1, and (b) Example run 2. RMSE_{pseudo-true} is the fitting accuracy between the pseudo and true labels, and RMSE_{NCLELM} and RMSE_{SSNCLELM} denote the prediction accuracy of the NCLELM, before and after including the pseudo-labeled data, respectively.

4.2. Application to an Industrial Chlortetracycline Fermentation Process

4.2.1. Process Description

Chlortetracycline (CTC) is a type of broad-spectrum antibiotic, which has been widely used in pharmaceutical and agricultural production and animal husbandry. The industrial CTC fermentation process under study is carried out by Charoen Pokphand Group Co., Ltd (Zhumadian, China), as illustrated in Figure 9 [65]. Generally, CTC is produced through

the cultivation of *Streptomyces aureofaciens* in an air-lift stirred fermenter with a volume of 120 m³. During the fed-batch cultivation process, real-time measurements of substrate concentration are desirable, to facilitate efficient feeding control. However, up to now, the feeding control is often still operated manually, due to the lack of reliable online hardware sensors for substrate concentration measurement, which is usually achieved through offline chemical analysis, with a long delay of 6 h. Thus, soft sensors are desirable to enable real-time estimations of substrate concentration. In this work, the process variables listed in Table 3 were used as inputs for soft sensor development.

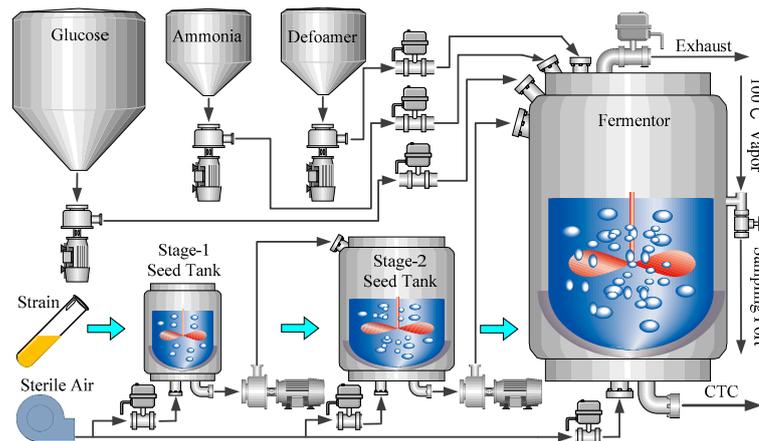


Figure 9. Flow diagram of an industrial fed-batch CTC fermentation process.

Table 3. Input variables for soft sensor development in the industrial CTC fermentation process.

No.	Variable Description	No.	Variable Description
1	Cultivation time (min)	6	Volume of air consumption (m ³)
2	Temperature (°C)	7	Substrate feed rate (L/h)
3	pH	8	Volume of substrate consumption (L)
4	Dissolved oxygen concentration (%)	9	Volume of ammonia consumption (L)
5	Air flow rate (m ³ /h)	10	Volume of culture medium (m ³)

4.2.2. Analysis and Comparison of Prediction Results

In this case study, a total of 14 batches of process data were collected from an industrial fermenter, including 351 labeled samples and 3183 unlabeled samples. Among the labeled samples, 124 were utilized for model training, 29 were used for validation, and the remaining 198 samples served as the testing set. In addition, 3183 unlabeled samples were used for the pseudo labeling optimization and semi-supervised learning. Moreover, the parameter settings for different soft sensor models were the same as those in the application to the penicillin fermentation process. Consequently, a total of 375 SSNCLELM models were obtained, and 264 were selected using the ensemble pruning strategy for the construction of the EnSSNCLELM model.

