ACS OMEGA

http://pubs.acs.org/journal/acsodf

Article

Deep Learning for Operating Performance Assessment of Industrial Processes with Layer Attention-Based Stacked Performance-Relevant Denoising Auto-Encoders

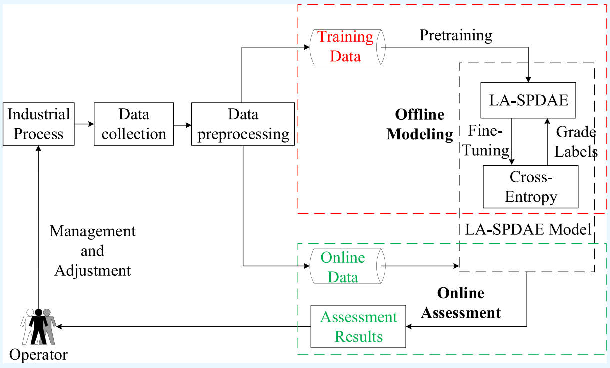
Yan Liu,* Zhe Ma, and Fuli Wang

Cite This: *ACS Omega* 2023, 8, 14583−14596

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: The operating performance of a process may degenerate due to process interferences and operation errors, which cancel the benefits of technology design and economic production. Traditional operating performance assessment methods are either lack of real-time due to the post-analysis or difficult to distinguish performance grades for process data with weak differences and strong noise interferences based on shallow learning structures. In this paper, a new layer attention-based stacked performance-relevant denoising auto-encoder (LA-SPDAE) is proposed for the operating performance assessment of industrial processes. It overcomes the defect that the original SDAE ignores task-relevant information in training and only uses the feature of the last hidden layer to complete special tasks. In this study, the original SDAE is improved by optimizing the cross-entropy loss of the performance grade labels in the layer-wise pretraining, which is named stacked performance-relevant denoising auto-encoder (SPDAE), and the performance-relevant features can be extracted under supervision. Moreover, for making good use of performance-relevant features of each layer, they are fused by adaptive weights based on the layer attention mechanism. In the case study of cyanide leaching, the assessment accuracy of the proposed LA-SPDAE model is up to 99.85% under the corrupted proportion of 20%, and the advantage is still maintained as the proportion increases to 80%, which demonstrates the superiority of LA-SPDAE compared with conventional deep neural networks and shallow learning structures.



1. INTRODUCTION

With increasing competition for marketing and shortage of raw material resources, modern enterprises pay more attention to maximizing comprehensive economic benefits by optimizing operation and production management,[1] and the operating performance assessment of industrial processes has received considerable attention in applications and academia during the past decades.[2,3] Through the real-time assessment of the process operating performance, the enterprise can grasp the operation of the production timely and make adjustments at the initial stage of performance degradation[4] so as to avoid unnecessary economic loss and ensure the improvement of benefits.

Being different from process monitoring, which is to distinguish between normal and fault,[5] the process operating performance assessment is to identify the performance grades, such as optimal, fine, general, and poor, under the normal conditions. The potential problem posed by this difference is that the differences among different performance grades are significantly less than those between normal and fault conditions, so that the methods with stronger feature discrimination may be more suitable for the process operating performance assessment. Additionally, due to complex production environment and control strategy, the actual industrial data are often disturbed by serious noise,[6] which significantly increases the difficulty in distinguishing different performance grades. As a result, both the weak differences and strong noise interferences bring great challenges to evaluating the operating performance of industrial processes.

With continued advancements of sensors and data storage equipment, a large amount of process data can be obtained more easily, and many data-driven assessment methods with respect to the process operating performance have been developed on the basis of multivariate statistics and data distribution description techniques. For linear industrial processes, i.e., the correlation between process variables is linear or nearly linear, some traditional multivariate statistical analysis methods were improved to accurately extract the features related to the

Received: January 20, 2023
Accepted: March 30, 2023
Published: April 10, 2023

https://doi.org/10.1021/acsomega.3c00414

performance index[7,8] and applied to the process operating performance assessment.[9,10] However, complex process mechanism and fluctuating external environment often lead to nonlinear correlation between process variables, making those linear assessment methods perform poorly, and some nonlinear assessment methods came into being.[11,12] As a result of the load change, feedstock variation, and parameter drift, many industrial processes are multimodal, and operating performance assessment methods based on data distribution description were sequentially proposed to handle this characteristic.[2,13] Concerning the process dynamic characteristics, some conjoint analysis assessment methods were proposed to capture the representative features from industrial data.[14,15] In practice production, some qualitative information described by state descriptions or semantics is the useful information in improving the performance of assessment models. In light of that, both of the qualitative and quantitative information were used to evaluate the operating performance of industrial processes.[16] All of the aforementioned methods provide important solutions for the operating performance assessment of various process characteristics and have achieved satisfactory results in the cases where data of different performance grades are significantly different. However, from the perspective of model structure, these assessment methods are shallow learning methods because the features are only extracted once. It is hard to meet the requirements for the assessment task in terms of the noise reduction capability and feature extraction depth, posing a challenge for extracting discriminative and representative deep features from the process data with weak differences and strong noise interferences.

As one of the most brilliant branches in machine learning, deep learning (DL) methods have thrived in recent years.[17,18] A growing number of researchers try to extract deep features from different kinds of information based on DL technologies, including convolutional neural networks,[19] stacked autoencoder (SAE),[20] long short-term memory,[21] and so on. Among numerous DL methods, SAE is widely used in deep feature extraction because of its strong abilities to deal with the nonlinearity and reveal the low-dimensional essence structure of high-dimensional data. As a modified version of SAE in robustness, the stacked denoising auto-encoders (SDAE) were developed.[22] SDAE is stacked by multiple denoising autoencoders (DAEs) hierarchically. Because the SDAE learns to remove noise and recover input data from their corrupted version, the intermediate features extracted by SDAE- are more robust than those extracted by SAE-. Since SDAEs was proposed, it has been applied in many different areas, including process monitoring,[23] fault isolation,[24] defect recognition,[25] and so on.

Nevertheless, SDAE seeks to reconstruct the input data rather than working on representing special tasks through unsupervised learning, so it is unable to learn task-relevant features completely, and many task-irrelevant features may be extracted. The irrelevant information could spread to higher layers during the construction of a deep neural network (DNN). To improve the network performance, the guidance of task information is of great significance, and some semi-supervised and supervised SAE emerged.[26−28] However, due to ignoring data corruption in the pretraining process, these methods still have weak abilities to deal with industrial data disturbed by strong noise.

Another noteworthy issue is that although a higher hidden layer tends to extract more abstract features, the features of lower hidden layers still have utilization values. That is to say, features of different hidden layers of a DNN are characterizations of input data at different levels, having different contributions for special tasks. To make rational use of features of each hidden layer, evaluating the contributions and controlling the amount of information are very important. The attention mechanism is a common means to measure the importance of information. It can be considered as to screen out critical information from the input data and focus on it for making an effective decision. So far, the attention mechanism has been applied to a wide variety of tasks, such as speech recognition,[29] industrial soft sensing,[30] classification problem,[31] and so on.

For the operating performance assessment of industrial processes, not all process information is closely related to the process operating performance. Conversely, some information may be useless and even interferes with distinguishing performance grades. Moreover, since the actual industrial data are easily polluted by noise, a robust feature extraction method is needed to improve the anti-interference of the assessment model. In addition, although the feature of the last hidden layer is the high abstraction and conciseness of the input data, features of other hidden layers still have valuable information for performance assessment. Given the above, a novel assessment strategy called layer attention-based stacked performance-relevant DAE (LA-SPDAE) is proposed for the operating performance assessment of industrial processes. First, by optimizing the cross-entropy loss of performance grade labels in the layer-wise pretraining procedure, each performance-relevant DAE (PDAE) extracts the features closely related to the process operating performance from input data under supervision. At the same time, by corrupting a fraction of input data, PDAE gives a more robust feature expression of original inputs and enhances the ability for feature extraction. After that, multiple PDAEs are stacked to form the stacked PDAEs (SPDAE). The performance-relevant features extracted by the former PDAE are used as the inputs of the later one, so that deep performance-relevant features are extracted gradually. After the pretraining procedure, instead of utilizing the performance-relevant feature of the last layer, layer attention weights are calculated to make full use of performance-relevant features of all layers, and the assessment results are given based on their adaptive integration.

The contributions of this paper are three-fold: (1) a novel supervised network structure introducing the performance grade labels into DAE is proposed and named PDAE; (2) by stacking multiple PDAEs layer by layer, a SPDAE model is constructed, having the ability of extracting deep performance-relevant features; and (3) the attention mechanism is adopted for fusing performance-relevant features of every hidden layer of the SPDAE adaptively, which improves the performance of the assessment model.

The rest of this paper is organized as follows. In Section 2, the original SDAE and the principle of the attention mechanism are introduced briefly. Then the LA-SPDAE-based operating performance assessment method is proposed in Section 3, which includes the model structure and training strategy. In Section 4, the effectiveness of the proposed assessment approach is illustrated by a case of gold cyanide leaching process, and its superiority is also verified as opposed to some related models. The conclusions and potential future development are elaborated in Section 5.

https://doi.org/10.1021/acsomega.3c00414

2. RELATED WORKS

2.1. Stacked Denoising Auto-Encoders. SDAE is a DNN constructed by stacking several DAEs layer by layer. It is a variant of the original SAE in terms of robustness. DAE is an unsupervised network including one encoder and one decoder. It is trained to reconstruct a clean "repaired" input from its corrupted version. The common types of corruptions include additive isotropic Gaussian noise, masking noise (force some elements to zero randomly), and salt-and-pepper noise (set some elements to their maximum or minimum randomly).[22] The structure of the DAE is shown in Figure 1.

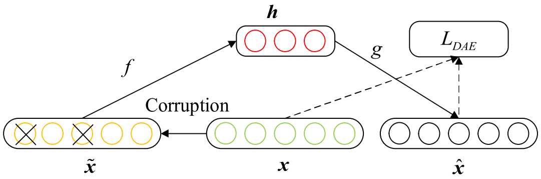


Figure 1. Structure of DAE.

Take an SDAE network with K hidden layers as an example. $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]^T \in R^{N\times J_x}$ is the training data with N samples and J_x variables, and $\boldsymbol{x}_n \in R^{J_x\times 1}$ is the nth sample of $\boldsymbol{X}$. The training of the SDAE contains two procedures, i.e., pretraining hierarchically and global fine-tuning. In the procedure of the pretraining, the input data $\boldsymbol{X}$ are first corrupted as $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, ..., \tilde{\boldsymbol{x}}_N]^T \in R^{N\times J_x}$. Then, the encoder maps $\tilde{\boldsymbol{X}}$ to the hidden layer to obtain the features $\boldsymbol{H}^1 = [\boldsymbol{h}_1^1, \boldsymbol{h}_2^1, ..., \boldsymbol{h}_N^1]^T \in R^{N\times J_{h_1}}$, where J_{h_1} is the number of hidden layer nodes of the first DAE, and the decoder implements self-reconstruction from $\boldsymbol{H}^1$ to the output layer to obtain the reconstructed input data $\hat{\boldsymbol{X}}^1 = [\hat{\boldsymbol{x}}_1^1, \hat{\boldsymbol{x}}_2^1, ..., \hat{\boldsymbol{x}}_N^1]^T \in R^{N\times J_x}$ determined by the information loss.

