IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 21, 2020, accepted May 30, 2020, date of publication June 4, 2020, date of current version June 17, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3000004*

# Deep Learning With Tensor Factorization Layers for Sequential Fault Diagnosis and Industrial Process Monitoring

**LIN LUO[1,2], LEI XIE[2], AND HONGYE SU[2], (Senior Member, IEEE)**
[1]School of Information and Control Engineering, Liaoning Shihua University, Fushun 113001, China
[2]State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

Corresponding author: Lei Xie (leix@iipc.zju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703191, in part by the Foundation of Liaoning Educational Committee under Grant L2017LQN028, and in part by the Scientific Research Foundation of Liaoning Shihua University under Grant 2017XJJ-012.

**ABSTRACT** Fault diagnosis technology is crucial to ensure the long-term reliability of the industrial process control system. With the increase of industrial data availability, conventional monitoring approaches may not function well under the assumption that the training and the application data come from the same distribution. Following the intuition that industrial data exhibit time dependency and inherent complex characteristics, this paper proposes an adaptive sequential fault diagnosis method based on a tensor factorization layer merged with deep neural network model (TF-DNN). Tensor representation is firstly applied to preserve the number of the raw data entries and their sequential dependence between observations. Multilinear mapping with tensor-to-tensor projection is then to transform the input and hidden tensor to the low-dimensional tensors, which makes the learned representations sparser with the grid-like structures. With the tensor factorization layers, the proposed deep network shares efficient knowledge across the spatiotemporal features of fault data. The outstanding performance of this method is demonstrated and compared to the existing models in the benchmark Tennessee Eastman process and a real industrial methanol plant.

**INDEX TERMS** Fault detection and diagnosis, industrial chemical process, deep learning, tensor factorization layer, feature representation.

## I. INTRODUCTION

With the progress of computer technique, electronics and information technology, the modern industrial control systems tend to be large-scale, multivariable and complex. To attain desired operational goals and economic benefits, great importance has been attached to the process monitoring of the industrial process control system, especially fault detection and diagnosis (FDD). In general, there are three categories of classical FDD methods: model-, data-, and knowledge-based method. In practice, almost all FDD strategies require process data for validation. Compared with the other two methods, data-driven method is well suitable for distilling the data into knowledge without the dependency of mechanistic model and expert knowledge [1]–[3]. Therefore, data-driven models have become considerable growth over the last couple of decades.

In recent decades, a wide variety of multivariate statistical techniques have been widely applied to data-driven process monitoring. The representative examples are the latent variable techniques, including principal component analysis (PCA) [4], partial least squares (PLS) [5] and independent component analysis (ICA) [6]. For the emergence of ''data rich but information poor'' problem [7] in the process monitoring task, latent variable models are particularly suitable to extract a low-dimensional representation from routine operating data. However, for the nonlinear and multimodal problem in a real industrial scenario, traditional latent variable methods often fail to meet the expectation due to the inadequacies of the resulting linear feature representation. The kernel trick [8]–[11] successfully extends these methods to nonlinear cases by performing a nonlinear transformation of the original variables into a higher dimensional feature space,

The associate editor coordinating the review of this manuscript and approving it for publication was Qingchao Jiang.

which is implicitly defined by a kernel function. However, the single feature extraction step is often difficult to learn the total information in large-scale complex processes, especially in the era of big data. In other words, the feature extraction of the kernel trick can be regarded as shallow learning networks, thereby limiting their capability to exploit not only the temporal information but also the spatial information.

