

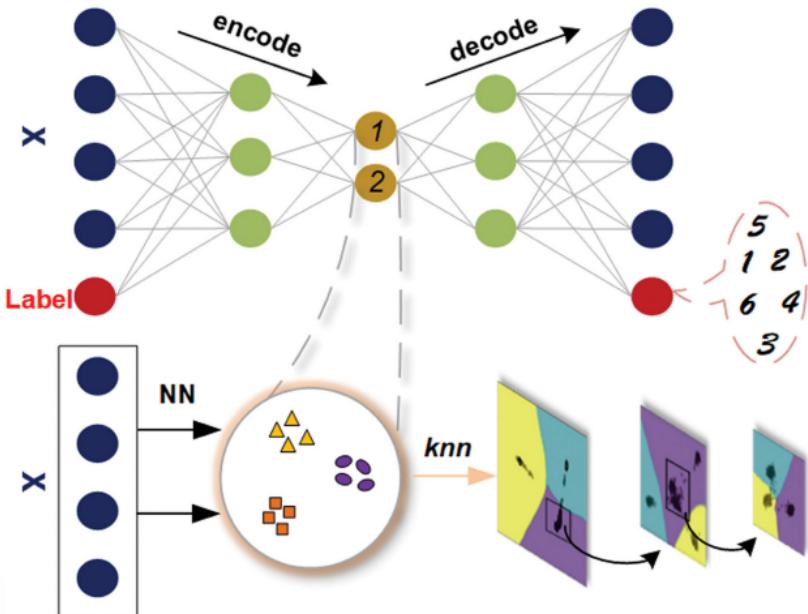
# I&EC research

Industrial & Engineering Chemistry Research

June 12, 2019

Volume 58, Number 23

[pubs.acs.org/IERC](https://pubs.acs.org/IERC)



Published by the American Chemical Society for Applied Chemistry and Chemical Engineering



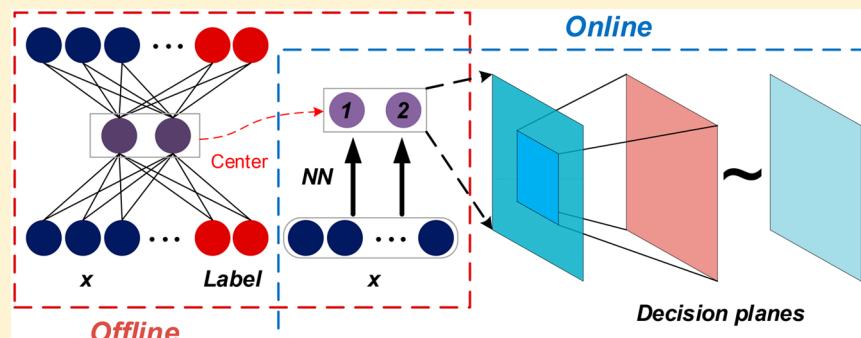
ACS Publications  
Most Trusted. Most Cited. Most Read.

[www.acs.org](https://www.acs.org)

# Using Labeled Autoencoder to Supervise Neural Network Combined with *k*-Nearest Neighbor for Visual Industrial Process Monitoring

Shifu Yan and Xuefeng Yan\*<sup>10</sup>

Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shanghai 200237, P. R. China



**ABSTRACT:** Data clustering and visualization are efficient methods for visual industrial process monitoring. Self-organizing map (SOM) and *t*-distributed stochastic neighbor embedding (*t*-SNE) are two of the most widely used algorithms. However, these methods do not make full use of label information. This paper presents a new strategy that uses labeled autoencoder (LAE) to supervise a neural network (NN) combined with *k*-nearest neighbor (*k*NN) for visual process monitoring. The LAE, trained simultaneously by process variables together with corresponding labels, extracts 2D features that are visualized. Then a feedforward NN is supervised by these features to reproduce the clustering results from process variables. After data clustering and visualization are implemented, the decision plane is further established using *k*NN for online process monitoring. In this way, the category of a new sample can be intuitively predicted based on the projected area on such a plane. For data with excessive categories, multilayer decision plane can be created one by one using multiple LAEs and NNs. The proposed strategy achieves satisfactory results on different cases of the Tennessee-Eastman process.

## 1. INTRODUCTION

With the large amount of data recorded, great progress has been achieved in industrial process monitoring using data-driven methods, including fault detection, identification, and isolation.<sup>1–4</sup> The most popular methods involve principal component analysis (PCA),<sup>5,6</sup> partial least-squares (PLS),<sup>7,8</sup> independent component analysis (ICA),<sup>9,10</sup> Fisher discriminant analysis (FDA),<sup>11</sup> and their variants. As dimensions of data increase exponentially, dimensionality reduction and feature extraction are gradually becoming the key implementations. These algorithms mainly extract efficient features based on their corresponding multivariate statistical theories. After dimensionality reduction, visualizing features in the low-dimension space has received extensive attention in recent years because of their benefits to engineers and operators. However, issues in visualizing the results of process monitoring are challenging because such low-dimension features may be insufficient to represent the original data.