The encoder and decoder processes in the first DAE are listed as follows

$$\boldsymbol{h}_n^1 = f(\boldsymbol{W}_e^1\tilde{\boldsymbol{x}}_n + \boldsymbol{b}_e^1),\ n = 1, 2, ..., N \tag{1}$$

$$\hat{\boldsymbol{x}}_n = g(\boldsymbol{W}_d^1\boldsymbol{h}_n^1 + \boldsymbol{b}_d^1) \tag{2}$$

where $f(\cdot)$ and $g(\cdot)$ are activation functions; the sigmoid, hyperbolic tangent, and rectified linear unit (ReLU) are common activation functions in a neural network; $\boldsymbol{W}_e^1 \in R^{J_{h_1}\times J_x}$ and $\boldsymbol{b}_e^1 \in R^{J_{h_1}}$ are the weight matrix and bias vector connecting the input and hidden layers of the first DAE; $\boldsymbol{W}_d^1 \in R^{J_x\times J_{h_1}}$ and $\boldsymbol{b}_d^1 \in R^{J_x}$ are the weight matrix and bias vector connecting the hidden and output layers of the first DAE. Then, the parameter set of the first DAE is denoted as $\theta_{\mathrm{DAE}}^1 = \{\boldsymbol{W}_e^1, \boldsymbol{b}_e^1; \boldsymbol{W}_d^1, \boldsymbol{b}_d^1\}$, which can be learned by minimizing the reconstruction error loss function with the backpropagation (BP) algorithm as follows

$$L_{\mathrm{DAE}}^1(\theta_{\mathrm{DAE}}^1) = \frac{1}{N}\sum_{n=1}^{N} \|\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n\|^2. \tag{3}$$

When the first encoder converges, it is fixed temporarily. Then, the first-layer hidden features $\boldsymbol{H}^1$ are corrupted as $\tilde{\boldsymbol{H}}^1 = [\tilde{\boldsymbol{h}}_1^1, \tilde{\boldsymbol{h}}_2^1, ..., \tilde{\boldsymbol{h}}_N^1]^T$ and used as the inputs to train the second DAE for obtaining the parameter set $\theta_{\mathrm{DAE}}^2 = \{\boldsymbol{W}_e^2, \boldsymbol{b}_e^2; \boldsymbol{W}_d^2, \boldsymbol{b}_d^2\}$ and the second-layer hidden features $\boldsymbol{H}^2 = [\boldsymbol{h}_1^2, \boldsymbol{h}_2^2, ..., \boldsymbol{h}_N^2]^T$. As an analogy, all K DAEs can be pretrained progressively.

After all layers have been pretrained, the parameter set $\theta_{\mathrm{SDAE}} = \{\boldsymbol{W}_e^1, \boldsymbol{b}_e^1; \boldsymbol{W}_e^2, \boldsymbol{b}_e^2; ...; \boldsymbol{W}_e^K, \boldsymbol{b}_e^K\}$ is used as the initialization of the SDAE network. Then, a BP-based global fine-tuning is implemented to adjust the network parameters simultaneously. In this way, because the parameters are pretrained, the network falling into local extremum can be largely avoided.[32,33] Once an SDAE has been built, final features of the raw input data are obtained from the last hidden layer and used as the input of stand-alone supervised learning algorithms for complex practical tasks.

2.2. Attention Mechanism. Recently, attention mechanism-based technologies are more and more widely used for various industrial applications. Attention itself can be seen as a soft selection over the input. In this way, the model can focus on inputs that are important for special tasks. As a consequence, the model can handle inputs with a certain degree of redundancy without reducing performance.

An attention network is actually a kind of fully connected neural network, whose input is a vector and the output is a real value between 0 and 1 or a real vector with values between 0 and 1. Mathematically, the attention values can be expressed by eqs 4–6 as follows

$$\boldsymbol{h}_m = \tilde{f}(\boldsymbol{x}),\ m = 1, 2, ..., M \tag{4}$$

$$e_m = \boldsymbol{v}\cdot \tanh(\boldsymbol{W}\boldsymbol{h}_m + \boldsymbol{b}) \tag{5}$$

$$w_m = \frac{|e_m|}{\sum_{j=1}^{M} |e_j|} \tag{6}$$

where $\boldsymbol{h}_m$ represents the mth feature extracted from input data $\boldsymbol{x}$ through a network $\tilde{f}(\cdot)$; tanh$(\cdot)$ is the hyperbolic activation function of the attention network; $\boldsymbol{v}$, $\boldsymbol{W}$, and $\boldsymbol{b}$ are the parameters to be learned; e_m is the attention value of feature $\boldsymbol{h}_m$ and between 0 and 1; and w_m is the normalized attention value and named attention weight here. Then, all features extracted from $\boldsymbol{x}$ can be integrated by attention weights as follows

$$\boldsymbol{h} = \sum_{m=1}^{M} w_m\boldsymbol{h}_m \tag{7}$$

In eq 7, the more critical the feature is, the greater the attention weight is.

3. LAYER ATTENTION-BASED STACKED PDAE FOR PROCESS OPERATION PERFORMANCE ASSESSMENT

The SDAE can robustly learn the high-level features which are considered as abstract representations of raw input data. For the operating performance assessment, however, not all input information is related with the process operating performance. Since the goal of SDAE is to reconstruct the input data rather than the representation of process operating performance due to its unsupervised pretraining, it is difficult to learn discriminative performance-relevant features, and some performance-irrelevant features may still be retained in the extracted features. These performance-irrelevant features may spread to the higher layers along with the layer-wise pretraining and degrade the performance of the assessment model. In order to improve the ability of operating performance assessment, an excellent assessment model should have the ability to identify the performance-relevant features and remove performance-irrelevant ones accurately. Moreover, the performance-relevant features of different layers are exactly different representations and abstractions about the operating performance at different levels. Those features play different roles in operating performance assessment and should be used rationally with a tendency. However, SDAE only uses the last layer hidden feature to implement the output modeling task. There is still much space to improve the utilization of the information in different layers of a DNN, helping to better evaluate the process operating performance. In light of that, the LA-SPDAE is proposed for process operating performance assessment in this section. In each PDAE, its difference from the DAE is that the performance grade labels are added to the output layer and used for guiding the feature extraction instead of reconstructing the input data, which produces the performance-relevant features from its input data under the constraint that the performance grade should be well estimated. Furthermore, through stacking multiple PDAEs layer by layer, hierarchical performance-relevant features are consecutively extracted from the raw input data. To make full use of performance-relevant features of different layers, the layer attention mechanism is imposed on them automatically. With the layer attention weights, performance-relevant features of different layers are further integrated to obtain the final features for the operating performance assessment.

3.1. Performance-Relevant Denoising Auto-Encoders.

The PDAE is composed of the input, hidden, and output layers, whose network structure is shown in Figure 2. Assume that there are C performance grades, and the performance grade label is denoted as $\boldsymbol{y} = [y_1, y_2, ..., y_C]^T \in R^C$. For a sample of performance grade c, $c = 1, 2,..., C$, its label is $[0, ..., 0, \underset{c^{\text{th}}}{1}, 0, ..., 0]^T$. Use $\{\boldsymbol{W}_e^P, \boldsymbol{b}_e^P\}$ and $\{\boldsymbol{W}_d^P, \boldsymbol{b}_d^P\}$ to separately represent the parameters in the encoder and decoder of the PDAE. The encoder of the PDAE is the same as that listed in eq 1, and the decoder is listed as follows

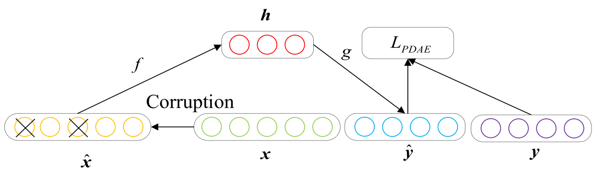


Figure 2. Structure of PDAE.

$$\hat{\boldsymbol{y}} = \tilde{g}(\boldsymbol{W}_d^P\boldsymbol{h} + \boldsymbol{b}_d^P) \tag{8}$$

where $\tilde{g}(\cdot)$ is the activation function.

For the multiple performance grades problem, the SoftMax function is usually used as the activation function and given below

$$\tilde{g}(\boldsymbol{x}) = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_C \end{bmatrix} = \begin{bmatrix} p(\hat{y} = 1|\boldsymbol{h},\boldsymbol{\theta}_{\text{SM}}) \\ p(\hat{y} = 2|\boldsymbol{h},\boldsymbol{\theta}_{\text{SM}}) \\ \vdots \\ p(\hat{y} = C|\boldsymbol{h},\boldsymbol{\theta}_{\text{SM}}) \end{bmatrix} = \frac{1}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^T\boldsymbol{h})} \begin{bmatrix} \exp(\boldsymbol{\theta}_1^T\boldsymbol{h}) \\ \exp(\boldsymbol{\theta}_2^T\boldsymbol{h}) \\ \vdots \\ \exp(\boldsymbol{\theta}_C^T\boldsymbol{h}) \end{bmatrix} \tag{9}$$

where $\hat{y}_c = p(\hat{y} = c|\boldsymbol{h},\boldsymbol{\theta}_{\text{SM}}) = \exp(\boldsymbol{\theta}_c^T\boldsymbol{h})/\sum_{c=1}^{C}\exp(\boldsymbol{\theta}_c^T\boldsymbol{h})$ denotes the posterior probability of $\boldsymbol{h}$ belonging to the cth performance grade; $\boldsymbol{\theta}_{\text{SM}} = [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, ..., , \boldsymbol{\theta}_C^T]^T$ is the parameter set of the SoftMax classifier, and $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_C]^T$.

For the training data $\boldsymbol{X}$, the performance grade labels are denoted as $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N]^T \in R^{N\times C}$, where $\boldsymbol{y}_n = [y_n^1, y_n^2, ..., y_n^C]^T$, and then the parameter set of the PDAE, i.e., $\theta_{\text{PDAE}} = \{\boldsymbol{W}_e^P, \boldsymbol{b}_e^P; \boldsymbol{W}_d^P, \boldsymbol{b}_d^P\}$, is trained by the BP algorithm through minimizing the following cross-entropy loss function

$$L_{\text{PDAE}}(\theta_{\text{PDAE}}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_n^c \log(\hat{y}_n^c) \tag{10}$$

where $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, ..., \hat{\boldsymbol{y}}_N]^T \in R^{N\times C}$ are the posterior probabilities of $\boldsymbol{X}$ belonging to each performance grade, and $\hat{\boldsymbol{y}}_n = [\hat{y}_n^1, \hat{y}_n^2, ..., \hat{y}_n^C]^T$.