The substrate concentration prediction results for the different soft sensor methods are presented in Table 4. It is obvious that ELM performed much worse than the other methods, while the proposed EnSSNCLELM method performed best. Compared with the supervised single ELM model, NCLELM achieved a performance enhancement, due to the introduction of NCL ensemble learning. Meanwhile, both CoELM and SSELM also reduced the prediction errors, due to the use of unlabeled data information. Furthermore, by applying an ensemble learning philosophy and using NCLELM as the base learning technique, EnNCLELM_{avg} and EnNCLELM_{pls} delivered a much better estimation performance than the single NCLELM. Comparing EnNCLELM_{pls} and EnNCLELM_{avg} indicates that the PLS stacking outperformed the simple averaging rule. In addition, compared

with EnNCLELM_{pls}, the prediction accuracy of EnSSNCLELM was significantly improved, because of the incorporation of semi-supervised learning. A similar situation also appeared in the performance comparison of EnNCLELM_{avg} and EnSSNCLELM_{avg}. Similar to the application in the penicillin fermentation process, once again, the application results demonstrated the superiority of the proposed method over the traditional supervised and semi-supervised methods.

Table 4. Prediction results of the various soft sensor methods for the CTC fermentation process.

Method	RMSE	R ²
ELM	0.5993	0.8009
NCLELM	0.5214	0.8527
CoELM	0.5033	0.8614
SSELM	0.5566	0.8303
EnNCLELM _{avg}	0.4916	0.8699
EnSSNCLELM _{avg}	0.4654	0.8834
EnNCLELM _{pls}	0.4214	0.9044
EnSSNCLELM	0.3561	0.9317

More intuitively, the scatter plots of the predicted versus actual substrate concentration values using the NCLELM, EnNCLELM_{pls}, and EnSSNCLELM models are depicted in Figure 10. In the scatter plot, the accuracy of the predictive model is evaluated by the closeness of points to the diagonal line. The closer the scatter points are to the diagonal line, the higher the model accuracy. From the figure, one can see at a glance that the test samples exhibit an unbalanced distribution. In the small-value zone (about 1–4), the testing samples are dense, while in the large-value zone (about 4–7), the samples are sparse. Furthermore, it can be readily observed that the NCLELM, EnNCLELM_{pls}, and EnSSNCLELM methods provided a different prediction performance. For the small-value zone, NCLELM suffers from large deviations from the diagonal line, implying a poor prediction accuracy. EnNCLELM_{pls} reduces the deviations by introducing ensemble learning, thus, indicating a significant accuracy improvement. In comparison, EnSSNCLELM achieved the best closeness to the diagonal line, due to the exploitation of unlabeled data, which led to further enhancement of the model performance. Differently, in the large-value zone, one can find an interesting aspect, in that EnNCLELM_{pls} performed even worse than NCLELM. One possible reason is that some of the NCLELM base models for the ensemble construction of EnNCLELM_{pls} were not well learnt, due to the insufficiency of training samples in the large-value zone, and, thus, the combination of such models resulted in a performance deterioration instead of improvement. By contrast, the proposed EnSSNCLELM approach still obtained the most competitive prediction accuracy in the large-value zone, which was mainly because the inclusion of high-confidence pseudo-labeled data significantly improved the generalization capability of the SSNCLELM base models.

In addition, Figure 11 presents the performance of the individual models before and after incorporating semi-supervised learning for the industrial fed-batch CTC fermentation process. One can see that SSNCLELM base models achieved a better performance than the NCLELM base models, except for a few cases, which is attributed to the augmentation of the labeled training data, resulting from the good pseudo-label data. After introducing the stacking combination, EnNCLELM_{pls} delivered a performance close to that of the best base model, while EnSSNCLELM produced much better predictions than all the base models. In comparison, both EnNCLELM_{avg} and EnSSNCLELM_{avg} did not deliver a prediction accuracy enhancement. This was mainly because the simple averaging rule cannot consider the differences among the base models. Overall, the competitive performance of EnSSNCLELM was mainly due to the introduction of semi-supervised learning for enhancing the accuracy and diversity of the base models, as well as the utilization of ensemble learning, to further improve the prediction accuracy and reliability.