For data clustering and visualization, one of the most widely used tools is the self-organizing map (SOM), a neural network (NN) with special structure.<sup>12</sup> Yu et al.<sup>13</sup> classified the operating condition based on the trajectory of samples on the SOM. Garcia and Gonzalez<sup>14</sup> adopted SOM and *k*-means for

monitoring wastewater treatment process. However, when process variables have high dimensionality, traditional SOM may not meet our needs. Thus, methods for data dimensionality reduction and feature analysis are considered to improve SOM. Chen et al.<sup>15</sup> used correlative component analysis (CCA) to analyze the relationship between variables before training SOM. Tang et al.<sup>16</sup> utilized the dynamic partial least-squares to handle the dynamic process and applied the extracted features to multi-SOM (MSOM). Except for SOM, *t*-distributed stochastic neighbor embedding (*t*-SNE) is another efficient algorithm that can effectively visualize high-dimension data in a 2D or 3D mapping plane.<sup>17</sup> For visual fault diagnosis, Tang and Yan used NNs to model the original data and low-dimension features extracted by FDA and *t*-SNE.<sup>18</sup> However, label information is not utilized during the training of these two methods. In addition, the proper size of SOM needs to be predefined and the transformation matrix of *t*-SNE is absent, which limits their application.

Received: March 9, 2019

Revised: May 20, 2019

Accepted: May 21, 2019

Published: May 22, 2019

As data dimensions continue to increase, feature extraction becomes more difficult for visualization. In this case, NNs have been a relatively effective solution because of their learning ability, especially for unsupervised learning.<sup>19</sup> Among NNs, autoencoder (AE) is a nonconvex technology for reducing dimensions and treated as a nonlinear PCA.<sup>20</sup> By minimizing the squared error between input and output, neurons in hidden layers can largely exploit both the linear and nonlinear characteristics of data. For more complex issues, multiple AEs can be stacked into a deep model to extract more abstract and useful features in view of deep learning ideas.<sup>21</sup> In the applications of process monitoring, Ma et al.<sup>22</sup> applied a deep coupling AE to multimodal sensors. Qi et al.<sup>23</sup> used stacked sparse AE to diagnose faults in rotating machinery. When 2D features of the original data are extracted, AE can implement data clustering and visualize the representative features in a mapping plane. In this way, strategy can be designed for visual process monitoring using AE.

This paper presents a new strategy for extracting more efficient features for clustering and visualization. The proposed strategy uses a labeled autoencoder to supervise a neural network (LAE-NN) combined with *k*-nearest neighbor (*k*NN) for data clustering and visual process monitoring. The labeled autoencoder (LAE) with two hidden neurons is designed and trained by process variables together with corresponding labels. Therefore, LAE can make full use of label information and distinguish different categories of data compared with SOM and *t*-SNE. After 2D features are obtained, clustering centers are calculated and another feedforward NN is supervised to reproduce the clustering results. Thus, data are clustered and visualized in a mapping plane. To determine the boundaries of different categories on this plane, *k*NN algorithm is utilized to upgrade the mapping plane to a decision plane. For data with excessive categories, multilayer decision planes can be created one by one using multiple LAEs with NNs. With the use of well-trained NN(s), categories of testing samples are predicted by mapping the samples into such decision plane(s). Judging different conditions in an industrial process is beneficial to operators.

The rest of this paper is arranged as follows. AE and *k*NN are introduced in section 2. The detailed steps of visual clustering and monitoring scheme are described in section 3. The proposed method is evaluated by different cases of the Tennessee–Eastman process (TEP) in section 4. Finally, conclusions are presented in section 5.

## 2. PRELIMINARIES

**2.1. Autoencoder.** Autoencoder (AE) is a three-layer NN with the same sizes of input and output (Figure 1). Assume that each sample of the input data with *m* variables is collected

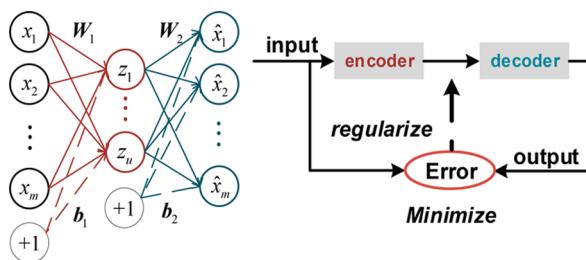


Figure 1. Structure of autoencoder and variants.

as  $\mathbf{x} = [x_1, x_2, \dots, x_m]^T \in \Re^m$ , and the output one is reconstructed as  $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m]^T \in \Re^m$ . The AE maps  $\mathbf{x}$  into the hidden layer as  $\mathbf{z} = [z_1, z_2, \dots, z_u]^T \in \Re^u$  using (1).

$$\mathbf{z} = \sigma_1(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \quad (1)$$

where  $\mathbf{W}_1 \in \Re^{u \times m}$  and  $\mathbf{b}_1 \in \Re^u$  are the weights and bias between the input and hidden layers, respectively. The activation function  $\sigma_1(\cdot)$  of neurons can be linear or nonlinear including sigmoid, tanh, rectified linear unit (relu), and so on. Similarly, the decoder reconstructs output  $\hat{\mathbf{x}}$  using hidden features  $\mathbf{z}$  based on (2).

$$\hat{\mathbf{x}} = \sigma_2(\mathbf{W}_2 \mathbf{z} + \mathbf{b}_2) \quad (2)$$

where  $\mathbf{W}_2 \in \Re^{m \times u}$  and  $\mathbf{b}_2 \in \Re^m$  are the weights and bias between hidden and output layers, respectively, and  $\sigma_2(\cdot)$  is the activation function of output neurons.