Due to the remarkable representation ability and the stacking structure, deep neural network (DNN) has become an outstanding technology and a powerful solution to distil useful knowledge from large-scale data [12]. Intuitively, DNNs utilize the hierarchical hidden layers as a feature learning module to extract features. Although there exists simple nonlinear transformation on each hidden layer, the stacking structure of these transformations is capable of capturing valuable information from highly complex systems. Consequently, supervised DNN is usually trained with an end-to-end manner which learns the interpretable feature representation and classification simultaneously. In recent years, DNNs have attracted researchers' and engineers' attention and have been recognized as a useful technique for fault detection and diagnosis. For example, Zhang and Zhao [13] exploited the deep belief network (DBN) to build the FDD strategy for a chemical benchmark process. They utilized the mutual information technology to overcome the curse of dimensionality and complexities in each sub-network. Zhao *et al.* [14] studied the fault diagnosis of chemical process data by using long short-term memory (LSTM) and batch normalization. Wu and Zhao [15] investigated a deep convolutional neural network to extract the features in both spatial and temporal domains. Zhou *et al.* [16] designed novel generator and discriminator of Generative Adversarial Network (GAN) for the influence of unbalanced faults in the chemical process. They extracted unbalanced fault features in the generator, which consists of Auto Encoder (AE) rather than the samples. Zhao *et al.* [12] reviewed the application of DNNs to machine health monitoring systems.

The most-used layer in the networks may be the full-connected (FC) one where neurons have connections to all activations in the previous layer. The neurons organized in an FC input layer are often arranged as a vector form, where the input often corresponds to the information of a channel or a sensor. In such a case, the subsequent feature learning on FC layers is for the spatial domain. In essence, FDD is a spatio-temporal sequence forecasting problem with the sequence of past sensors as input and the sequence of future sensors as output. The performance may be degraded, especially when both the high dimensionality and multi-step predictions emerged in FDD development. The success of the DNNs in sequential fault diagnosis makes it consider a further improvement on the capacity of a network, including the width and depth in the hidden layers [17]. A natural way to construct wider layers is to increase the size of hidden units, and it leads to the greater representational capacity of the model. However, the number of parameters grows exponentially with the number of units. The depth of DNN architecture is associated with the number of layers, and a desired depth can be acquired by stacking multiple layers. Parts of the information from the input, however, are potentially lost due to gradient vanishing. Tensor factorization [18]–[20] is a useful tool for finding latent structures in tensor. It puts forward the relative measures to restrict the parameter number in the case of widening the network. On the other hand, the vanishing gradient problem can be mitigated through the use of the rectified linear unit (ReLU) activation function.

In this paper, we propose a novel end-to-end DNN based FDD framework which adopts a tensor factorization layer to compress the dense FC layer. The resulting layer does not only enable efficient knowledge sharing across the spatiotemporal features of fault data, but also implicitly prunes redundant dense connections. The main contributions of this paper are outlined as follows: 1) the proposed framework utilizes the multilinear discriminant analysis (MDA) on the layers to widen the neural network and share the parameters across different the temporal features. 2) By stacking the multiple layers, an end-to-end trainable model is built for the application of large-scale complex chemical data. 3) We experimentally show that the proposed approach achieves the promising performance compared with the state-of-the-art models on multiclass Tennessee Eastman benchmark datasets. 4) We experimentally show that the DNN applying the tensor factorization layer consistently outperforms the state-of-the-art recurrent neural network and its variant with the FC layers for a real-life industrial methanol plant.

This paper is organized as follows. Section II briefly summarizes the mathematical notation and the tensor factorization with the subspace constraints. Section III introduces ideas for the stacking deep network design based on the tensor factorization layers in detail. The experimental results on the benchmark Tennessee Eastman process are presented to demonstrate the superiority of our proposed method in Section IV. A real-life industrial methanol plant is employed to demonstrate the feasibility and effectiveness of the proposed model in Section V. Finally, conclusions are discussed in Section VI.

## II. TENSORIZATION OPERATION WITH MULTILINEAR DISCRIMINANT ANALYSIS

MDA is the extended version of Linear Discriminant Analysis (LDA) which is a classical supervised linear projection technique. MDA [19] learns a tensor subspace aiming to maximize the interclass distances while minimize the intraclass distances. To measure the separation of samples in the tensor subspace, the ratio between interclass distances and intraclass distances are usually maximized.