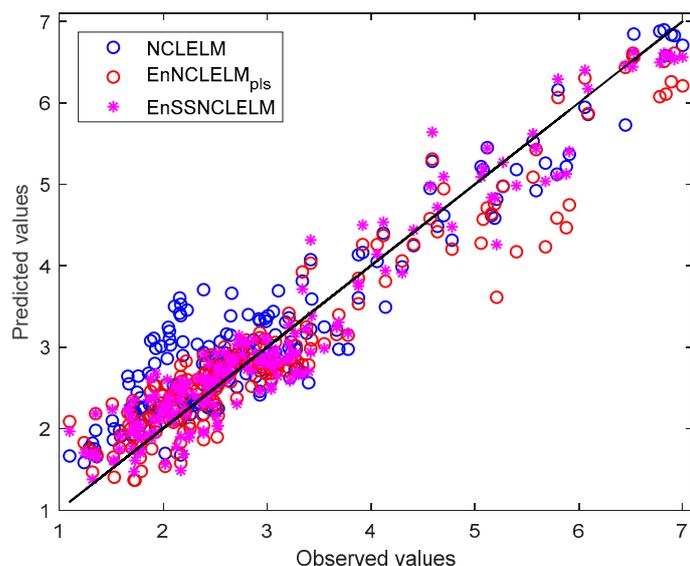


Figure 10. Scatter plots of the observed and predicted substrate concentration using the NCLELM, EnNCLELM_{pls}, and EnSSNCLELM methods.

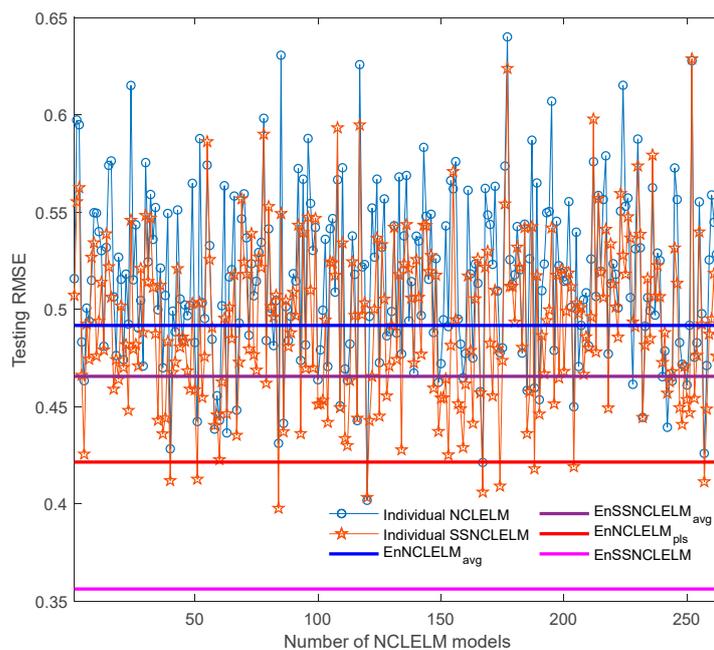


Figure 11. The performance of individual models before and after incorporating semi-supervised learning for industrial fed-batch CTC fermentation process.

The above case study results further indicate the effectiveness of the proposed MLPLO technique in extending the labeled training data and verify the superiority of the proposed EnSSNCLELM method over traditional soft sensor methods, for providing accurate quality variable predictions in industrial processes, especially for application scenarios with limited labeled data.

5. Conclusions

A novel semi-supervised soft sensor modeling method, referred to as EnSSNCLELM, was proposed for the quality variable prediction of industrial processes, where the labeled data are scarce but the unlabeled data are rich. The proposed approach aims to enhance the inferential performance by leveraging both the labeled and unlabeled data, as well as

combining semi-supervised and ensemble learning paradigms to achieve complementary strengths. First, to reduce the prediction instability and improve the prediction accuracy of ELM, the NCLELM algorithm was developed by introducing the NCL rule into ELM modeling. In this way, the diversity among the base ELM models is explicitly considered during NCLELM modeling. Second, by using NCLELM as the base learning technique, we proposed a MLPLO approach to provide high-confidence pseudo-labeled data, where the issue of estimating pseudo labels is formulated as an explicit optimization problem and solved through evolutionary optimization. In addition, based on the extended labeled data, a set of diverse SSNCLELM models were built. Finally, these models were combined using a stacking strategy, after ensemble pruning according to the performance enhancement of semi-supervised models against supervised models. The application results from two chemical processes indicated that the proposed algorithm allows for a significant performance improvement over the traditional supervised and semi-supervised soft sensor methods.