Constructed by the encoder and decoder, AE keeps the input and output consistent by optimizing the following objective function.

$$\min_{\theta} J_1(\theta) = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \quad (3)$$

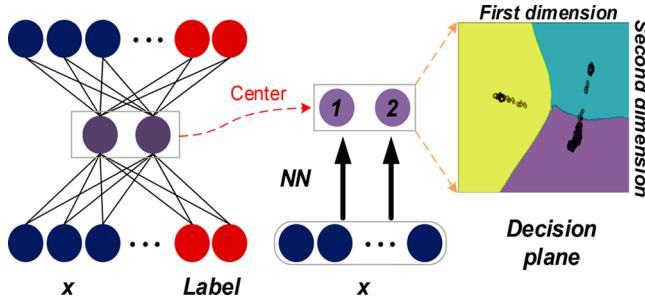
where *n* is the number of samples. The above parameters in (3),  $\theta = [\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2]$  can be trained by the backpropagation algorithm and other methods such as stochastic gradient descent (SGD), which are also efficient for searching the best solution. Therefore, the obtained features  $\mathbf{z}$  can largely represent the input data after the training phase.

As an unsupervised learning framework, AE is usually used for dimensionality reduction or feature extraction. When neurons or weights are regularized according to different principles, corresponding variants of AE can be derived, such as sparse AE, denosing AE, and variational AE. At the same time, these AEs can be stacked into a deep model that can mine additional information on data with strong learning ability.

**2.2. *k*-Nearest Neighbor.** *k*NN is a clustering algorithm whose main idea is that an unknown sample belongs to the major category of the *k* samples that own the smallest distance with it.<sup>24</sup> This algorithm is suitable for judging the unknown data visualized on a 2D plane. Generally, the distance can be measured by Euclidean distance and other principles.

## 3. SCHEME OF VISUAL PROCESS MONITORING

**3.1. Single LAE-NN.** Before training the models,  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T \in \Re^{n \times m}$  with *c* categories are collected and the corresponding labels are described as  $\mathbf{Y} = [y_1, y_2, \dots, y_n]^T \in \Re^{n \times c}$ . On the basis of the principal of one-hot encoding,  $y_k = [0, 0, \dots, 0, 1, 0, 0, \dots, 0]^T \in \Re^c$  is defined for the sample  $\mathbf{x}_k$  in the *k*th category where the *k*th element is one. To make full use of label information, process variables  $\mathbf{X}$  combined with labels  $\mathbf{Y}$  are both treated as the input of LAE, and 2D features  $\mathbf{z} = [z_1 z_2]^T$  are extracted for visualization (Figure 2). For direct representation of the original data, the activation function of the encoder and decoder in this study are tanh and linear, respectively. During training, the objective function  $J_2$  based on the mean square error (MSE) is defined as follows:



**Figure 2.** Structure of labeled autoencoder supervising neural network to create plane.

$$\min_{\theta} J_2(\theta) = \frac{1}{n} \left( \sum_{i=1}^n \|x_i - \hat{x}_i\|_2^2 + \alpha \|y_i - \hat{y}_i\|_2^2 \right) \quad (4)$$

where  $\theta$  contains the parameters that need to be trained in the NN.  $\alpha$  is the penalty for the learning error of labels, with the value depending on the number of categories and the difficulty of clustering. Specifically, excessive categories need a bigger  $\alpha$ . The features of  $x$  involving label information are further extracted, whereas the label-irrelevant part is eliminated. Data with the obtained features  $z = [z_1, z_2]^T$  can be easily distinguished between categories when features that do not represent the label information are removed.

Afterward, the clustering centers  $C = [c_1, c_2, \dots, c_n]^T \in \mathbb{R}^{n \times 2}$  of different categories are calculated based on the corresponding 2D features. To predict the categories of unknown samples, another feedforward NN is supervised by the centers to reproduce the clustering results using process variables. The boundaries of the features with different categories on the same plane should also be specified for process monitoring. Thus, objective function  $J_3$  for this NN is defined as

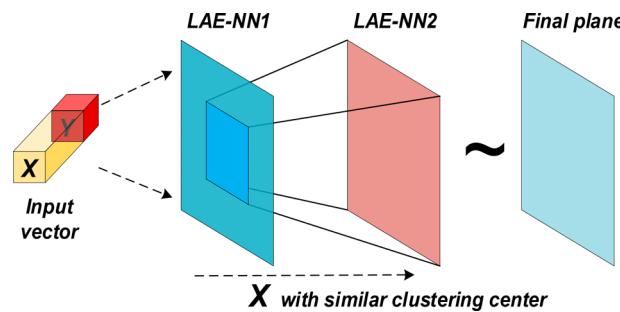
$$\min_{\theta} J_3(\theta) = \frac{1}{n} \left( \sum_{i=1}^n \|\hat{z}_i - c_i\|_2^2 + \beta \max\{0, \|\hat{z}_i - c_i\|_2^2 - R_i\} \right) \quad (5)$$

where  $\beta$  is also a penalty for samples that deviated far from the clustering centers and  $R_i$ , decided manually, is the radius for corresponding categories. If the boundaries of different categories are unclear,  $R_i$  will tend to be a small value. In a direct way,  $R_i$  can be set to the same value  $R$  for different categories in this study. The output  $\hat{z} = [\hat{z}_1, \hat{z}_2]^T$  of the well-trained NN can be visualized in a 2D plane.