Specifically, consider a high-dimensional input can be tensorized by a set of $N$ labeled tensor data samples $\mathcal{X}_1, \cdots, \mathcal{X}_N \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}, n = 1, \cdots, N$ with class labels $c_i, i = 1, \cdots, C$, where $C$ is the class number in total and $K$ is the number of dimensions (ways) of the tensor. Let $\mathcal{X}_{i,j}$ represents the $j$-th sample from class $c_i$, $n_i$ is the

samples number in class $c_i$, the mean tensor of class $c_i$ is $\mathcal{M}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathcal{X}_{i,j}$ and the mean tensor of all samples can be denoted as $\mathcal{M} = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\mathcal{X}_{i,j} = \frac{1}{N}\sum_{i=1}^{C} n_i\mathcal{M}_i$.

The MDA objective is to find a tensor-to-tensor projection $\{\mathbf{U}_k \in \mathbb{R}^{I_k\times P_k}, k = 1,\cdots,K, P_k < I_k\}$ that maps the original tensor space $\mathcal{X}_{i,j}$ into a tensor subspace $\mathcal{W}_{i,j}$. For the projection of a tensor $\mathcal{X}_{i,j}$ to another tensor $\mathcal{W}_{i,j}$ in a lower-dimensional tensor space for all $k, K$ projection matrices $\{\mathbf{U}_k\}$ are used,

$$\mathcal{W}_{i,j} = \mathcal{X}_{i,j}\times_1 \mathbf{U}_1^T \times_2 \cdots \times_K \mathbf{U}_K^T \tag{1}$$

such that the ratio of the interclass and intraclass distances defined as follow is maximized

$$J(\mathbf{U}_k^*) = \arg\max_{\{\mathbf{U}_k\}} \frac{\sum_{i=1}^{C} n_i\|\bar{\mathcal{W}}_i - \bar{\mathcal{W}}\|_F^2}{\sum_{i=1}^{C}\sum_{j=1}^{n_i}\|\mathcal{W}_{i,j} - \bar{\mathcal{W}}\|_F^2} \tag{2}$$

where $\bar{\mathcal{W}}_i$ is the mean tensor of class $c_i$, such that,

$$\begin{aligned}\bar{\mathcal{W}}_i &= \frac{1}{n_i}\sum_{j=1}^{n_i}\mathcal{W}_{i,j} = \frac{1}{n_i}\sum_{j=1}^{n_i}\left(\mathcal{X}_{i,j}\prod_{k=1}^{K}\times_k\mathbf{U}_k^T\right)\\ &= \mathcal{M}_i\prod_{k=1}^{K}\times_k\mathbf{U}_k^T\end{aligned} \tag{3}$$

and $\bar{\mathcal{W}}$ is the overall mean given by

$$\bar{\mathcal{W}} = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\mathcal{W}_{i,j} = \mathcal{M}\prod_{k=1}^{K}\times_k\mathbf{U}_k^T \tag{4}$$

From the projection in Equation (1), there exists a dependency between each mode-$k$ vector. The dependency makes difficult to iteratively solve the optimization in Equation (2). In order to solve the dependency problem, we use the orthogonal constraints [20] on each projection, that is,

$$\mathbf{U}_k^T\mathbf{U}_k = \mathbf{I} \tag{5}$$

and learn the optimal tensor subspace $\mathcal{W}_{i,j}$ by iteratively maximizing the following equation

$$J(\mathbf{U}_k^*) = \arg\max_{\{\mathbf{U}_k\}} \frac{\mathrm{tr}\left(\mathbf{U}_k^T S_b^k\mathbf{U}_k\right)}{\mathrm{tr}\left(\mathbf{U}_k^T S_w^k\mathbf{U}_k\right)} \tag{6}$$

where $S_b^k$ and $S_w^k$ denote the interclass and intraclass scatter matrices in mode-$k$, $\mathrm{tr}(\cdot)$ is the trace operator. The interclass and intraclass scatter matrices can be obtained by mode-$k$ unfolding,

$$\begin{aligned}S_b^k = \sum_{i=1}^{C} n_i &\left[(\mathcal{M}_i - \mathcal{M})\prod_{k=1,k\neq n}^{K}\times_k\mathbf{U}_k^T\right]_{(k)}\\ &\cdot\left[(\mathcal{M}_i - \mathcal{M})\prod_{k=1,k\neq n}^{K}\times_k\mathbf{U}_k^T\right]_{(k)}^T\end{aligned} \tag{7}$$