The success of EnSSNCLELM is highly dependent on obtaining high-quality pseudo-labeled data, as well as building and combining diverse and accurate semi-supervised base models. Thus, how to formulate and solve the PLOP problem deserves further studies, e.g., defining more effective optimization objectives and employing multi-objective pseudo-labeling optimization, instead of a single-objective method. Furthermore, it is of great importance to ensure safe learning of the base models, i.e., avoiding reducing learning performance significantly when using unlabeled data. This may be achieved through efficiently selecting well-performing base models for ensemble construction and utilizing informative unlabeled data for semi-supervised modeling. These issues are left to be investigated in the near future.

Author Contributions: Conceptualization, H.J. and Y.L.; methodology, Y.L.; software, Y.L. and B.Y.; validation, Y.L.; investigation, S.D.; data curation, S.D.; writing—original draft preparation, Y.L.; writing—review and editing, H.J. and X.C.; visualization, S.D.; supervision, H.J.; project administration, H.J.; funding acquisition, H.J. and B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Natural Science Foundation of China (NSFC) (grants Nos., 62163019, 61763020 and 61863020) and the Applied Basic Research Project of Yunnan Province (202101AT070096).

Conflicts of Interest: The authors declare no competing financial interest.

References

1. Weber, R.; Brosilow, C. The use of secondary measurements to improve control. *AIChE J.* **1972**, *18*, 614–623. [[CrossRef](#)]
2. Fortuna, L.; Graziani, S.; Rizzo, A.; Xibilia, M.G. *Soft Sensors for Monitoring and Control of Industrial Processes*; Springer Science & Business Media: Berlin, Germany, 2007.
3. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven Soft Sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [[CrossRef](#)]
4. Ge, Z.; Song, Z.; Gao, F. Review of Recent Research on Data-Based Process Monitoring. *Ind. Eng. Chem. Res.* **2013**, *52*, 3543–3562. [[CrossRef](#)]
5. Yin, S.; Li, X.; Gao, H.; Kaynak, O. Data-Based Techniques Focused on Modern Industry: An Overview. *IEEE Trans. Ind. Electron.* **2015**, *62*, 657–667. [[CrossRef](#)]
6. Ge, Z. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemom. Intell. Lab. Syst.* **2017**, *171*, 16–25. [[CrossRef](#)]
7. Qin, S.J.; Chiang, L.H. Advances and opportunities in machine learning for process data analytics. *Comput. Chem. Eng.* **2019**, *126*, 465–473. [[CrossRef](#)]
8. Liu, Y.; Xie, M. Rebooting data-driven soft-sensors in process industries: A review of kernel methods. *J. Process Control* **2020**, *89*, 58–73. [[CrossRef](#)]
9. Zhu, Q.-X.; Zhang, X.-H.; He, Y.-L. Novel Virtual Sample Generation Based on Locally Linear Embedding for Optimizing the Small Sample Problem: Case of Soft Sensor Applications. *Ind. Eng. Chem. Res.* **2020**, *59*, 17977–17986. [[CrossRef](#)]
10. He, Y.-L.; Hua, Q.; Zhu, Q.-X.; Lu, S. Enhanced virtual sample generation based on manifold features: Applications to developing soft sensor using small data. *ISA Trans.* **2021**, in press. [[CrossRef](#)]