In this way, data clustering and visualization can be completed. During the classification phase, the area of different categories on such plane needs to be clearly determined. Therefore, kNN algorithm is adopted here. First, mesh the plane and predict the category of each grid based on Euclidean distance. Then, divide the area of different categories to create the decision plane (Figure 2). For testing, the 2D features of new samples are calculated and then categories can be predicted on the decision plane.

**3.2. Multilayer LAE-NN.** For data sets with few categories, a single decision plane created by LAE-NN is enough for classification. However, the clustering and classification effects will worsen if the categories increase. Generally speaking, if

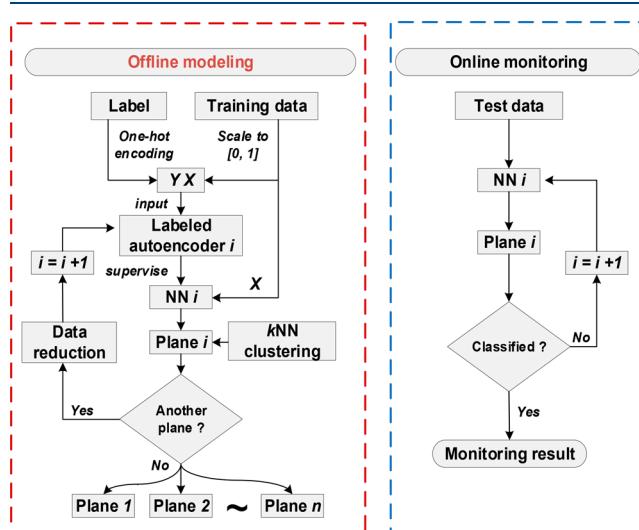
data sets involve categories that are easy and difficult to distinguish, the clustering results of latter will be affected by the former and will become indistinguishable. Therefore, multilayer planes are created by multiple LAEs and NNs. In this case, the indistinguishable data sets can be treated as the same category in the former decision plane and further put into the next plane created by the second LAE-NN. As shown in Figure 3, the numbers of LAEs and NNs increase as the data



**Figure 3.** Decision planes created by multilayer LAE-NN.

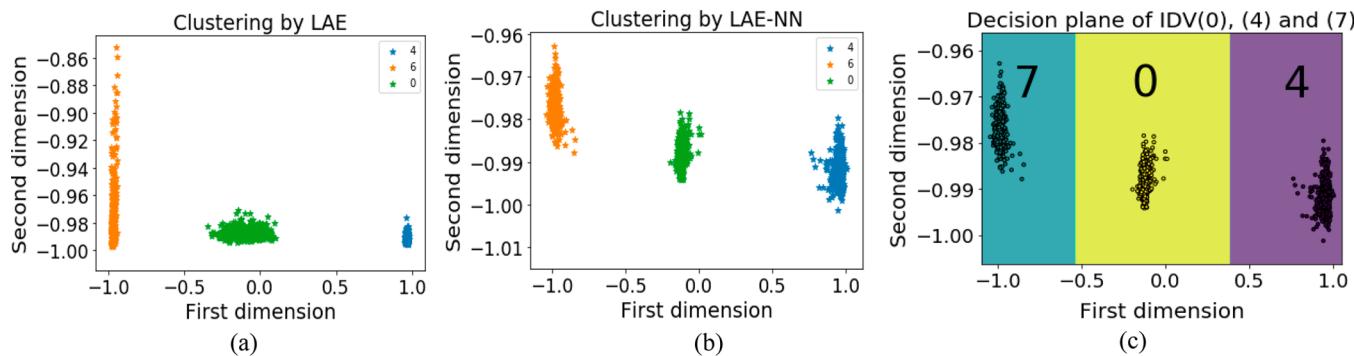
categories increase. If the rest of the data can be distinguished to a certain extent or no more category can be distinguished, the models can be completed.

**3.3. Flowchart.** The simplified flowchart of the proposed scheme for visual process monitoring is shown in Figure 4. Two parts are included, offline modeling and online monitoring, and the steps are described as follows.