$$S_w^k = \sum_{i=1}^{C}\sum_{j=1}^{n_j}\left[(\mathcal{X}_{i,j} - \mathcal{M}_i)\prod_{k=1,k\neq n}^{K}\times_k\mathbf{U}_k^T\right]_{(k)}$$

$$\cdot\left[(\mathcal{X}_{i,j} - \mathcal{M}_i)\prod_{k=1,k\neq n}^{K}\times_k\mathbf{U}_k^T\right]_{(k)}^T \tag{8}$$

## III. DEEP TENSOR FACTORIZATION NETWORK FOR FAULT DIAGNOSIS

In the industrial process, the measurements are often mutually correlated not only in the temporal domain but also in the spatial domain, since the mass/energy balances and a low sampling rate of the hardware sensor usually arise. It is difficult to extract the spatially latent local structures and local correlations in the dense connection. Therefore, this paper incorporates the concept of tensor factorization (TF) into FC layers of DNN and proposes a TF-DNN scheme to solve this problem. Moreover, the model works in an end-to-end manner, supporting the feature extraction and the classification simultaneously.

### A. MERGING TF AND DEEP NETWORK

Consider a sequence classification task for a desired output $\mathbf{y}_t \in \mathbb{R}^C$ given a sequential 2D tensor input $X_n = \{\mathbf{x}_t \in \mathbb{R}^D | \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T\} \in \mathbb{R}^{D\times T}, n = 1,\cdots,N$ belonging to $C$ classes at each time-step $t \in \{1,\cdots,T\}$, $l$-st FC layers applying the previous tensor factorization stage to the input time series can be seen as follows,

$$\mathbf{W}^l = \mathbf{U}_1^l\mathbf{H}^{l-1} \tag{9}$$

$$\mathbf{H}^l = \mathbf{W}^l\mathbf{U}_2^l + \mathbf{B} \tag{10}$$

$$\mathbf{A}^l = f\left(\mathbf{H}^l\right) \tag{11}$$

$$\mathbf{Z} = \underbrace{\left(\mathbf{U}_1^l\cdots\mathbf{U}_1^1\right)}_{\mathbf{U}_1}\times X_n \times \underbrace{\left(\mathbf{U}_2^l\cdots\mathbf{U}_2^1\right)}_{\mathbf{U}_2} \tag{12}$$

where $\mathbf{Z}$ is the output layer, $\mathbf{U}_1^l \in \mathbb{R}^{P\times D_l}$ and $\mathbf{U}_2^l \in \mathbb{R}^{T\times C}$ are the projection matrix obtained from Equation (6), $D_l$ is the size of the previous hidden state (for input layer, $D_l = D$), $\mathbf{B} \in \mathbb{R}^{P\times C}$ is the bias, $\mathbf{A}^l$ and $\mathbf{H}^l$ are respectively the activations of the neurons and hidden states in layer $l$, and $f(\cdot)$ corresponds to the nonlinear transformation function, such as ReLU [21]. Note that $P$ is the tensor size and the tensorization operation can be a stack structure in this work. The softmax layer is shown as follows,

$$\tilde{\mathbf{y}} = \frac{\exp\left(\mathbf{Z}\right)}{\sum_{j=1}^{C}\exp\left(\mathbf{Z}_j\right)} \tag{13}$$

where $\mathbf{Z}_j$ is the row vector of the output layer.