11. Liu, Y.; Yang, C.; Liu, K.; Chen, B.; Yao, Y. Domain adaptation transfer learning soft sensor for product quality prediction. *Chemom. Intell. Lab. Syst.* **2019**, *192*, 103813. [[CrossRef](#)]
12. Liu, Y.; Yang, C.; Zhang, M.; Dai, Y.; Yao, Y. Development of Adversarial Transfer Learning Soft Sensor for Multi-Grade Processes. *Ind. Eng. Chem. Res.* **2020**, *59*, 16330–16345. [[CrossRef](#)]
13. Lyu, Y.; Chen, J.; Song, Z. Synthesizing labeled data to enhance soft sensor performance in data-scarce regions. *Control Eng. Pract.* **2021**, *115*, 104903. [[CrossRef](#)]
14. Ge, Z. Active learning strategy for smart soft sensor development under a small number of labeled data samples. *J. Process Control* **2014**, *24*, 1454–1461. [[CrossRef](#)]
15. Tang, Q.; Li, D.; Xi, Y. A new active learning strategy for soft sensor modeling based on feature reconstruction and uncertainty evaluation. *Chemom. Intell. Lab. Syst.* **2018**, *172*, 43–51. [[CrossRef](#)]
16. Zhu, X.; Goldberg, A.B. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2009**, *3*, 1–130. [[CrossRef](#)]
17. Kostopoulos, G.; Karlos, S.; Kotsiantis, S.; Ragos, O. Semi-supervised regression: A recent review. *J. Intell. Fuzzy Syst.* **2018**, *35*, 1483–1500. [[CrossRef](#)]
18. Ge, Z. Semi-supervised data modeling and analytics in the process industry: Current research status and challenges. *IFAC J. Syst. Control* **2021**, *16*, 100150. [[CrossRef](#)]
19. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Mach. Learn.* **2019**, *109*, 373–440. [[CrossRef](#)]
20. Chapelle, O.; Scholkopf, B.; Zien, A. Semi-supervised learning (chappelle, o. et al., eds.; 2006) [book reviews]. *IEEE Trans. Neural Netw.* **2009**, *20*, 542. [[CrossRef](#)]
21. Ge, Z.; Song, Z. Semisupervised Bayesian method for soft sensor modeling with unlabeled data samples. *AIChE J.* **2011**, *57*, 2109–2119. [[CrossRef](#)]
22. Zheng, J.; Song, Z. Semisupervised learning for probabilistic partial least squares regression model and soft sensor application. *J. Process Control* **2018**, *64*, 123–131. [[CrossRef](#)]
23. Sedghi, S.; Sadeghian, A.; Huang, B. Mixture semisupervised probabilistic principal component regression model with missing inputs. *Comput. Chem. Eng.* **2017**, *103*, 176–187. [[CrossRef](#)]
24. Shao, W.; Ge, Z.; Song, Z. Soft-Sensor Development for Processes With Multiple Operating Modes Based on Semisupervised Gaussian Mixture Regression. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 2169–2181. [[CrossRef](#)]
25. Shao, W.; Ge, Z.; Song, Z. Quality variable prediction for chemical processes based on semisupervised Dirichlet process mixture of Gaussians. *Chem. Eng. Sci.* **2019**, *193*, 394–410. [[CrossRef](#)]
26. Shao, W.; Ge, Z.; Song, Z. Semi-supervised mixture of latent factor analysis models with application to online key variable estimation. *Control Eng. Pract.* **2019**, *84*, 32–47. [[CrossRef](#)]