Compared with the DAE, PDAE changes the focus of feature representations from reconstructing input data to learning unique characteristics of each performance grade, which is helpful to improve the discrimination ability of the assessment model.

3.2. Stacked Performance-Relevant Denoising Auto-Encoders.

Considering the fact that a single PDAE is not enough to extract deep performance-relevant features from the input data with strong noise, several PDAEs will be stacked hierarchically to constructed the SPDAE network.

Suppose that there are K layers in a SPDAE network. Initially, input data $\boldsymbol{X}$ and performance grade labels $\boldsymbol{Y}$ are used to pretrain the first PDAE, and then the first-layer performance-relevant features $\boldsymbol{H}_P^1 = [\boldsymbol{h}_1^{P,1}, \boldsymbol{h}_2^{P,1}, ..., \boldsymbol{h}_N^{P,1}]^T$ are obtained, where $\boldsymbol{h}_n^{P,1}$ presents the performance-relevant feature extracted from $\boldsymbol{x}_n$. $\hat{\boldsymbol{Y}}^1 = [\hat{\boldsymbol{y}}_1^1, \hat{\boldsymbol{y}}_2^1, ..., \hat{\boldsymbol{y}}_N^1]^T$ denote the posterior probabilities estimated by the first PDAE.

The loss function of the first PDAE is as follows

$$L_{\text{PDAE}}^1(\theta_{\text{PDAE}}^1) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_n^c \log(\hat{y}_n^{1,c}) \tag{11}$$

where $\theta_{\text{PDAE}}^1 = \{\boldsymbol{W}_e^{P,1}, b_e^{P,1}; \boldsymbol{W}_d^{P,1}, b_d^{P,1}\}$ is the parameter set of the first PDAE, and $\hat{y}_n^{1,c}$ is the cth element of $\hat{\boldsymbol{y}}_n^1$, $n = 1, 2, ..., N$. Then, the first-layer performance-relevant features $\boldsymbol{H}_P{}^1$ and performance grade labels $\boldsymbol{Y}$ are used to pretrain the second PDAE, obtaining the parameter set

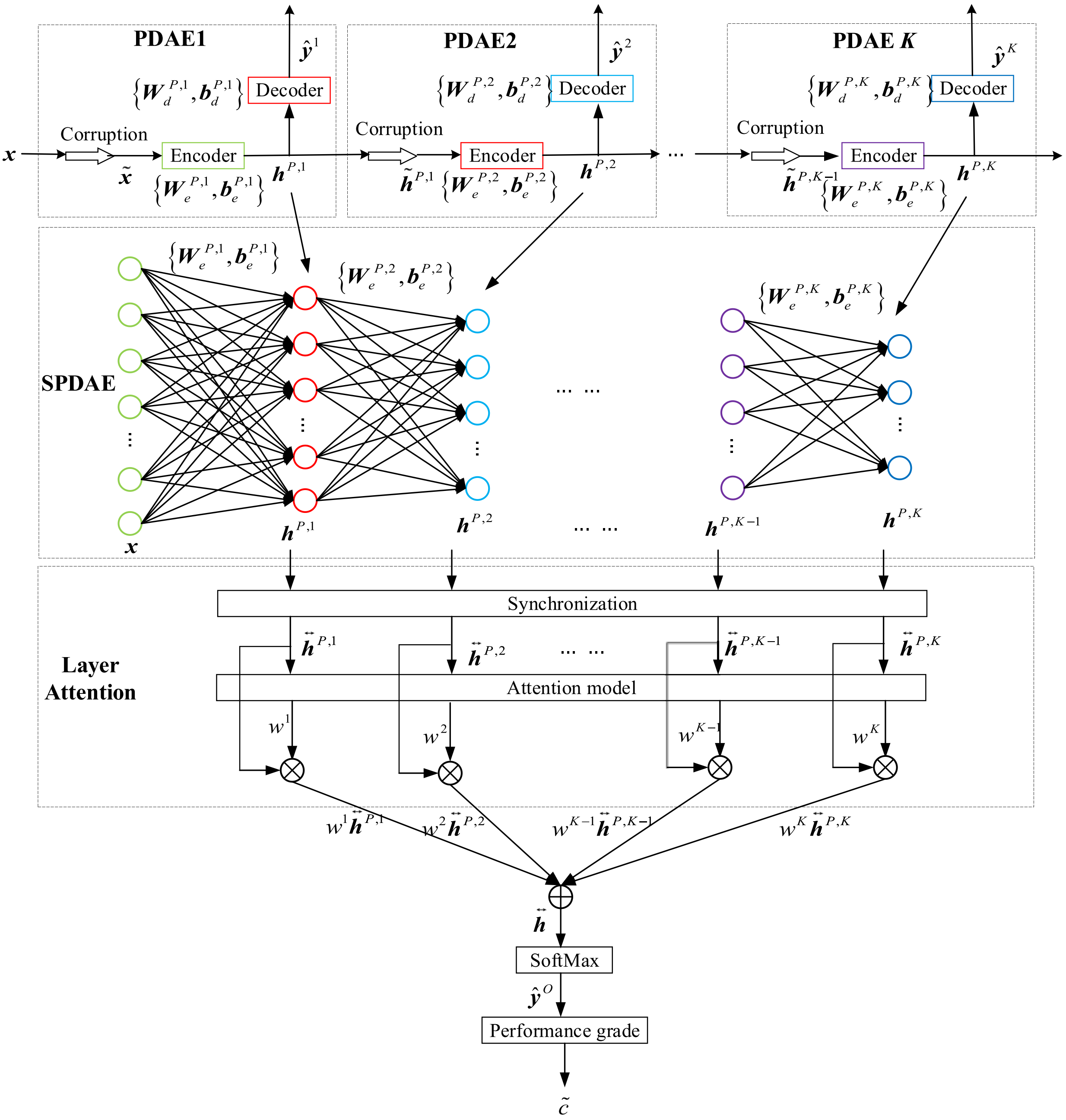

PDAE1
PDAE2
PDAE K
$\hat{y}^1$
$\hat{y}^2$
$\hat{y}^K$
$\{W_d^{P,1}, b_d^{P,1}\}$
$\{W_d^{P,2}, b_d^{P,2}\}$
$\{W_d^{P,K}, b_d^{P,K}\}$
Decoder
Corruption
Encoder
x
$\tilde{x}$
$\{W_e^{P,1}, b_e^{P,1}\}$
$h^{P,1}$
$\tilde{h}^{P,1}$
$\{W_e^{P,2}, b_e^{P,2}\}$
$h^{P,2}$
$\tilde{h}^{P,K-1}$
$\{W_e^{P,K}, b_e^{P,K}\}$
$h^{P,K}$
SPDAE
$h^{P,K-1}$
Synchronization
Layer Attention
$\overleftrightarrow{h}^{P,1}$
$\overleftrightarrow{h}^{P,2}$
$\overleftrightarrow{h}^{P,K-1}$
$\overleftrightarrow{h}^{P,K}$
Attention model
w^1
w^2
w^{K-1}
w^K
$w^1\overleftrightarrow{h}^{P,1}$
$w^2\overleftrightarrow{h}^{P,2}$
$w^{K-1}\overleftrightarrow{h}^{P,K-1}$
$w^K\overleftrightarrow{h}^{P,K}$
$\overleftrightarrow{h}$
SoftMax
$\hat{y}^O$
Performance grade
$\tilde{c}$

Figure 3. Structure of LA-SPDAE.

$\theta_{\text{PDAE}}^2 = \{\boldsymbol{W}_e^{P,2}, b_e^{P,2}; \boldsymbol{W}_d^{P,2}, b_d^{P,2}\}$ and the second-layer performance-relevant features $\boldsymbol{H}_P^2 = [\boldsymbol{h}_1^{P,2}, \boldsymbol{h}_2^{P,2}, ..., \boldsymbol{h}_N^{P,2}]^T$.

Assume that all the first k PDAEs have been pretrained already, and the k-layer performance-relevant features are denoted as $\boldsymbol{H}_P^k = [\boldsymbol{h}_1^{P,k}, \boldsymbol{h}_2^{P,k}, ..., \boldsymbol{h}_N^{P,k}]^T$, which will be further adopted to construct the k + 1th PDAE. The relationships among the input, hidden, and output data are as follows

$$\boldsymbol{h}_n^{P,k+1} = f(\boldsymbol{W}_e^{P,k+1}\tilde{\boldsymbol{h}}_n^{P,k} + \boldsymbol{b}_e^{P,k+1}),\quad k = 1, 2, ..., K-1 \tag{12}$$

$$\hat{\boldsymbol{y}}_n^{k+1} = g(\boldsymbol{W}_d^{P,k+1}\boldsymbol{h}_n^{P,k+1} + \boldsymbol{b}_d^{P,k+1}) \tag{13}$$

where $\tilde{\boldsymbol{h}}_n^{P,k}$ is the corruption of $\boldsymbol{h}_n^{P,k}$; $\theta_{\text{PDAE}}^{k+1} = \{\boldsymbol{W}_e^{P,k+1}, b_e^{P,k+1}; \boldsymbol{W}_d^{P,k+1}, b_d^{P,k+1}\}$ is the parameter set of the k + 1th PDAE, and $\hat{\boldsymbol{Y}}^{k+1} = [\hat{\boldsymbol{y}}_1^{k+1}, \hat{\boldsymbol{y}}_2^{k+1}, ..., \hat{\boldsymbol{y}}_N^{k+1}]^T$ are the posterior probabilities estimated by the k + 1th PDAE. Thereafter, it is pretrained through minimizing the cross-entropy loss function of the k + 1th PDAE as follows

$$L_{\text{PDAE}}^{k+1}(\theta_{\text{PDAE}}^{k+1}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_n^c \log(\hat{y}_n^{k+1,c}) \tag{14}$$

where $\hat{y}_n^{k+1,c}$ is the cth element of $\hat{\boldsymbol{y}}_n^{k+1}$.

After K PDAEs have been pretrained, $\theta_{\text{SPDAE}} = \{\boldsymbol{W}_e^{P,1}, b_e^{P,1}; \boldsymbol{W}_e^{P,2}, b_e^{P,2}; ...; \boldsymbol{W}_e^{P,K}, b_e^{P,K}\}$ is used as initial parameters to construct the SPDAE network. What needs to be specified is that, in the layer-wise pretraining, the input corruption of each individual layer is only applied in initial denoising-training for learning useful feature extractors. Once parameters are obtained, they will be used on uncorrupted inputs. In addition, the fundamental task of SPDAE is not to denoise per se. On the contrary, denoising is advocated as a criterion to learn more informative features.