Following the regular back-propagation procedure, the gradient can be calculated to update the parameters as follow,

$$\frac{\mathrm{d}\mathbf{W}^l}{\mathrm{d}\mathbf{U}_1^l} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{U}_1^l}\left[\mathbf{U}_1^l\mathbf{H}^{l-1}\right] = \mathbf{H}^{l-1} \tag{14}$$

$$\frac{\mathrm{d}\mathbf{W}^l}{\mathrm{d}\mathbf{H}^{l-1}} = \frac{\mathrm{d}}{\mathrm{d}\left(\mathbf{H}^{l-1}\right)}\left[\mathbf{U}_1^l\mathbf{H}^{l-1}\right] = \mathbf{U}_1^l \tag{15}$$

$$\frac{\mathrm{d}\mathbf{A}^l}{\mathrm{d}\mathbf{W}^l} = \frac{\mathrm{d}\mathbf{A}^l}{\mathrm{d}\mathbf{H}^l}\frac{\mathrm{d}\mathbf{H}^l}{\mathrm{d}\mathbf{W}^l} = \frac{\mathrm{d}\mathbf{A}^l}{\mathrm{d}\mathbf{H}^l}\cdot\mathbf{U}_2^l \tag{16}$$

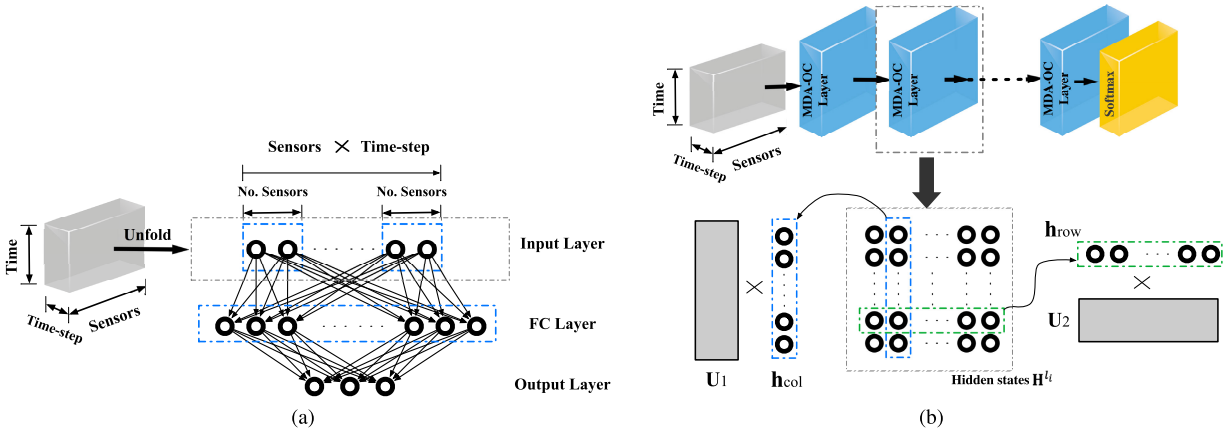



**FIGURE 1.** Illustration of original (a) FC layer and (b) tensor factorization layer, where $\mathbf{h}_{\text{col}}$ and $\mathbf{h}_{\text{row}}$ are the column vector and the row vector of hidden states, respectively. $\mathbf{U}_1$ and $\mathbf{U}_2$ are the projection matrices in layer $l$.

$$\frac{\mathrm{d}\mathbf{A}^l}{\mathrm{d}\mathbf{U}_2^l} = \frac{\mathrm{d}\mathbf{A}^l}{\mathrm{d}\mathbf{H}^l}\frac{\mathrm{d}\mathbf{H}^l}{\mathrm{d}\mathbf{U}_2^l} = \frac{\mathrm{d}\mathbf{A}^l}{\mathrm{d}\mathbf{H}^l}\cdot\mathbf{W}^l \tag{17}$$

where $\frac{\mathrm{d}\mathbf{A}^l}{\mathrm{d}\mathbf{H}^l}$ is the derivative of activation function. Let $c_1, c_2, \cdots, c_C$ be $C$ classes of fault, categorical cross-entropy loss used in our sequence classification task has the form,

$$L(\mathbf{U}_1, \mathbf{U}_2) = -\frac{1}{T-2}\sum_{t=1}^{T-2}\sum_{i=1}^{C}\left[\mathbf{y}_{t+2,i}\ln\tilde{\mathbf{y}}_{t+2,i}\right] \tag{18}$$

where $\tilde{\mathbf{y}}_{t+2,i} \in (0,1) : \sum_i \tilde{\mathbf{y}}_{t+2,i} = 1, \forall i, t$ is the output probability that sample $t$ belongs to class $c_i$.