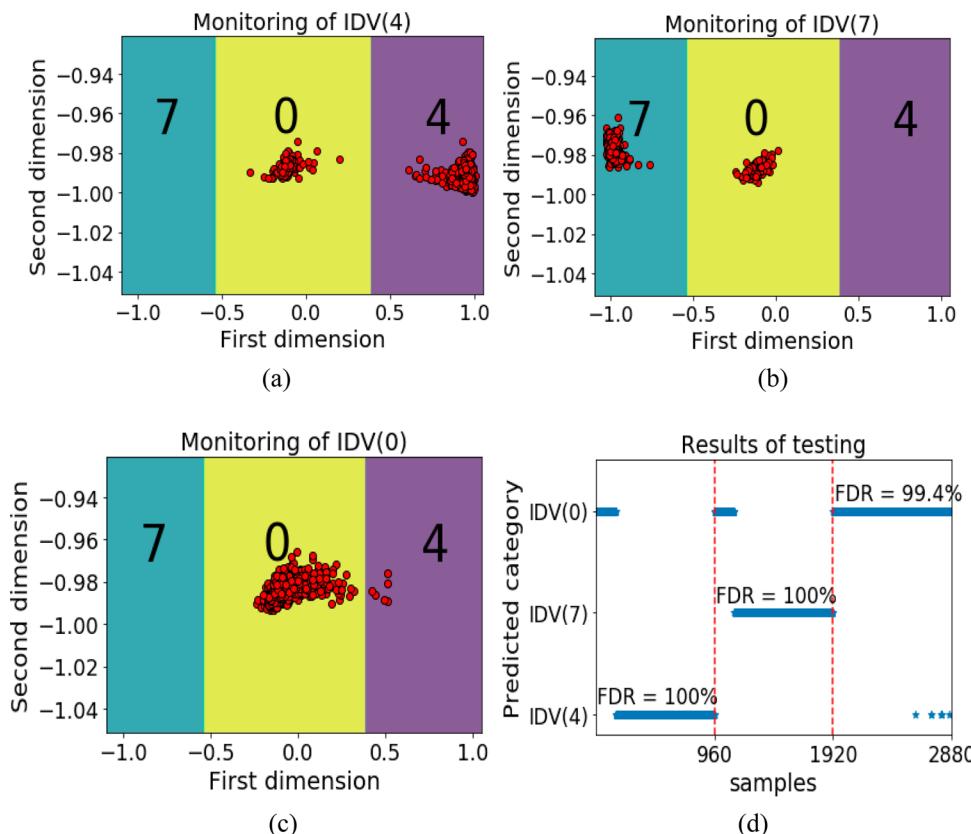


**Figure 4.** Flowchart of LAE-NN for process monitoring.

**Part 1: Offline modeling.** (1) Scale the training data  $X$  to  $[0,1]$  and define the corresponding label matrix  $Y$ ; (2) Train a LAE using  $[X,Y]$  and extract the 2D features  $z$ ; (3) Calculate the clustering centers  $C$  of different categories; (4) Use  $C$  to supervise another feedforward NN and obtain corresponding features  $\hat{z}$ ; (5) Mark the indistinguishable data as the same category and relabel the relevant data; (6) Run kNN algorithm based on  $\hat{z}$  and new labels to create a decision plane; and (7) Perform matrix reduction with the indistinguishable data sets and return to step 1) until the rest of the data can be distinguished to a certain extent, or no more category can be distinguished.



**Figure 5.** Offline modeling of IDV (0), (4), and (7). (a) Data clustering using LAE. (b) Data clustering using LAE-NN. (c) Decision plane created by kNN.



**Figure 6.** Online monitoring of IDV (0), (4), and (7). (a) Monitoring of IDV (4). (b) Monitoring of IDV (7). (c) Monitoring of IDV (0). (d) FDRs (%) of testing samples.

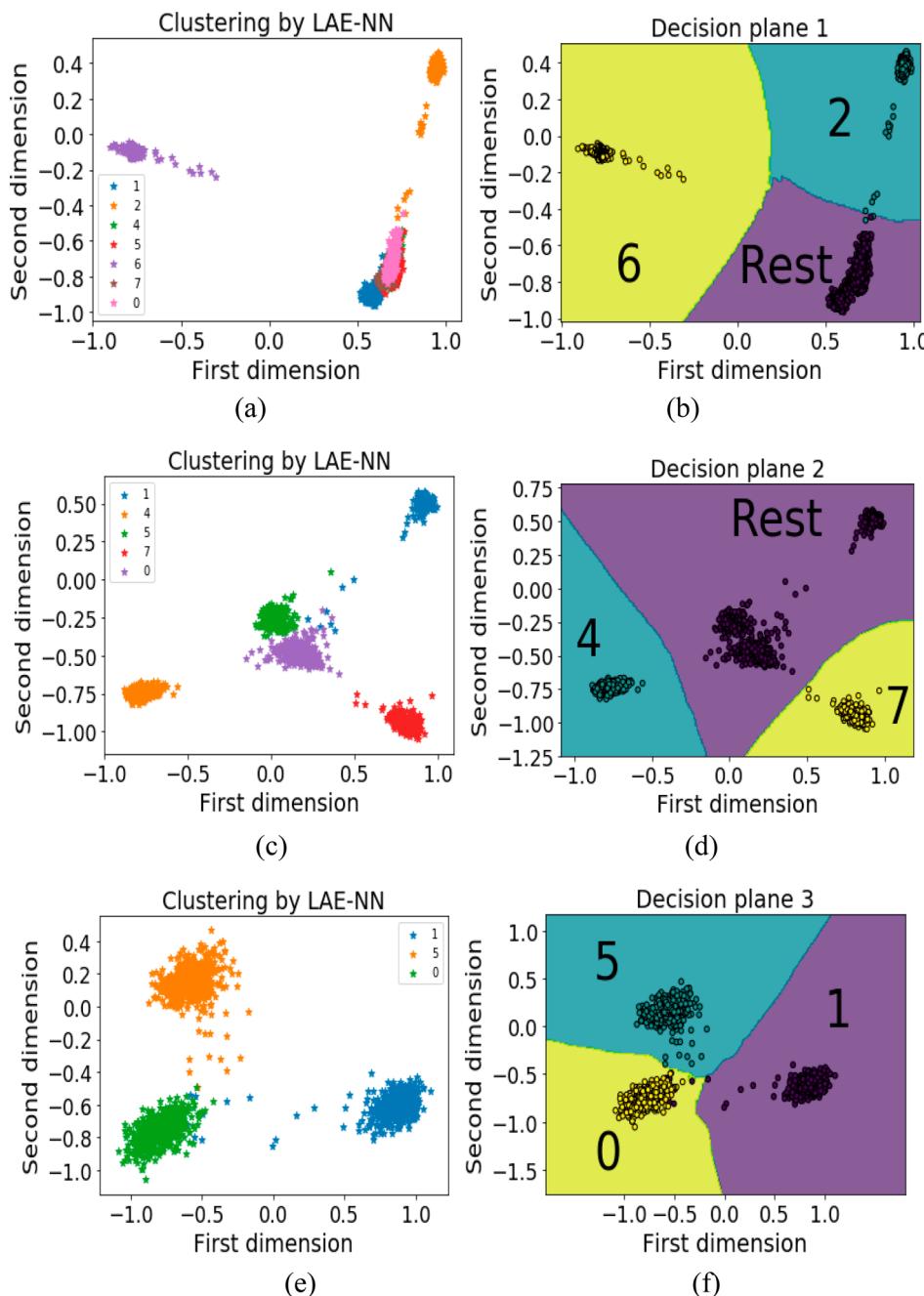
**Part 2: Online monitoring.** (1) Scale the testing sample  $x_{\text{new}}$ ; (2) Calculate the 2D features  $\hat{z}_{\text{new}}$  using the well-trained NN(s); (3) Predict the category in the mapping plane(s) one by one; and (4) Make a decision based on the results of fault diagnosis and monitoring.