3.3. LA-SPDAE for Operating Performance Assessment.

In each layer of the SPDAE, performance-relevant features represent a certain level of the characteristics of the input data associated with the process operating performance. Here, the layer attention mechanism is introduced to the SPDAE, termed LA-SPDAE, to determine the importance of the performance-relevant features of each layer adaptively and integrate them reasonably. The structure of the LA-SPDAE is shown in Figure 3.

Since the dimensions of $\boldsymbol{h}_n^{P,1}$, $\boldsymbol{h}_n^{P,2}$,..., $\boldsymbol{h}_n^{P,K}$ are usually unequal according to the structure characteristic of the SPDAE, these features need to be synchronized before integrating. Refer to the performance-relevant feature with the largest dimension, the vacancies of other features are padded with zeros from both ends. The synchronized performance-relevant features are recorded as $\overleftrightarrow{\boldsymbol{h}}_n^{P,1}$, $\overleftrightarrow{\boldsymbol{h}}_n^{P,2}$, ..., $\overleftrightarrow{\boldsymbol{h}}_n^{P,K} \in R^{J_{\overleftrightarrow{h}} \times 1}$ correspondingly, and $J_{\overleftrightarrow{h}}$ is the maximum dimension of the performance-relevant features. For example, if the dimensions of $\boldsymbol{h}_n^{P,1}$ and $\boldsymbol{h}_n^{P,2}$ are eight and five, the synchronized versions are $\overleftrightarrow{\boldsymbol{h}}_n^{P,1} = \boldsymbol{h}_n^{P,1}$ and $\overleftrightarrow{\boldsymbol{h}}_n^{P,2} = [0, \boldsymbol{h}_n^{P,2\mathrm{T}}, 0, 0]^T$.

Attention mechanism allows the model to capture the most important performance-relevant features for operating performance assessment. It can assign an adaptive weight rather than a fixed one to each layer performance-relevant feature. For input $\boldsymbol{x}_n$, the layer attention mechanism assigns attention weight w_n^k to the k-layer performance-relevant feature according to its potential relationship with the process operating performance. The structure of the layer attention model is as follows

$$e_n^k = \boldsymbol{v}_A \cdot \tanh(\boldsymbol{W}_A \overleftrightarrow{\boldsymbol{h}}_n^{P,k} + \boldsymbol{b}_A) \tag{15}$$

$$w_n^k = \frac{|e_n^k|}{\sum_{j=1}^{K} |e_n^j|}, \qquad k = 1, 2, ..., K, \qquad n = 1, 2, ..., N \tag{16}$$

where $\boldsymbol{v}_A \in R^{1 \times J_{\overleftrightarrow{h}}}$, $\boldsymbol{W}_A \in R^{J_{\overleftrightarrow{h}} \times J_{\overleftrightarrow{h}}}$ and $\boldsymbol{b}_A \in R^{J_{\overleftrightarrow{h}} \times 1}$ are the parameters to be learned of the layer attention model and e_n^k is the corresponding attention value of $\overleftrightarrow{\boldsymbol{h}}_n^{P,k}$. The normalized value w_n^k calculated by eq 16 is called layer attention weight of the k-layer performance-relevant feature $\overleftrightarrow{\boldsymbol{h}}_n^{P,k}$. All parameters in eq 15 form the parameter set $\theta_A = \{\boldsymbol{v}_A, \boldsymbol{W}_A, \boldsymbol{b}_A\}$, which can be learnt through the BP algorithm in training the whole LA-SPDAE network. Then, all performance-relevant features of $\boldsymbol{x}_n$ are integrated by the layer attention weights as follows

$$\overleftrightarrow{\boldsymbol{h}}_n = \sum_{k=1}^{K} w_n^k \overleftrightarrow{\boldsymbol{h}}_n^{P,k}, \qquad n = 1, 2, ..., N \tag{17}$$

After that, a SoftMax function is added to the top layer to set up the LA-SPDAE-based operating performance assessment model. It is shown as follows

$$g(\overleftrightarrow{\boldsymbol{h}}_n) = \begin{bmatrix} \hat{y}_n^{O,1} \\ \hat{y}_n^{O,2} \\ \vdots \\ \hat{y}_n^{O,C} \end{bmatrix} = \frac{1}{\sum_{c=1}^{C} \exp(\theta_{O,c}^T \overleftrightarrow{\boldsymbol{h}}_n)} \begin{bmatrix} \exp(\theta_{O,1}^T \overleftrightarrow{\boldsymbol{h}}_n) \\ \exp(\theta_{O,2}^T \overleftrightarrow{\boldsymbol{h}}_n) \\ \vdots \\ \exp(\theta_{O,C}^T \overleftrightarrow{\boldsymbol{h}}_n) \end{bmatrix} \tag{18}$$

where $\hat{y}_n^{O,c} = \exp(\theta_{O,c}^T \overleftrightarrow{\boldsymbol{h}}_n) / \sum_{c=1}^{C} \exp(\theta_{O,c}^T \overleftrightarrow{\boldsymbol{h}}_n)$ is the posterior probability of $\overleftrightarrow{\boldsymbol{h}}_n$ belonging to the c th performance grade, and $\sum_{c=1}^{C} \hat{y}_n^{O,c} = 1$; $\theta_O = \{\theta_{O,1}^T, \theta_{O,2}^T, ..., , \theta_{O,C}^T\}$ is the parameter set connecting the top layer and the output layer.

For training the LA-SPDAE network, the pretrained parameter set θ_{SPDAE} is used as the initial parameters. In addition, the parameter sets θ_O and θ_A are initialized randomly. Then, a BP-based global fine-tuning is put into effect to train these parameters simultaneously with the following loss function

$$L_{\mathrm{LA-SPDAE}}(\theta_{\mathrm{LA-SPDAE}}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_n^c \log(\hat{y}_n^{O,c}) \tag{19}$$

To evaluate the operating performance of a new sample $\boldsymbol{x}_{\mathrm{new}}$ using LA-SPDAE, the forward propagation is performed as follows

$$\boldsymbol{h}_{\mathrm{new}}^{P,1} = f(\boldsymbol{W}_e^{P,1} \boldsymbol{x}_{\mathrm{new}} + \boldsymbol{b}_e^{P,1}) \tag{20}$$

$$\boldsymbol{h}_{\mathrm{new}}^{P,k+1} = f(\boldsymbol{W}_d^{P,k} \boldsymbol{h}_{\mathrm{new}}^{P,k} + \boldsymbol{b}_d^{P,k}), \qquad k = 1, 2, .., K - 1 \tag{21}$$

$$e_{\mathrm{new}}^k = \boldsymbol{v}_A \cdot \tanh(\boldsymbol{W}_A \overleftrightarrow{\boldsymbol{h}}_{\mathrm{new}}^{P,k} + \boldsymbol{b}_A), \qquad k = 1, 2, ..., K \tag{22}$$

$$w_{\mathrm{new}}^k = \frac{|e_{\mathrm{new}}^k|}{\sum_{j=1}^{K} |e_{\mathrm{new}}^j|} \tag{23}$$

$$\overleftrightarrow{\boldsymbol{h}}_{\mathrm{new}} = \sum_{k=1}^{K} w_{\mathrm{new}}^k \overleftrightarrow{\boldsymbol{h}}_{\mathrm{new}}^{P,k} \tag{24}$$

$$\hat{y}_{\mathrm{new}}^{O,c} = \exp(\theta_{O,c}^T \overleftrightarrow{\boldsymbol{h}}_{\mathrm{new}}) / \sum_{c=1}^{C} \exp(\theta_{O,c}^T \overleftrightarrow{\boldsymbol{h}}_{\mathrm{new}}), \quad c = 1, 2, ..., C \tag{25}$$

where $\boldsymbol{h}_{\mathrm{new}}^{P,k}$ and $\overleftrightarrow{\boldsymbol{h}}_{\mathrm{new}}^{P,k}$ are k-layer performance-relevant feature and the synchronized one; w_{new}^k is the layer attention weight of $\overleftrightarrow{\boldsymbol{h}}_{\mathrm{new}}^{P,k}$; $\hat{y}_{\mathrm{new}}^{O,c}$ is the posterior probability of $\boldsymbol{x}_{\mathrm{new}}$ belonging to the cth performance grade, and that with maximum posterior probability is determined as the final performance grade for $\boldsymbol{x}_{\mathrm{new}}$, i.e.,

$$\tilde{c} = \arg \max_{1 \le c \le C} \{\hat{y}_{\mathrm{new}}^{O,c}\} \tag{26}$$

where $\tilde{c}$ is the performance grade of sample $\boldsymbol{x}_{\mathrm{new}}$ belonging to.

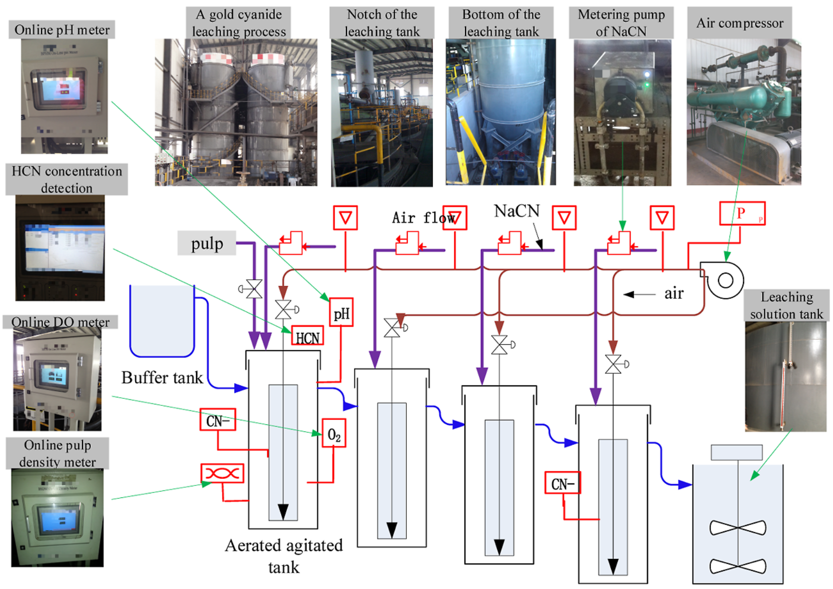


Figure 4. Flowchart of the gold cyanide leaching process.