The architecture of the original FC layer and the proposed TF-DNN scheme are shown in Fig. 1. In the traditional FC layer, the input layer is usually vectorized, as shown in Fig. 1(a). Different with FC layer, the input layer is directly represented by a second-order tensor, which can be seen in Fig. 1(b). Through incorporating the tensor factorization into the FC layers, the network can be evidently widened due to the interaction of the tensorized parameters. Meanwhile, the parameter number in the hidden state actually is reduced by the orthogonal constraints defined in Equation (5).

From Equation (1), it is important to note that the dependency between each mode-$k$ allows knowledge sharing across feature vectors in the dense connection of the deep network. For ease of exposition, we denote each column and row of input layer $\mathbf{H}^0 = X_n \in \mathbb{R}^{D\times T}$ as $\mathbf{h}_{\text{col}} \in \mathbb{R}^D$ and $\mathbf{h}_{\text{row}} \in \mathbb{R}^T$, respectively. From Equation (9), $\mathbf{W}^l$ can be rewritten as,

$$\mathbf{W} = \mathbf{U}_1 X_n = \mathbf{U}_1\mathbf{h}_{\text{col}} \tag{19}$$

Since $\mathbf{h}_{\text{col}}$ is already transformed by MDA with the orthogonal constraints, the information across the channels can be captured by the projection $\mathbf{U}_1$. Similarly, based on Equation (10), $\mathbf{H}^l$ can be rewritten as,

$$\mathbf{H} = \mathbf{h}_{\text{col}}^T\mathbf{U}_2 + \mathbf{B} \tag{20}$$

Equation (20) shows the temporal progress shared with the channels can be learned by the projection $\mathbf{U}_2$.

The essential advantage of TF layer is that parameters are shared across different the temporal evolution of the features through time. In industrial process fault diagnosis, this provides an effective way to extract fault features in both the spatial and the temporal domains. Most of the fault features are dependent on the temporal information, especially in the chemical process.

### B. CONVERGENCE ANALYSIS

For the proposed method, it is crucial to conduct the investigation on the convergence analysis with respect to the parameters. The proposed deep network can be trained by minimizing the risk $L(\mathbf{U}_1, \mathbf{U}_2)$ defined in Equation (18). Moreover, the softmax function is defined in Equation (13). During the backward pass, the element-wise gradient with the loss function using chain rules is expressed as,

$$\frac{\mathrm{d}L(\mathbf{U}_1, \mathbf{U}_2)}{\mathrm{d}\mathbf{U}_1} = -\frac{\mathrm{d}L}{\mathrm{d}\tilde{\mathbf{y}}}\frac{\mathrm{d}\tilde{\mathbf{y}}}{\mathrm{d}\mathbf{Z}}\frac{\mathrm{d}\mathbf{Z}}{\mathrm{d}\mathbf{U}_1} \tag{21}$$

$$\frac{\mathrm{d}L(\mathbf{U}_1, \mathbf{U}_2)}{\mathrm{d}\mathbf{U}_2} = -\frac{\mathrm{d}L}{\mathrm{d}\tilde{\mathbf{y}}}\frac{\mathrm{d}\tilde{\mathbf{y}}}{\mathrm{d}\mathbf{Z}}\frac{\mathrm{d}\mathbf{Z}}{\mathrm{d}\mathbf{U}_2} \tag{22}$$

Assuming that there exists a bounded open set $\mathbf{U}_k \in \mathbb{R}^{I_k\times P_k}$ such that $\{\mathbf{U}_1, \mathbf{U}_2\} \subset \mathbf{U}_k$ and

$$\sum_{i=1}^{C}\left\|\frac{\mathrm{d}L(\mathbf{U}_1, \mathbf{U}_2)}{\mathrm{d}\mathbf{U}_1}\right\| \le g \tag{23}$$

$$\sum_{i=1}^{C}\left\|\frac{\mathrm{d}L(\mathbf{U}_1, \mathbf{U}_2)}{\mathrm{d}\mathbf{U}_2}\right\| \le g \tag{24}$$

where $g$ is a function defined on a compact set.