#### 4. CASE STUDY

**4.1. Tennessee-Eastman Process.** TEP has been widely used to evaluate algorithms for process monitoring.<sup>25,26</sup> The complex system consists of five units involving four reactants and two products. As a benchmark, data sets under 21 faulty conditions together with a normal one are generated and collected. In addition, 52 variables are selected, including 41 measurements and 11 manipulated variables for the following experiments. Generally, 480 samples under 24 h are used to

train models and 960 samples under 48 h are adopted for testing. In the testing data, faults are introduced in the ninth hour, which means the former 160 samples are collected under normal conditions. The following experiments are run on a computer with Intel Xeon Silver 4110 2.1 GHz, 64GB RAM, and NVIDIA Quadro P600. The simulation code for TEP are available on <http://depts.washington.edu/control/LARRY/TE/download.html>.

**4.2. Experiment on IDV (0), (4), and (7).** The proposed strategy is illustrated using IDV (0), (4), and (7) as examples. Therefore, dimensions of process variables and corresponding labels are  $S_2$  and 3, respectively.  $X_1^{\text{train}} \in \mathbb{R}^{1440 \times 52}$  and  $Y_1 \in \mathbb{R}^{1440 \times 3}$  are used to train the LAE to acquire the clustering centers that are learned by a NN using  $X_1^{\text{train}}$  (Figure 5a,b, respectively). The obtained features are then used to



**Figure 7.** Offline modeling of IDV (0), (1), (2), (4), (5), (6), and (7). (a,b) Clustering result of decision plane 1. (c,d) Clustering result of decision plane 2. (e,f) Clustering result of decision plane 3.

create the decision plane by kNN (Figure 5c). During the training phase,  $\alpha = 1$ ,  $\beta = 0.5$ , and  $R = 0.4$  are set in (4) and (5). The structure of this NN is 52–36–25–15–2. Through grid search, the parameter  $k = 10$  is adopted for kNN algorithm. At the same time, the plane is meshed by the steps of 0.01.

In visual process monitoring, the 2D characteristics of samples in three data sets  $X_1^{\text{test}} \in \Re^{2880 \times 52}$  are mapped in Figure 6a–c. As mentioned above, each data set consists of 160 normal samples and 800 faulty samples. Obviously, faults can be well detected and classified. In addition, the trend of sample variations can be analyzed by the trajectories based on the sampling order. The fault diagnosis rate (FDR) defined in (6) of these faults is summarized in Figure 6d. For the three

categories, the decision plane can achieve the satisfied clustering and classification results.

$$\text{FDR} = \frac{\text{number of samples truly diagnosed}}{\text{number of faulty samples}} \quad (6)$$

**4.3. Experiment on Faults with the Same Type.** To evaluate the performance of LAE-NN on faults with the same type, step-kind faults including IDV (0), (1), (2), (4), (5), (6), and (7), are used in this section. Similarly,  $X_2^{\text{train}} \in \Re^{3360 \times 52}$  with  $Y_2 \in \Re^{3360 \times 7}$  are collected to obtain the clustering results by LAE-NN. kNN is then run to create the decision plane in Figure 7b. Among the parameters are  $\alpha = 1$ ,  $\beta = 0.5$ , and  $R = 0.1$ . In plane 1, IDV (2) and (6) can be well

distinguished, whereas the rest of the categories need further training. Thus, the rest are relabeled as the same one for decision plane 1. Two rounds of the above steps are repeated to create planes 2 and 3 in Figure 7c–f.