27. Wang, J.; Shao, W.; Zhang, X.; Qian, J.; Song, Z.; Peng, Z. Nonlinear variational Bayesian Student's-t mixture regression and inferential sensor application with semisupervised data. *J. Process Control* **2021**, *105*, 141–159. [[CrossRef](#)]
28. Zhu, X. *Learning from Labeled and Unlabeled Data with Label Propagation*; Tech Report; Carnegie Mellon University: Pittsburgh, PA, USA, 2002.
29. Zheng, W.; Gao, X.; Liu, Y.; Wang, L.; Yang, J.; Gao, Z. Industrial Mooney viscosity prediction using fast semi-supervised empirical model. *Chemom. Intell. Lab. Syst.* **2017**, *171*, 86–92. [[CrossRef](#)]
30. Zheng, W.; Liu, Y.; Gao, Z.; Yang, J. Just-in-time semi-supervised soft sensor for quality prediction in industrial rubber mixers. *Chemom. Intell. Lab. Syst.* **2018**, *180*, 36–41. [[CrossRef](#)]
31. Yan, W.; Guo, P.; Tian, Y.; Gao, J. A Framework and Modeling Method of Data-Driven Soft Sensors Based on Semisupervised Gaussian Regression. *Ind. Eng. Chem. Res.* **2016**, *55*, 7394–7401. [[CrossRef](#)]
32. Yao, L.; Ge, Z. Deep Learning of Semisupervised Process Data with Hierarchical Extreme Learning Machine and Soft Sensor Application. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1490–1498. [[CrossRef](#)]
33. Yan, W.; Xu, R.; Wang, K.; Di, T.; Jiang, Z. Soft Sensor Modeling Method Based on Semisupervised Deep Learning and Its Application to Wastewater Treatment Plant. *Ind. Eng. Chem. Res.* **2020**, *59*, 4589–4601. [[CrossRef](#)]
34. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT press: Cambridge, MA, USA, 2016.
35. Shang, C.; Yang, F.; Huang, D.; Lyu, W. Data-driven soft sensor development based on deep learning technique. *J. Process Control* **2014**, *24*, 223–233. [[CrossRef](#)]
36. Yan, X.; Wang, J.; Jiang, Q. Deep relevant representation learning for soft sensing. *Inf. Sci.* **2020**, *514*, 263–274. [[CrossRef](#)]
37. Yuan, X.; Ou, C.; Wang, Y.; Yang, C.; Gui, W. A novel semi-supervised pre-training strategy for deep networks and its application for quality variable prediction in industrial processes. *Chem. Eng. Sci.* **2020**, *217*, 115509. [[CrossRef](#)]
38. Yuan, X.; Ou, C.; Wang, Y.; Yang, C.; Gui, W. Deep quality-related feature extraction for soft sensing modeling: A deep learning approach with hybrid VW-SAE. *Neurocomputing* **2020**, *396*, 375–382. [[CrossRef](#)]
39. de Lima, J.M.M.; de Araujo, F.M.U. Ensemble Deep Relevant Learning Framework for Semi-Supervised Soft Sensor Modeling of Industrial Processes. *Neurocomputing* **2021**, *462*, 154–168. [[CrossRef](#)]
40. Triguero, I.; García, S.; Herrera, F. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowl. Inf. Syst.* **2015**, *42*, 245–284. [[CrossRef](#)]
41. Rosenberg, C.; Hebert, M.; Schneiderman, H. Semi-Supervised Self-Training of Object Detection Models. In Proceedings of the 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05), Washington, DC, USA, 5–7 January 2005; Volume 1, pp. 29–36.

42. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory—COLT' 98, Association for Computing Machinery, New York, NY, USA, 24–26 July 1998; pp. 92–100.
43. Zhou, Z.-H.; Li, M. Semi-supervised regression with co-training. In *IJCAI*; Morgan Kaufmann: Edinburgh, Scotland, 2005; pp. 908–913.
44. Zhou, Z.-H.; Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1529–1541. [[CrossRef](#)]
45. Gu, S.; Jin, Y. Multi-train: A semi-supervised heterogeneous ensemble classifier. *Neurocomputing* **2017**, *249*, 202–211. [[CrossRef](#)]
46. Li, M.; Zhou, Z.-H. Improve Computer-Aided Diagnosis With Machine Learning Techniques Using Undiagnosed Samples. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2007**, *37*, 1088–1098. [[CrossRef](#)]
47. Kang, P.; Kim, D.; Cho, S. Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Syst. Appl.* **2016**, *51*, 85–106. [[CrossRef](#)]
48. Bao, L.; Yuan, X.; Ge, Z. Co-training partial least squares model for semi-supervised soft sensor development. *Chemom. Intell. Lab. Syst.* **2015**, *147*, 75–85. [[CrossRef](#)]
49. Li, D.; Liu, Y.; Huang, D. Development of semi-supervised multiple-output soft-sensors with Co-training and tri-training MPLS and MRVM. *Chemom. Intell. Lab. Syst.* **2020**, *199*, 103970. [[CrossRef](#)]
50. Lv, M.; Li, Y.; Chen, L.; Chen, T. Air quality estimation by exploiting terrain features and multi-view transfer semi-supervised regression. *Inf. Sci.* **2019**, *483*, 82–95. [[CrossRef](#)]
51. Liang, Y.; Liu, Z.; Liu, W. A co-training style semi-supervised artificial neural network modeling and its application in thermal conductivity prediction of polymeric composites filled with BN sheets. *Energy AI* **2021**, *4*, 100052. [[CrossRef](#)]
52. Kaneko, H.; Funatsu, K. Adaptive soft sensor based on online support vector regression and Bayesian ensemble learning for various states in chemical plants. *Chemom. Intell. Lab. Syst.* **2014**, *137*, 57–66. [[CrossRef](#)]
53. Jin, H.; Chen, X.; Wang, L.; Yang, K.; Wu, L. Adaptive Soft Sensor Development Based on Online Ensemble Gaussian Process Regression for Nonlinear Time-Varying Batch Processes. *Ind. Eng. Chem. Res.* **2015**, *54*, 7320–7345. [[CrossRef](#)]
54. Liu, Y.; Huang, D.; Liu, B.; Feng, Q.; Cai, B. Adaptive ranking based ensemble learning of Gaussian process regression models for quality-related variable prediction in process industries. *Appl. Soft Comput.* **2021**, *101*, 107060. [[CrossRef](#)]
55. Chen, X.; Mao, Z.; Jia, R.; Zhang, S. Ensemble regularized local finite impulse response models and soft sensor application in nonlinear dynamic industrial processes. *Appl. Soft Comput.* **2019**, *85*, 105806. [[CrossRef](#)]
56. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms*; Taylor & Francis: Boca Raton, FL, USA, 2012.
57. Fazakis, N.; Karlos, S.; Kotsiantis, S.; Sgarbas, K. A multi-scheme semi-supervised regression approach. *Pattern Recognit. Lett.* **2019**, *125*, 758–765. [[CrossRef](#)]
58. Zhang, M.-L.; Zhou, Z.-H. Exploiting unlabeled data to enhance ensemble diversity. *Data Min. Knowl. Discov.* **2011**, *26*, 98–129. [[CrossRef](#)]
59. Sun, Q.; Ge, Z. Deep Learning for Industrial KPI Prediction: When Ensemble Learning Meets Semi-Supervised Data. *IEEE Trans. Ind. Inform.* **2021**, *17*, 260–269. [[CrossRef](#)]
60. Shao, W.; Tian, X. Semi-supervised selective ensemble learning based on distance to model for nonlinear soft sensor development. *Neurocomputing* **2017**, *222*, 91–104. [[CrossRef](#)]
61. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
62. Liu, Y.; Yao, X. Ensemble learning via negative correlation. *Neural Netw.* **1999**, *12*, 1399–1404. [[CrossRef](#)]
63. He, Y.-L.; Geng, Z.-Q.; Zhu, Q.-X. Data driven soft sensor development for complex chemical processes using extreme learning machine. *Chem. Eng. Res. Des.* **2015**, *102*, 1–11. [[CrossRef](#)]
64. Pan, B.; Jin, H.; Yang, B.; Qian, B.; Zhao, Z. Soft Sensor Development for Nonlinear Industrial Processes Based on Ensemble Just-in-Time Extreme Learning Machine through Triple-Modal Perturbation and Evolutionary Multiobjective Optimization. *Ind. Eng. Chem. Res.* **2019**, *58*, 17991–18006. [[CrossRef](#)]
65. Jin, H.; Pan, B.; Chen, X.; Qian, B. Ensemble just-in-time learning framework through evolutionary multi-objective optimization for soft sensor development of nonlinear industrial processes. *Chemom. Intell. Lab. Syst.* **2019**, *184*, 153–166. [[CrossRef](#)]
66. Shi, X.; Kang, Q.; An, J.; Zhou, M. Novel L1 Regularized Extreme Learning Machine for Soft-Sensing of an Industrial Process. *IEEE Trans. Ind. Inform.* **2021**, *18*, 1009–1017. [[CrossRef](#)]
67. Brown, G.; Wyatt, J.L.; Tino, P.; Bengio, Y. Managing diversity in regression ensembles. *J. Mach. Learn. Res.* **2005**, *6*, 1621–1650.
68. Alhamdoosh, M.; Wang, D. Fast decorrelated neural network ensembles with random weights. *Inf. Sci.* **2014**, *264*, 104–117. [[CrossRef](#)]
69. Chen, H.; Jiang, B.; Yao, X. Semisupervised Negative Correlation Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5366–5379. [[CrossRef](#)] [[PubMed](#)]
70. Jin, H.; Li, Z.; Chen, X.; Qian, B.; Yang, B.; Yang, J. Evolutionary optimization based pseudo labeling for semi-supervised soft sensor development of industrial processes. *Chem. Eng. Sci.* **2021**, *237*, 116560. [[CrossRef](#)]
71. Zhou, Z.-H.; Wu, J.; Tang, W. Ensembling neural networks: Many could be better than all. *Artif. Intell.* **2002**, *137*, 239–263. [[CrossRef](#)]
72. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning, Print*; MIT Press: Cambridge, MA, USA, 2008. [[CrossRef](#)]