*Theorem 1: Given the modeling risk dynamics defined by Equations (21) and (22), if the orthogonal constraints defined by Equation (5) are met, then the modeling risk is uniformly ultimately bounded.*

*Proof:* From Equation (21) and according to the chain rule, we have

$$\frac{\mathrm{d}L\left(\mathbf{U}_1,\mathbf{U}_2\right)}{\mathrm{d}\mathbf{U}_1}=-\frac{1}{T-2}\sum_{m=1}^{M}\left[\mathbf{y}_m\frac{1}{\tilde{\mathbf{y}}_m}\right]\cdot\mathbf{U}_2\cdot\frac{\mathrm{d}\tilde{\mathbf{y}}}{\mathrm{d}\mathbf{Z}} \tag{25}$$

where

$$\frac{\mathrm{d}\tilde{\mathbf{y}}}{\mathrm{d}\mathbf{Z}}=\begin{cases}\tilde{\mathbf{y}}_i\left(1-\tilde{\mathbf{y}}_i\right), & \text{for} \quad i=m\\ -\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_m, & \text{for} \quad i\neq m\end{cases} \tag{26}$$

and $M$ is the total number of row vector in the layer $l$. The gradient of the loss function is given by

$$\begin{aligned}\frac{\mathrm{d}L\left(\mathbf{U}_1,\mathbf{U}_2\right)}{\mathrm{d}\mathbf{U}_1}&=\frac{1}{T-2}\sum_{m\neq i}\left[\frac{\mathbf{y}_m}{\tilde{\mathbf{y}}_m}\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_m-\frac{\mathbf{y}_i}{\tilde{\mathbf{y}}_i}\tilde{\mathbf{y}}_i\left(1-\tilde{\mathbf{y}}_i\right)\right]\\&\quad\times\mathbf{U}_2\\&=\frac{1}{T-2}\sum_{m\neq i}\left[\mathbf{y}_m\tilde{\mathbf{y}}_i+\mathbf{y}_i\tilde{\mathbf{y}}_i-\mathbf{y}_i\right]\times\mathbf{U}_2\\&=\frac{1}{T-2}\left[\tilde{\mathbf{y}}_i\left(\sum_m\mathbf{y}_m\right)-\mathbf{y}_i\right]\times\mathbf{U}_2\end{aligned} \tag{27}$$

The derivation for the gradient of the loss function with respect to $\mathbf{U}_2$ is straightforward, which gives

$$\frac{\mathrm{d}L\left(\mathbf{U}_1,\mathbf{U}_2\right)}{\mathrm{d}\mathbf{U}_2}=\frac{1}{T-2}\mathbf{U}_1\times\left[\tilde{\mathbf{y}}_i\left(\sum_m\mathbf{y}_m\right)-\mathbf{y}_i\right] \tag{28}$$

According to $\sum_i\tilde{\mathbf{y}}_i=1$, Equations (27) and (28), we have,

$$\frac{\mathrm{d}L\left(\mathbf{U}_1,\mathbf{U}_2\right)}{\mathrm{d}\mathbf{U}_1}=\frac{1}{T-2}\left[\tilde{\mathbf{y}}_i-\mathbf{y}_i\right]\times\mathbf{U}_2 \tag{29}$$

$$\frac{\mathrm{d}L\left(\mathbf{U}_1,\mathbf{U}_2\right)}{\mathrm{d}\mathbf{U}_2}=\frac{1}{T-2}\mathbf{U}_1\times\left[\tilde{\mathbf{y}}_i-\mathbf{y}_i\right] \tag{30}$$