Algorithm 1: LA-SPDAE-based operating performance assessment

```
Input: Training set $\{\boldsymbol{X},\boldsymbol{Y}\}$; Hidden layers $K$; Hidden neurons $Q$;
        Pretraining: learning rate $\eta_1,\eta_2,\eta_3$, batch size $B_1$, epochs $E_1$;
        Fine-tuning: learning rate $\eta_4$, batch size $B_2$, epochs $E_2$.
Output: Performance grades $\tilde{c}$.
Offline modeling phase
Standardized training set $\{\boldsymbol{X},\boldsymbol{Y}\}$
Pretraining:
Randomly initialization of network parameters
$\boldsymbol{h}^{P,0} \leftarrow \boldsymbol{x}$;
for $k \leftarrow 1$ to $K$ do
  for $n \leftarrow 1$ to $N$ do
      $\tilde{\boldsymbol{h}}_n^{P,k-1} \leftarrow$ data corruption $(\boldsymbol{h}_n^{P,k-1})$;
      Encoder: $\boldsymbol{h}_n^{P,k} \leftarrow f(\boldsymbol{W}_e^{P,k}\tilde{\boldsymbol{h}}_n^{P,k-1}+\boldsymbol{b}_e^{P,k})$;
      Decoder: $\hat{\boldsymbol{y}}_n^{k} \leftarrow g(\boldsymbol{W}_d^{P,k}\boldsymbol{h}_n^{P,k}+\boldsymbol{b}_d^{P,k})$;
  end
end
Fine-tuning:
Initialize each layer with $\theta_{SPDAE}$, and random initialize $\theta_O$ and $\theta_A$;
for $n \leftarrow 1$ to $N$ do
    $\boldsymbol{h}_n^{P,0} \leftarrow \boldsymbol{x}_n$;
  for $k \leftarrow 1$ to $K$ do
      $\boldsymbol{h}_n^{k} \leftarrow f(\boldsymbol{W}_e^{k}\boldsymbol{h}_n^{k-1}+\boldsymbol{b}_e^{k})$;
      $\overleftrightarrow{\boldsymbol{h}}_n^{k} \leftarrow$ synchronize $(\boldsymbol{h}_n^{k})$;
      Calculate attention value: $e_n^k \leftarrow \boldsymbol{v}_A \tanh(\boldsymbol{W}_A\overleftrightarrow{\boldsymbol{h}}_n^{k}+\boldsymbol{b}_A)$;
  end
  Attention normalization: $w_n^k \leftarrow |e_n^k| \big/ \sum_{j=1}^{K}|e_n^j|, k=1,2,...,K$;
  Feature fusion: $\overleftrightarrow{\boldsymbol{h}}_n \leftarrow \sum_{k=1}^{K} w_n^k \overleftrightarrow{\boldsymbol{h}}_n^{k}$;
  Posterior probability: $\hat{y}_n^{O,c} \leftarrow \exp(\theta_{O,c}^T\overleftrightarrow{\boldsymbol{h}}_n) \Big/ \sum_{c=1}^{C}\exp(\theta_{O,c}^T\overleftrightarrow{\boldsymbol{h}}_n), c=1,2,...,C$;
end
Online assessment phase
Input the new sample $\boldsymbol{x}_{new}$ to the LA-SPDAE network;
Obtain the assessment result: $\tilde{c} \leftarrow \arg\max_{1\le c\le C}\{\hat{y}_{new}^{O,c}\}$;
```

4. CASE STUDY

In this study, the proposed LA-SPDAE-based assessment model will be tested on a gold cyanide leaching process in Shandong, China. The simulation configurations are as follows: operating system: 64-bit Microsoft Windows 10; CPU: Intel i5-11300H (3.10 GHz); RAM: 16GB; Software: Python 3.8.

4.1. Process Description.

Cyanide leaching is an important process in hydrometallurgy. Figure 4 is the flowchart of the gold cyanide leaching process, consisting of four cascaded pneumatic mixing tanks, a buffer tank, a leaching solution tank, and an air compressor. Cyanide and oxygen are important reactants in the cyanide leaching. Sodium cyanide (NaCN) is a commonly used reagent, and the oxygen provides the oxidation-reduction potential to promote the reaction. Although the gold is inert to oxidation, it is widely accepted that pure gold cyanide is an electrochemical process, and the oxidation of gold is a prerequisite for its dissolution in the alkaline cyanide lixiviant. Then, the cyanogen gold complexes, $[Au(CN)_2]^-$, are synthesized in the pulp. To avoid the generation of hydrogen cyanide (HCN), pH is controlled at about 12 by adding calcium hydroxide ($Ca(OH)_2$) to the pulp.

Leaching rate is the crucial index for evaluating the operating performance of the cyanide leaching process. It is generally believed that the higher the leaching rate is, the better the process operating performance is. However, the leaching rate cannot be measured online and can only be analyzed offline with a certain time delay. The process variables of the gold cyanide leaching process are shown in Table 1. Known from expert experience and process mechanism analysis, the concentration of HCN is mainly used to ensure production safety and has little effect on the leaching rate. Moreover, in actual production, the airflow is usually large enough to make the dissolved oxygen in a saturated state; thus, both the airflow and the concentration of

Table 1. Process Variables of the Cyanide Leaching Process

no.	variable description	units
1	NaCN flow in the first pneumatic mixing tank	m^3/s
2	NaCN flow in the second pneumatic mixing tank	m^3/s
3	NaCN flow in the fourth pneumatic mixing tank	m^3/s
4	concentration of the pulp	wt %
5	air flow in the first pneumatic mixing tank	m^3/s
6	air flow in the second pneumatic mixing tank	m^3/s
7	air flow in the third pneumatic mixing tank	m^3/s
8	air flow in the fourth pneumatic mixing tank	m^3/s
9	concentration of dissolved oxygen	mg/L
10	CN^-concentration in the first pneumatic mixing tank	mol/L
11	CN^-concentration in the fourth pneumatic mixing tank	mol/L
12	concentration of HCN	mg/m^3

https://doi.org/10.1021/acsomega.3c00414

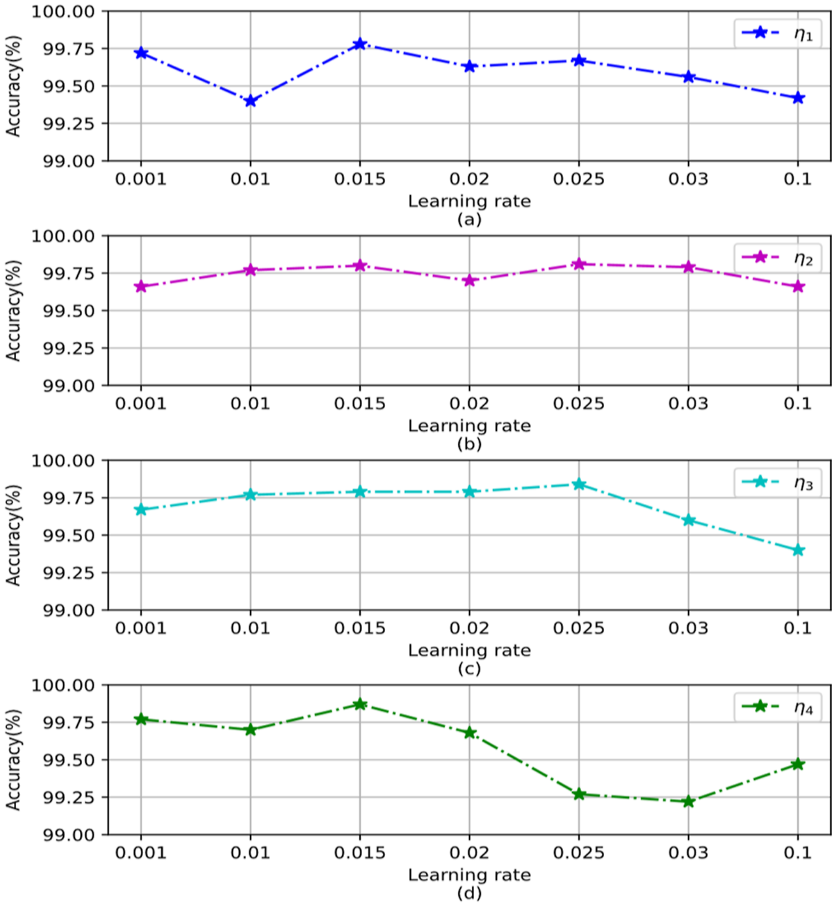


Figure 5. Assessment accuracy of the validation data set under different learning rates (a) η_1; (b) η_2; (c) η_3; and (d) η_4.

the dissolved oxygen have little effect on the leaching rate in the circumstances. On account of the above reasons, variables 5−9 and 12 are used to simulate the interferences of performance-irrelevant features in this study. Beyond that, all the other process variables affect the leaching rate directly or indirectly and have strong or weak correlations with the process operating performance.

4.2. Analysis and Discussion. Normally, original data collected from the cyanide leaching process are difficult to use directly because a variety of problems can be encountered, including data missing, outliers, invariant values, and so on. What needs to be stated here is that the data used in this case study have been preprocessed by members of our research group, that is, the data do not encounter the problems mentioned above and can be regarded as clean samples. According to the leaching rates from low to high and the expert experience, the operating performance of the cyanide leaching process is divided into four performance grades, i.e., poor, general, fine, and optimal, which are numbered as 1, 2, 3, and 4 successively. 10,000 samples were selected from the historical production data set, and each performance grade accounts for a quarter. 80% of samples are selected randomly and used for training, and the remaining 20% is used for testing. In the test data set, each performance grade contains 500 samples. To demonstrate the performance of the LA-SPDAE model, the samples will be corrupted according to different proportions ν of zero-masking noise, whether for offline modeling or online test. For simplicity, LA-SPDAE (ν) is used to represent the LA-SPDAE established based on corrupted inputs under proportion ν.

In the construction of a DNN, the critical point is to determine network parameters and hyperparameters. Nevertheless, no universal method is available to decide on them theoretically. Here, the commonly used grid search[34] and cross-validation[35,36] are combined to choose the hyperparameters of the proposed network. First, the training data set is further equally divided into five parts, and 5-fold cross-validation is adopted to determine the hyperparameters based on the average of the assessment accuracy on validation data sets. The hyperparameters involve the numbers of nodes in three layers, pretraining learning rates of each layer, fine-tuning learning rate, number of layers, batch sizes, and epochs during pretraining and fine-tuning procedures. The learning rates are explored under a temporary network structure, which has three hidden layers and each has 8, 5, and 3 nodes from the shallow layer to the deep one. The candidate set of the pretraining learning rate η_1, η_2, η_3 and fine-tuning learning rate η_4 is {0.001, 0.01, 0.015, 0.02, 0.025, 0.03, 0.1}. According to the principle of the LA-SPDAE, the pretraining procedure is carried out layer by layer. The performance-relevant feature of the previous layer is used as the input of the current layer. Figure 5 shows the assessment accuracy of each SPDAE under different pretraining learning rates and the final assessment accuracy in the fine-tuning procedure. Since a large learning rate can make the network converge quickly, large learning rates are selected to train each layer and fine-tune the whole network without causing shock.

Then, the optimal learning rates are set as $\eta_1 = 0.015$, $\eta_2 = 0.025$, $\eta_3 = 0.025$, and $\eta_4 = 0.015$.