73. Bansal, J.C.; Singh, P.K.; Pal, N.R. *Evolutionary and Swarm Intelligence Algorithms*; Springer: Berlin/Heidelberg, Germany, 2019.
74. Dasgupta, D.; Michalewicz, Z. *Evolutionary Algorithms in Engineering Applications*; Springer Science & Business Media: New York, NY, USA, 2013.
75. Huang, L.; Deng, X.; Bo, Y.; Zhang, Y.; Wang, P. Evolutionary optimization assisted delayed deep cycle reservoir modeling method with its application to ship heave motion prediction. *ISA Trans.* **2021**, in press. [[CrossRef](#)] [[PubMed](#)]
76. Ma, M.; Sun, C.; Mao, Z.; Chen, X. Ensemble deep learning with multi-objective optimization for prognosis of rotating machinery. *ISA Trans.* **2021**, *113*, 166–174. [[CrossRef](#)]
77. Haidong, S.; Ziyang, D.; Junsheng, C.; Hongkai, J. Intelligent fault diagnosis among different rotating machines using novel stacked transfer auto-encoder optimized by PSO. *ISA Trans.* **2020**, *105*, 308–319. [[CrossRef](#)] [[PubMed](#)]
78. Jian, J.-R.; Zhan, Z.-H.; Zhang, J. Large-scale evolutionary optimization: A survey and experimental comparative study. *Int. J. Mach. Learn. Cybern.* **2019**, *11*, 729–745. [[CrossRef](#)]
79. Shao, W.; Tian, X. Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models. *Chem. Eng. Res. Des.* **2015**, *95*, 113–132. [[CrossRef](#)]
80. Jin, H.; Shi, L.; Chen, X.; Qian, B.; Yang, B.; Jin, H. Probabilistic wind power forecasting using selective ensemble of finite mixture Gaussian process regression models. *Renew. Energy* **2021**, *174*, 1–18. [[CrossRef](#)]
81. Li, Y.-F.; Liang, D.-M. Safe semi-supervised learning: A brief introduction. *Front. Comput. Sci.* **2019**, *13*, 669–676. [[CrossRef](#)]
82. Yu, J. Multiway Gaussian Mixture Model Based Adaptive Kernel Partial Least Squares Regression Method for Soft Sensor Estimation and Reliable Quality Prediction of Nonlinear Multiphase Batch Processes. *Ind. Eng. Chem. Res.* **2012**, *51*, 13227–13237. [[CrossRef](#)]
83. Jin, H.; Chen, X.; Yang, J.; Zhang, H.; Wang, L.; Wu, L. Multi-model adaptive soft sensor modeling method using local learning and online support vector regression for nonlinear time-variant batch processes. *Chem. Eng. Sci.* **2015**, *131*, 282–303. [[CrossRef](#)]
84. Birol, G.; Ündey, C.; Cinar, A. A modular simulation package for fed-batch fermentation: Penicillin production. *Comput. Chem. Eng.* **2002**, *26*, 1553–1565. [[CrossRef](#)]