For comparison, the FDRs (%) of LAE-NN, SOM, CCA-SOM, and various statistical methods<sup>15</sup> on these faults are summarized in Table 1. The LAE-NN achieved perfect

**Table 1. FDRs (%) of LAE-NN and Other Methods on the Step-Kind Faults<sup>a</sup>**

fault	LAE-NN	SOM	CCA-SOM	Chiang (worst/best)
IDV (0)	99.8	60	<b>100</b>	—
IDV (1)	<b>99.3</b>	92.5	97.5	12.0/98.7
IDV (2)	98.6	97.5	97.5	55.9/ <b>99.0</b>
IDV (4)	<b>100</b>	65	97.5	0.0/88.1
IDV (5)	<b>99.9</b>	60	97.5	0.0/99.4
IDV (6)	<b>100</b>	97.5	<b>100</b>	16.6/ <b>100</b>
IDV (7)	99.9	42.5	95	2.2/ <b>100</b>
avg	<b>99.64</b>	73.57	97.86	14.45/97.53

<sup>a</sup>The bold font represents the best results.

performance in the step-kind faults versus other methods. In other words, the 2D features obtained by LAE-NN can preserve the potential characteristics of data with step-kind fault. The clustering centers supervised by the labels can be well learned by a deep NN due to its large capacity and learning ability.

**4.4. Experiment on Faults with Different Types.** For faults with the step-kind effect, the proposed strategy outperforms some other methods. Next, data with different faulty types are adopted including IDV (0), (1), (2), (4)–(8), (10)–(12), (14), (17), and (18). Except for the faults described before, IDV (8) and (10)–(12) are caused by random variation and IDV (14) and (15) are sticking faults. As for IDV (16)–(18), the faulty types are still unclear. In total, 6720 and 11,200 samples are involved in the training and testing phases, respectively. At the same time,  $\alpha = 2$ ,  $\beta = 0.5$ , and  $R = 0.05$  are set. The process of training and testing is no longer repeated here. The FDRs (%) in comparison with multimodel PCA (MPCA), multimodel nonlinear PCA (MNLPCA), SOM, and MSOM<sup>27</sup> are summarized in Table 2.

LAE-NN clearly works better in step-kind faults than in other types, especially in random variation. Thus, the LAE is more efficient in faults that occur in the same direction than in multidirection faults, such as random variation. Because of random variation, the clustering and classification of some faults, like IDV (3), (5), and (9) are largely affected. Therefore, they are not considered here. Although the proposed strategy does not perform well on all faults, satisfactory results have been achieved in most cases, which are notable and worth exploring in the future.

## 5. CONCLUSIONS

A new strategy for visual process monitoring using LAE-NN combined with kNN is presented in this paper. With the labels, LAE extracts features containing label information that are beneficial for clustering. Thus, visualization of data belonging to different categories can be implemented under the unsupervised learning mechanism. Afterward, the clustering results are learned by NNs without labels because of their strong learning ability. Therefore, the original data are mapped into the 2D plane, and boundaries of different categories are

**Table 2. FDRs (%) of LAE-NN and Other Methods on the Faults with Different Types<sup>a</sup>**

fault	LAE-NN	MPCA	MNLPCA	SOM	MSOM
IDV (0)	89.8	—	—	—	—
IDV (1)	97.4	96.9	96.6	98.0	<b>99.3</b>
IDV (2)	<b>98.0</b>	1.6	23.1	90.5	97.3
IDV (4)	<b>99.3</b>	1.7	50.9	97.4	97.4
IDV (5)	<b>98.3</b>	0.0	0.0	21.7	1.1
IDV (6)	<b>99.8</b>	94.4	98.6	94.4	0.0
IDV (7)	98.6	99.0	99.9	99.7	<b>100</b>
IDV (8)	42.4	79.5	61.8	74.4	<b>92.9</b>
IDV (10)	<b>89.9</b>	0.1	61.7	73.9	88.1
IDV (11)	23.4	66.7	89.4	58.3	<b>92.1</b>
IDV (12)	<b>59.5</b>	0.0	13.2	51.2	57.9
IDV (14)	79.8	30.0	73.5	79.6	<b>99.1</b>
IDV (17)	43.4	59.8	88.0	87.8	<b>96.8</b>
IDV (18)	<b>85.4</b>	1.7	60.2	71.0	82.0
avg	<b>78.93</b>	40.88	62.84	76.76	77.23

<sup>a</sup>The bold font represents the best results.

determined using kNN. In accordance with the results of the experiments, LAE-NN exhibits satisfactory performance for visual process monitoring. The topic is thus worth further study because only the basic algorithms are used.

## ■ AUTHOR INFORMATION

### Corresponding Author

\*Address: P.O. Box 293, MeiLong Road No. 130, Shanghai 200237, P. R. China. Tel: 0086-21-64251036. E-mail: [xfyang@ecust.edu.cn](mailto:xfyang@ecust.edu.cn).

### ORCID

Xuefeng Yan: [0000-0001-5622-8686](https://orcid.org/0000-0001-5622-8686)

### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

The authors are grateful for the support of the National Natural Science Foundation of China (21878081) and Fundamental Research Funds for the Central Universities under Grant of China (222201917006).