The second hyperparameter to be explored is the number of hidden layers, and its influence on the assessment accuracy and training time is shown in Figure 6. It can be seen from Figure 6 that, with the increasing number of hidden layers, the assessment accuracy and the training time increase continuously. However, the performance does not improve any longer when the number of hidden layers increases to five. The LA-SPDAE model with three layers shows better performance and uses shorter training time, so it can be considered as the optimal model for the training data set.

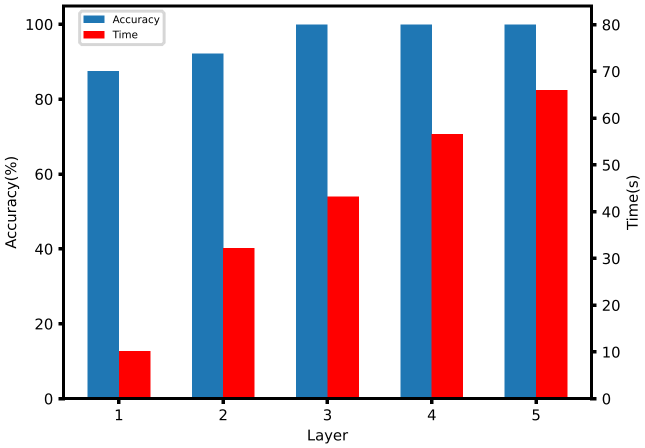


Figure 6. Layer number versus assessment accuracy and training time.

Additionally, to avoid overfitting and ensure the generalization ability of the network, the criterion for choosing the number of layer nodes is as compact as possible while ensuring accuracy, and the final numbers of nodes in layers 1, 2, and 3 are eight, five, and three. Small batch size will reduce the convergence speed; large batch size will not only consume too much memory but also adversely affect the optimization process and generalization performance of the network. Small epoch will lead to under-fitting of training, resulting in poor generalization ability; large epoch easily leads to over-fitting of training. Therefore, the batch sizes and epochs used in pretraining and fine-tuning procedures are chosen by comparing the training curves, i.e., training and validation errors versus amounts of training time. Then, appropriate batch sizes and epochs are selected according to experience. The hyperparameters of the LA-SPDAE model are given in Table 2.

Using the hyperparameters listed in Table 2, the losses and accuracy of the LA-SPDAE model on the training and validation data sets are shown in Figure 7, where the curves about the verification data set are drawn based on the result of one of the five verification data sets. Seen from Figure 7, LA-SPDAE performs well on both the training and verification data sets, indicating that the determined parameters and hyperparameters are appropriate.

In this case, the support vector machine (SVM),[37] total projection to latent structures (TPLS),[7] SAE, SDAE, SPDAE, and two DNNs named DNN1 and DNN2 are constructed for comparison. The SVM and TPLS are shallow models, and others are DNNs. Both DNN1 and DNN2 are three-layer fully connected neural networks, but DNN2 has one more corruption layer than DNN1 before the input layer. Except that there are no fusion layers in those DNNs, their structures are the same as that of the LA-SPDAE. It is noteworthy that although the structures of DNN2 and SPDAE look the same, their training methods are completely different, resulting in different network models. For the DNN2, a whole forward propagation and a whole fine-tuning are usually regarded as completing a round of training, which will be repeated until satisfactory accuracy is obtained. SPDAE adopts a greedy layer-wise training algorithm as described in Section 3.2. It can find good initial weights for reducing the likelihood of the network becoming trapped in a local minimum. Because there is a corruption layer in each PDAE pretraining, the weights of SPDAE are different from those of DNN2.

Table 2. Hyperparameters of LA-SPDAE

hyperparameters	value
number of hidden layers	3
number of input layer nodes	12
number of layer 1 nodes	8
number of layer 2 nodes	5
number of layer 3 nodes	3
number of fusion nodes	8
number of output layer nodes	4
pretraining learning rate of layer 1, η_1	0.015
pretraining learning rate of layer 2, η_2	0.025
pretraining learning rate of layer 3, η_3	0.025
pretraining batch size	16
pretraining epochs	200
fine-tuning learning rate, η_4	0.015
fine-tuning batch size	16
fine-tuning epochs	100

Table 3 gives the total training times of different assessment models. Seen from Table 3, LA-SPDAE spends more training time than TPLS, SVM, SAE, DNN1, and DNN2, but it still takes less training time than SPDAE and SDAE. That is to say, although the structure of LA-SPDAE is more complex and more parameters need to be trained than those of SPDAE and SDAE, LA-SPDAE still convergences faster than SPDAE and SDAE for its superior ability in feature extraction.

Figure 8 shows the visual results of performance-relevant features extracted from every layer of the LA-SPDAE (20%) through t-distributed stochastic neighbor embedding (*t*-SNE).[38] It can be seen that the performance-relevant features of different grades in the first and second layers are not separated entirely. With increasing feature extraction depth, the separability of performance-relevant features of different grades is significantly enhanced, and the fused features of different layers have the best separability. That is to say, with increasing number of layers, LA-SPDAE captures more informative features from input data and provides a better description of different performance grades based on the fused features.

Figure 9 exhibits the detailed online assessment results based on different assessment models. The accuracy of TPLS, SVM, SAE, DNN1, DNN2(20%), SDAE (20%), and SPDAE (20%) are 42.55, 61, 91.55, 90.7, 99.15, 98.55, and 98.7%, respectively. The proposed LA-SPDAE (20%) acquires the highest value 99.85%.

In addition, the precision ratio *p*, recall ratio *r*, and F1 score are also used to illustrate the performances of different assessment models. As can be seen from Table 4, LA-SPDAE achieves the highest averages of the precision ratio *p*, recall ratio *r*, and F1 score compared with other models, which can be attributed to the following four aspects. The first aspect is the model structure. All DNNs perform better than SVM and TPLS.

https://doi.org/10.1021/acsomega.3c00414
ACS Omega 2023, 8, 14583−14596

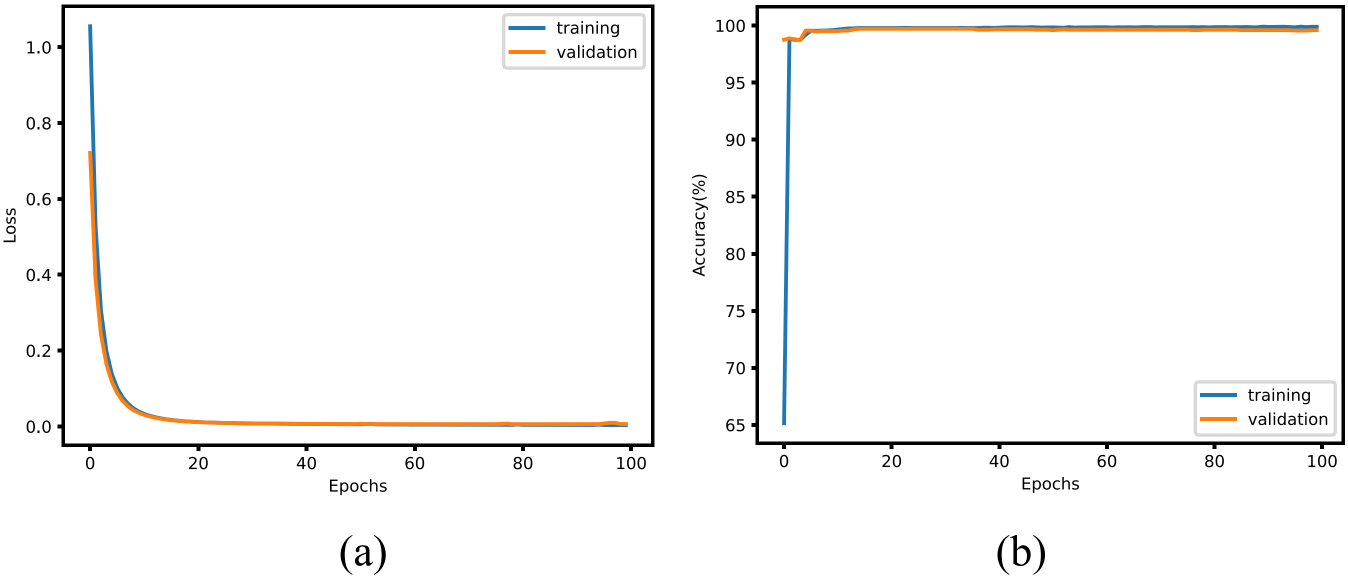


Figure 7. Losses and accuracy in the fine-tuning procedure for the LA-SPDAE; (a) losses; (b) accuracy.

Table 3. Training Times of Different Assessment Models (ν = 20%)

models	LA-SPDAE	SPDAE	SDAE	SAE	DNN1	DNN2	SVM	TPLS
time (s)	43.27	52.26	54.59	31.77	11.72	12.02	0.4	0.31

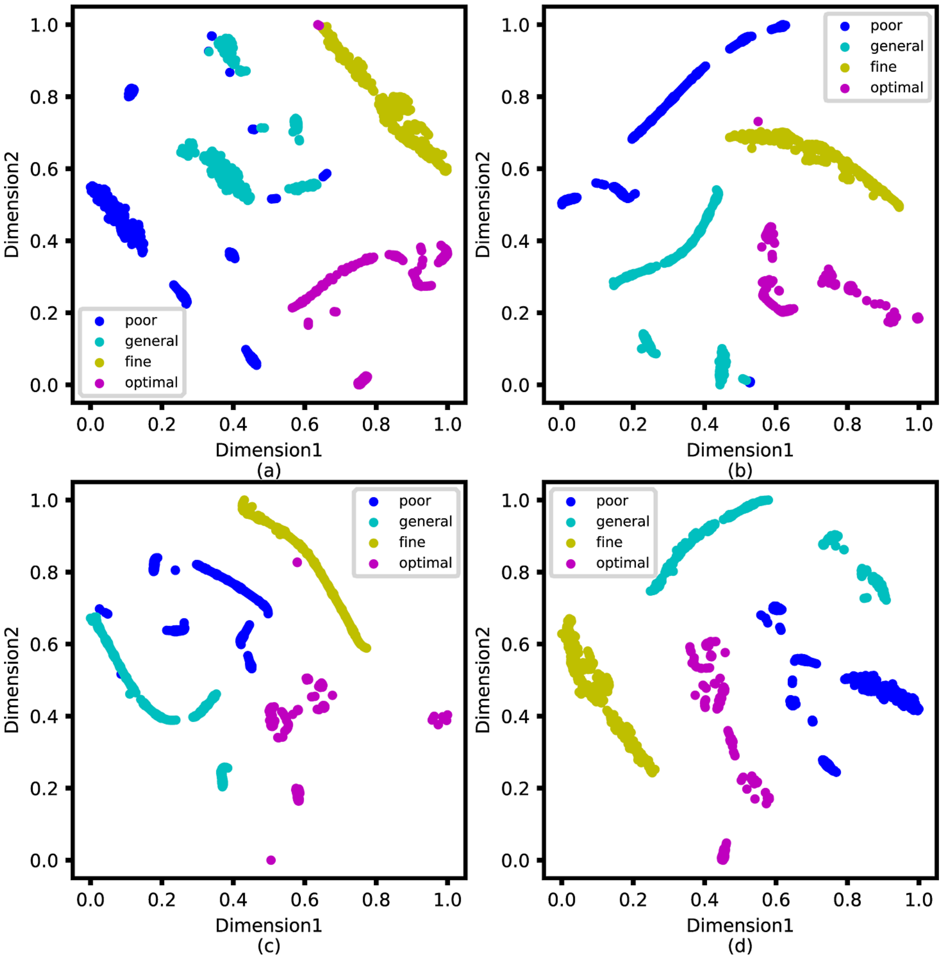


Figure 8. *t*-SNE embedding of performance-relevant features from different layers of LA-SPDAE (20%); (a) layer 1; (b) layer 2; (c) layer 3; and (d) fusion layer.

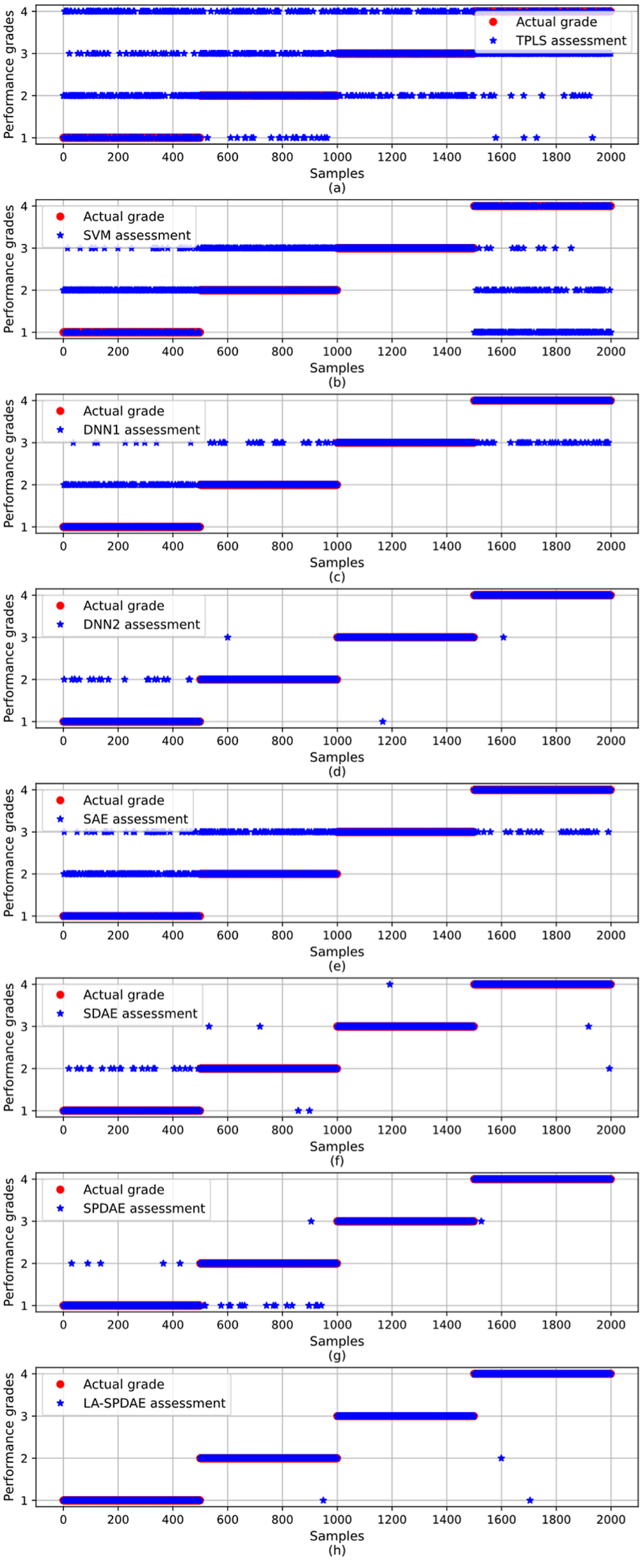


Figure 9. Detailed online assessment results based on different assessment models: (a) TPLS; (b) SVM; (c) DNN1; (d) DNN2 (20%); (e) SAE; (f) SDAE (20%); (g) SPDAE (20%); and (h) LA-SPDAE (20%).

It suggests that, through multilayer nonlinear mapping, the models with deep structures show stronger abilities in learning and extracting essential features from the raw input data, but SVM and TPLS lack such capability for nonlinear data disturbed by noise. The second is the utilization of the "corruption" link. While denoising, the input corruption of each layer can help train a useful feature extractor to obtain better higher-level and invariant feature representations, making SDAE, SPDAE, DNN2, and LA-SPDAE get better assessment accuracy than SAE and DNN1. The third one is to enhance the learning of performance-relevant features. By optimizing the cross-entropy loss of performance grade labels in the layer-wise pretraining procedure, each PDAE is supervised to extract performance-relevant features, which is conducive to distinguishing different performance grades more accurately, so the DNN2, SPDAE, and LA-SPDAE perform better than the SDAE. The last aspect is the adaptive integration of performance-relevant features of all layers. By introducing the attention mechanism, the LA-SPDAE can achieve more accurate assessment results than the DNN2 and SPDAE. It indicates that performance-relevant features of each layer contain useful information for the operating performance assessment, and taking full advantage of them helps to improve the discrimination of the performance grades. To sum up, the LA-SPDAE shows the most outstanding performance of all compared models in this case study.

To investigate the immediacy of different models in online assessment, the assessment time of each model is given in Table 5. Because the structure of the proposed LA-SPDAE is more complex, its online assessment takes more time than for other compared models. However, assessment time of one sample less than 0.2 millisecond (ms) can fully cope with the high sampling frequency in actual production.

In order to further explore the importance of the performance-relevant features of each layer in integrated utilization, the layer attention weights corresponding to different layers on the test data set are examined and presented in Figure 10. It can be seen that, for different performance grades, the distributions of layer attention weights are different. Specifically, the performance-relevant features of layer 1 and layer 2 are the main components that make up the fused features of the poor grade, while the fused features of the general grade are the integration of performance-relevant features from layer 2 and layer 3. For the fine and optimal grades, the fused features are mainly composed of single performance-relevant feature from layer 2 and layer 3, respectively. In other words, in operating performance assessment with multiple performance grades, each performance grade should not be characterized only by the features of a certain layer. The effective utilization of the performance-relevant features of each layer can provide more accurate online assessment results.

Finally, the influence of corrupted proportion ν is investigated and shown in Figure 11. The LA-SPDAE performs better than all other models for a relatively wide range of corrupted proportions. As the corrupted proportion increases, the accuracy of LA-SPDAE does not decrease significantly. What this comes down to is that, by introducing the performance grade label, LA-SPDAE can extract the unique characteristics of each performance grade more accurately, and it also makes full use of the information of each layer under the support of the layer attention mechanism. These techniques make the LA-SPDAE more capable of extracting unique invariant features from changing corrupted inputs, making LA-SPDAE more robust to noise data.

5. CONCLUSIONS

In this paper, a novel operating performance assessment strategy based on LA-SPDAE is proposed for industrial data with weak

Table 4. Precision Ratio p, Recall Ratio r, and F1 Score of Different Assessment Methods

index	methods	grade 1	grade 2	grade 3	grade 4	average
p(%)	TPLS	84.09	48.79	37.91	27.02	49.45
	SVM	49.14	47.04	69.06	**100**	66.31
	DNN1	**100**	78.88	88.97	**100**	91.96
	DNN2	**100**	**99.8**	96.9	**100**	99.17
	SAE	**100**	76.45	83.33	**100**	89.95
	SDAE	99.58	95.57	99.4	99.8	98.59
	SPDAE	96.3	98.97	99.6	**100**	98.72
	LA-SPDAE	99.6	**99.8**	**100**	**100**	**99.85**
r(%)	TPLS	29.6	52.6	64.6	23.4	42.55
	SVM	40	62	**100**	42	61
	DNN1	74.4	92.6	**100**	95.8	90.7
	DNN2	96.8	**100**	**100**	99.8	99.15
	SAE	69.2	**100**	**100**	80	87.3
	SDAE	95.6	99.2	99.8	99.6	98.55
	SPDAE	99	96	**100**	**99.8**	98.7
	LA-SPDAE	**100**	99.8	**100**	99.6	**99.85**
F1 score (%)	TPLS	43.79	50.63	47.78	25.08	41.82
	SVM	44.1	53.49	81.7	59.15	59.61
	DNN1	85.32	85.19	94.16	97.85	90.63
	DNN2	98.37	**99.9**	98.43	**99.9**	99.15
	SAE	81.8	86.66	90.91	88.89	87.07
	SDAE	97.55	97.35	99.6	99.7	98.55
	SPDAE	97.63	97.46	99.8	**99.9**	98.69
	LA-SPDAE	**99.8**	99.8	**100**	99.8	**99.85**

Table 5. Online Assessment Times of Different Models

models	LA-SPDAE	SPDAE	SDAE	SAE	DNN1	DNN2	SVM	TPLS
time (ms)	0.12	0.069	0.064	0.062	0.064	0.063	0.031	0.1

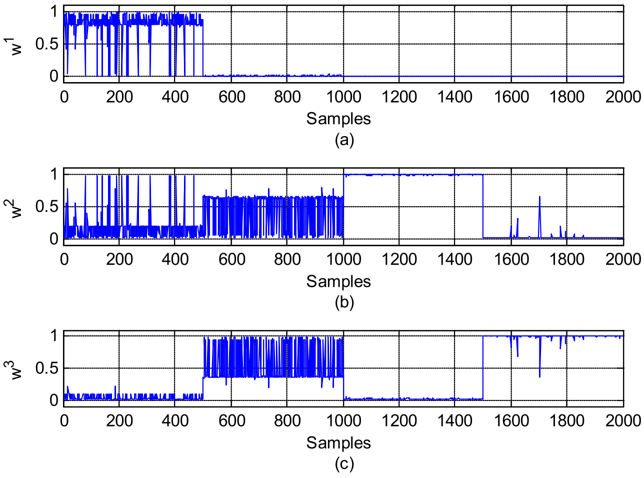


Figure 10. Attention weights of performance-relevant features; (a) layer 1; (b) layer 2; and (c) layer 3.