## ■ REFERENCES

- (1) Qin, S. J. Survey on data-driven industrial process monitoring and diagnosis. *Annual Review Control* **2012**, *36* (2), 220–234.
- (2) Ge, Z.; Song, Z.; Gao, F. Review of Recent Research on Data-Based Process Monitoring. *Ind. Eng. Chem. Res.* **2013**, *52* (10), 3543–3562.
- (3) Wang, Y.; Si, Y.; Huang, B.; Lou, Z. Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017. *Can. J. Chem. Eng.* **2018**, *96* (10), 2073–2085.
- (4) Yin, S.; Li, X.; Gao, H.; Kaynak, O. Data-Based Techniques Focused on Modern Industry: An Overview. *IEEE Transactions on Industrial Electronics* **2015**, *62* (1), 657–667.
- (5) Zhao, C.; Gao, F. Fault-relevant principal component analysis (FPCA) method for multivariate statistical modeling and process monitoring. *Chemom. Intell. Lab. Syst.* **2014**, *133*, 1–16.
- (6) Jiang, Q.; Huang, B.; Yan, X. GMM and optimal principal components-based Bayesian method for multimode fault diagnosis. *Comput. Chem. Eng.* **2016**, *84*, 338–349.
- (7) Ma, M.; Khatibisepehr, S.; Huang, B. A Bayesian Framework for Real-Time Identification of Locally Weighted Partial Least Squares. *AIChE J.* **2015**, *61* (2), 518–529.

- (8) Li, G.; Qin, S. J.; Zhou, D. Geometric properties of partial least squares for process monitoring. *Automatic* **2010**, *46* (1), 204–210.
- (9) Tong, C.; Palazoglu, A.; Yan, X. Improved ICA for process monitoring based on ensemble learning and Bayesian inference. *Chemom. Intell. Lab. Syst.* **2014**, *135*, 141–149.
- (10) Wang, J.; Zhang, Y.; Cao, H.; Zhu, W. Dimension reduction method of independent component analysis for process monitoring based on minimum mean square error. *J. Process Control* **2012**, *22* (2), 477–487.
- (11) Yu, J. Localized Fisher Discriminant Analysis Based Complex Chemical Process Monitoring. *AIChE J.* **2011**, *57* (7), 1817–1828.
- (12) Brito Da Silva, L. E.; Wunsch, D. C. An Information-Theoretic-Cluster Visualization for Self-Organizing Maps. *IEEE Transactions on Neural Network and Learning Systems* **2018**, *29* (6), 2595–2613.
- (13) Yu, H.; Khan, F.; Garaniya, V.; Ahmad, A. Self-Organizing Map Based Fault Diagnosis Technique for Non-Gaussian Processes. *Ind. Eng. Chem. Res.* **2014**, *53* (21), 8831–8843.
- (14) Lopez Garcia, H.; Machon Gonzalez, L. Self-organizing map and clustering for wastewater treatment monitoring. *Engineering Applications of Artificial Intelligence* **2004**, *17* (3), 215–225.
- (15) Chen, X.; Yan, X. Using improved self-organizing map for fault diagnosis in chemical industry process. *Chemical Engineering Research & Design* **2012**, *90* (12), 2262–2277.
- (16) Tang, J.; Yan, X. DPLS-MSOM Modeling for Visual Industrial Fault Diagnosis and Monitoring Based on Variation Data from Normal to Anomalous. *Ind. Eng. Chem. Res.* **2017**, *56* (49), 14518–14529.
- (17) Van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **2008**, *9*, 2579–2605.
- (18) Tang, J.; Yan, X. Neural network modeling relationship between inputs and state mapping plane obtained by FDA-t-SNE for visual industrial process monitoring. *Applied Soft Computing* **2017**, *60*, 577–590.
- (19) Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *35* (8), 1798–1828.
- (20) Hinton, G.; Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313* (5786), 504–507.
- (21) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521* (7553), 436–444.
- (22) Ma, M.; Sun, C.; Chen, X. Deep Coupling Autoencoder for Fault Diagnosis With Multimodal Sensory Data. *IEEE Transactions on Industrial Informatics* **2018**, *14* (3), 1137–1145.
- (23) Qi, Y.; Shen, C.; Wang, D.; Shi, J.; Jiang, X.; Zhu, Z. Stacked Sparse Autoencoder-Based Deep Network for Fault Diagnosis of Rotating Machinery. *IEEE Access* **2017**, *5*, 15066–15079.
- (24) Liu, Z.; Pan, Q.; Dezert, J. A new belief-based K-nearest neighbor classification method. *Pattern Recognition* **2013**, *46* (3), 834–844.
- (25) Downs, J. J.; Vogel, E. F. A PLANT-WIDE INDUSTRIAL PROCESS CONTROL PROBLEM. *Comput. Chem. Eng.* **1993**, *17* (3), 245–255.
- (26) Ricker, N. L.; Lee, J. H. Nonlinear model predictive control of the Tennessee Eastman challenge process. *Comput. Chem. Eng.* **1995**, *19* (9), 961–981.
- (27) Robertson, G.; Thomas, M. C.; Romagnoli, J. A. Topological preservation techniques for nonlinear process monitoring. *Comput. Chem. Eng.* **2015**, *76*, 1–16.