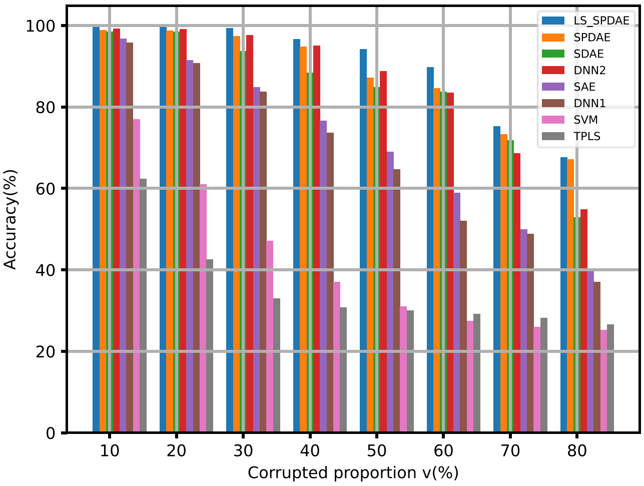


Figure 11. Accuracy versus corrupted proportion ν.

differences and strong noise interferences. The single DAE is first improved to PDAE, so that it can train a more robust feature extractor while extracting the performance-relevant feature from the training data disturbed by strong noise. By stacking multiple PDAEs hierarchically, the performance-relevant features are successively learned, and these from different layers are further fused based on layer attention weights, which enables LA-SPDAE to make full use of information at all layers and improves the discrimination ability of the assessment model for data with weak differences. Finally, through the case study in cyanide leaching, the proposed LA-SPDAE model shows advantages in a variety of evaluation indicators compared with some conventional DNNs and shallow models.

Actual industrial processes usually have the characteristics of long process and large lag. It usually takes several hours, days, or even longer for raw materials from input to output. If the assessment of product quality and operating performance can only be carried out after the production, it is a post-assessment. Although the assessment results have certain guiding significance for future productions, it has no effect on the economic losses caused by previous unsatisfactory operation. This study proposed a method of online assessment for the process operating performance based on real-time collected production data, which can help managers and operators to timely grasp the operating performance and production index that will be obtained under current operation state. Compared with post-assessment, this is pre-assessment. It can find out the poor operating performance of the production in time and make adjustments at the initial stage of performance degradation, so as to avoid unnecessary economic loss and ensure the improvement of benefits for practical applications.

However, all training data used in this study are labeled, and the actual production usually contains a large number of unlabeled samples. When there are not enough labeled samples to build a reliable model, unlabeled samples should be applied together with the labeled samples to improve the model performance. In future work, we will concentrate on some extensions to the proposed method. First, semi-supervised deep learning methods will be explored for the operating performance assessment combining labeled and unlabeled data. In addition, the information collected in the actual production process is usually multi-source heterogeneous information, including images, sounds, and so on, which involves feature extraction and feature fusion of different types of information in the operating performance assessment. In particular, it is very significant to provide a reasonable operation adjustment strategy when the process operating performance is not unsatisfactory, so the determination of adjustment strategy for non-optimal operating performance will also be investigated.

■ AUTHOR INFORMATION

Corresponding Author

Yan Liu − *College of Information Science & Engineering, Northeastern University, Shenyang 110819, China;* orcid.org/0000-0002-8003-8222; Email: liuyan@ise.neu.edu.cn

Authors

Zhe Ma − *College of Information Science & Engineering, Northeastern University, Shenyang 110819, China*

Fuli Wang − *College of Information Science & Engineering, Northeastern University, Shenyang 110819, China*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acsomega.3c00414

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank the National Key Research and Development Program of China (grant no. 2021YFF0602404), the National Natural Science Foundation of China (grant nos. 62073060, 61973057, 62073157, and 61973304), and the Fundamental Research Funds for the Central Universities (grant no. N2104011).

■ REFERENCES

(1) Liu, C.; Li, H.; Shi, Y. A unitary distributed subgradient method for multi-agent optimization with different coupling sources. *Automatica* **2020**, *114*, 108834.

(2) Ye, L.; Liu, Y.; Fei, Z.; Liang, J. Online probabilistic assessment of operating performance based on safety and optimality indices for multimode industrial processes. *Ind. Eng. Chem. Res.* **2009**, *48*, 10912−10923.

(3) Liu, Y.; Wang, F.; Chang, Y. Online fuzzy assessment of operating performance and cause identification of nonoptimal grades for industrial processes. *Ind. Eng. Chem. Res.* **2013**, *52*, 18022−18030.

(4) Chuah, L. F.; Chew, K. W.; Bokhari, A.; Mubashir, M.; Show, P. L. Biodegradation of crude oil in seawater by using a consortium of symbiotic bacteria. *Environ. Res.* **2022**, *213*, 113721.

(5) Zhao, C.; Wang, W.; Tian, C.; Sun, Y. Fine-scale modeling and monitoring of wide-range nonstationary batch processes with dynamic analytics. *IEEE Trans. Ind. Electron.* **2021**, *68*, 8808−8818.

(6) Chuah, L. F.; Mokhtar, K.; Bakar, A. A.; Othman, M. R.; Osman, N. H.; Bokhari, A.; Mubashir, M.; Abdullah, M. A.; Hasan, M. Marine environment and maritime safety assessment using Port State Control database. *Chemosphere* **2022**, *304*, 135245.

(7) Zhou, D.; Li, G.; Qin, S. J. Total projection to latent structures for process monitoring. *AIChE J.* **2010**, *56*, 168−178.

(8) Zhao, C.; Gao, F.; Niu, D.; Wang, F. A two-step basis vector extraction strategy for multiset variable correlation analysis. *Chemom. Intell. Lab. Syst.* **2011**, *107*, 147−154.

(9) Liu, Y.; Chang, Y.; Wang, F. Online process operating performance assessment and nonoptimal cause identification for industrial processes. *J. Process Control* **2014**, *24*, 1548−1555.

(10) Liu, Y.; Wang, F.; Chang, Y. Operating optimality assessment based on optimality related variations and nonoptimal cause identification for industrial processes. *J. Process Control* **2016**, *39*, 11−20.

(11) Chu, F.; Dai, W.; Shen, J.; Ma, X.; Wang, F. Online complex nonlinear industrial process operating optimality assessment using modified robust total kernel partial M-regression. *Chin. J. Chem. Eng.* **2018**, *26*, 775−785.

(12) Liu, Y.; Wang, F.; Chang, Y. Operating optimality assessment and cause identification for nonlinear industrial processes. *Chem. Eng. Res. Des.* **2017**, *117*, 472−487.

(13) Liu, Y.; Wang, X.; Wang, F.; Gao, F. Gaussian process regression and Bayesian inference based operating performance assessment for multiphase batch processes. *Ind. Eng. Chem. Res.* **2018**, *57*, 7232−7244.

(14) Zou, X.; Zhao, C. Meticulous assessment of operating performance for processes with a hybrid of stationary and nonstationary variables. *Ind. Eng. Chem. Res.* **2019**, *58*, 1341−1351.

(15) Zhong, L.; Chang, Y.; Wang, F.; Gao, S. Distributed operating performance assessment of the plant-wide process based on data-driven hybrid characteristics decomposition. *Ind. Eng. Chem. Res.* **2020**, *59*, 15682−15696.

(16) Zou, X.; Chang, Y.; Wang, F.; Zhao, L. Process operating performance optimality assessment with coexistence of quantitative and qualitative information. *Can. J. Chem. Eng.* **2018**, *96*, 179−188.

(17) Wu, H.; Han, Y.; Jin, J.; Geng, Z. Novel deep learning based on data fusion integrating correlation analysis for soft sensor modeling. *Ind. Eng. Chem. Res.* **2021**, *60*, 10001−10010.

(18) Xu, B.; Guo, F.; Zhang, W.; Li, G.; Wen, C. E^2 DNet: an ensembling deep neural network for solving nonconvex economic dispatch in smart grid. *IEEE Trans. Ind. Inf.* **2022**, *18*, 3066−3076.

(19) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436−444.

(20) Charte, D.; Charte, F.; García, S.; del Jesus, M. J.; Herrera, F.; Herrera, F. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Inf. Fusion* **2018**, *44*, 78−96.

(21) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735−1780.

(22) Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.; Bottou, L. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371−3408.

(23) Zhang, Z.; Jiang, T.; Li, S.; Yang, Y. Automated feature learning for nonlinear process monitoring−An approach using stacked denoising autoencoder and k-nearest neighbor rule. *J. Process Control* **2018**, *64*, 49−61.

(24) Yu, W.; Zhao, C. Robust monitoring and fault isolation of nonlinear industrial processes using denoising autoencoder and elastic net. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 1083−1091.

(25) Yu, J. Enhanced stacked denoising autoencoder-based feature learning for recognition of wafer map defects. *IEEE Trans. Semiconduct. Manuf.* **2019**, *32*, 613−624.

(26) Chai, Z.; Song, W.; Wang, H.; Liu, F. A semi-supervised autoencoder using label and sparse regularizations for classification. *Appl. Soft Comput.* **2019**, *77*, 205−217.

(27) Wang, Y.; Yang, H.; Yuan, X.; Shardt, Y.; Yang, C.; Gui, W. Deep learning for fault-relevant feature extraction and fault classification with stacked supervised auto-encoder. *J. Process Control* **2020**, *92*, 79−89.

(28) Yuan, X.; Zhou, J.; Huang, B.; Wang, Y.; Yang, C.; Gui, W. Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy. *IEEE Trans. Ind. Inf.* **2020**, *16*, 3721−3730.

(29) Chorowski, J.; Bahdanau, D.; Cho, K.; Bengio, Y. End-to-end continuous speech recognition using attention-based recurrent NN: first results. arXiv:1412.1602v1 **2014**, 1−10.

(30) Yuan, X.; Li, L.; Shardt, Y.; Wang, Y.; Yang, C. Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development. *IEEE Trans. Ind. Electron.* **2021**, *68*, 4404−4414.

(31) Zhu, H.; Ma, M.; Ma, W.; Jiao, L.; Hong, S.; Shen, J.; Hou, B. A spatial-channel progressive fusion ResNet for remote sensing classification. *Inf. Fusion* **2021**, *70*, 72−87.

(32) Hinton, G. E.; Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504−507.

(33) Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. Greedy Layer-Wise Training of Deep Networks. In *International Conference on Neural Information Processing Systems*, 2006; pp 153−160.

(34) Yan, T.; Shen, S.; Zhou, A.; Chen, X. Prediction of geological characteristics from shield operational parameters by integrating grid search and K-fold cross validation into stacking classification algorithm. *J. Rock Mech. Geotech. Eng.* **2022**, *14*, 1292−1303.

(35) Yang, M.; Wu, Y. A cross-validated feature selection (CVFS) approach for extracting the most parsimonious feature sets and discovering potential antimicrobial resistance (AMR) biomarkers. *Comput. Struct. Biotechnol. J.* **2023**, *21*, 769−779.

(36) Amir, S.; Mohammad, M.; Reza, D.; Maryam, N.; Mohammad, E.; Alireza, B.; Sajjad, H.; Eslamian, S. *Handbook of Hydroinformatics*; Candice Janco, 2023, pp 89−105.

(37) Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273−297.

(38) van der Maaten, L.; Hinton, G. E. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.