If the orthogonal constraints on the projections $\mathbf{U}_1$ and $\mathbf{U}_2$ are met, the following inequalities are hold,

$$\begin{aligned}\sum_{i=1}^{C}\left\|\frac{\mathrm{d}L\left(\mathbf{U}_1,\mathbf{U}_2\right)}{\mathrm{d}\mathbf{U}_1}\right\|&=\frac{1}{T-2}\sum_{i=1}^{C}\left\|\left[\tilde{\mathbf{y}}_i-\mathbf{y}_i\right]\times\mathbf{U}_2\right\|\\&\leq\frac{1}{T-2}\sum_{i=1}^{C}\left\|\tilde{\mathbf{y}}_i-\mathbf{y}_i\right\|\end{aligned} \tag{31}$$

$$\begin{aligned}\sum_{i=1}^{C}\left\|\frac{\mathrm{d}L\left(\mathbf{U}_1,\mathbf{U}_2\right)}{\mathrm{d}\mathbf{U}_2}\right\|&=\frac{1}{T-2}\sum_{i=1}^{C}\left\|\mathbf{U}_2\times\left[\tilde{\mathbf{y}}_i-\mathbf{y}_i\right]\right\|\\&\leq\frac{1}{T-2}\sum_{i=1}^{C}\left\|\tilde{\mathbf{y}}_i-\mathbf{y}_i\right\|\end{aligned} \tag{32}$$

□

### C. FAULT DIAGNOSIS BASED ON TF-DNN

With the proposed TF-DNN, the diagnosis of fault data is straightforward. The off-line modeling and online monitoring procedure are summarized below.

- *Offline modeling:*
  1) Collect process data and labels as training data.
  2) Preprocess the training data with missing measurement imputations, outliers removing and z-score normalization.
  3) TF-DNN network is designed for the training data. (See Fig. 1 (b)).
  4) Train TF-DNN network with ReLU activation, dropout, batch-normalization and Adam.
  5) Compute the categorical cross-entropy loss with Equation (18).
  6) Output the parameters for TF-DNN network.
- *Online modeling:*
  1) Sample a new augmented test data$\{\mathbf{x}_{t-2}^{\text{new}},\mathbf{x}_{t-1}^{\text{new}},\mathbf{x}_{t}^{\text{new}}\}$, $t\geq 3$.
  2) Preprocess the test data according to the mapping of the training features.
  3) Feed test data to the TF-DNN network.

**TABLE 1. Process faults in the TE process.**

| IDVs | Description | Type |
|---|---|---|
| IDV(1) | A/C feed ratio, B composition constant | Step |
| IDV(2) | B composition, A/C ratio constant | Step |
| IDV(3) | D feed temperature | Step |
| IDV(4) | Reactor cooling water inlet temperature | Step |
| IDV(5) | Condenser cooling water inlet temperature | Step |
| IDV(6) | A feed loss | Step |
| IDV(7) | C header pressure loss reduced | Step |
| IDV(8) | A, B, C feed composition | Random variation |
| IDV(9) | D feed temperature | Random variation |
| IDV(10) | C feed temperature | Random variation |
| IDV(11) | Reactor cooling water inlet temperature | Random variation |
| IDV(12) | Condenser cooling water inlet temperature | Random variation |
| IDV(13) | Reaction kinetics | Slow drift |
| IDV(14) | Reactor cooling water valve | Sticking |
| IDV(15) | Condenser cooling water valve | Sticking |
| IDV(16) - (20) | Unknown | Unknown |
| IDV(21) | Valve (Stream 4) | Constant position |

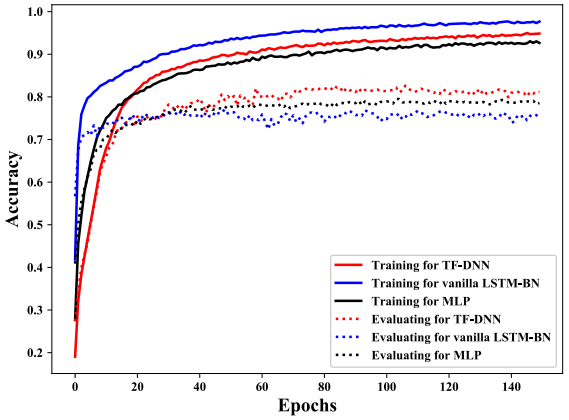



**FIGURE 2. The overall accuracy curves of TF-DNN, vanilla LSTM-BN and MLP on 21 faults during training stage.**

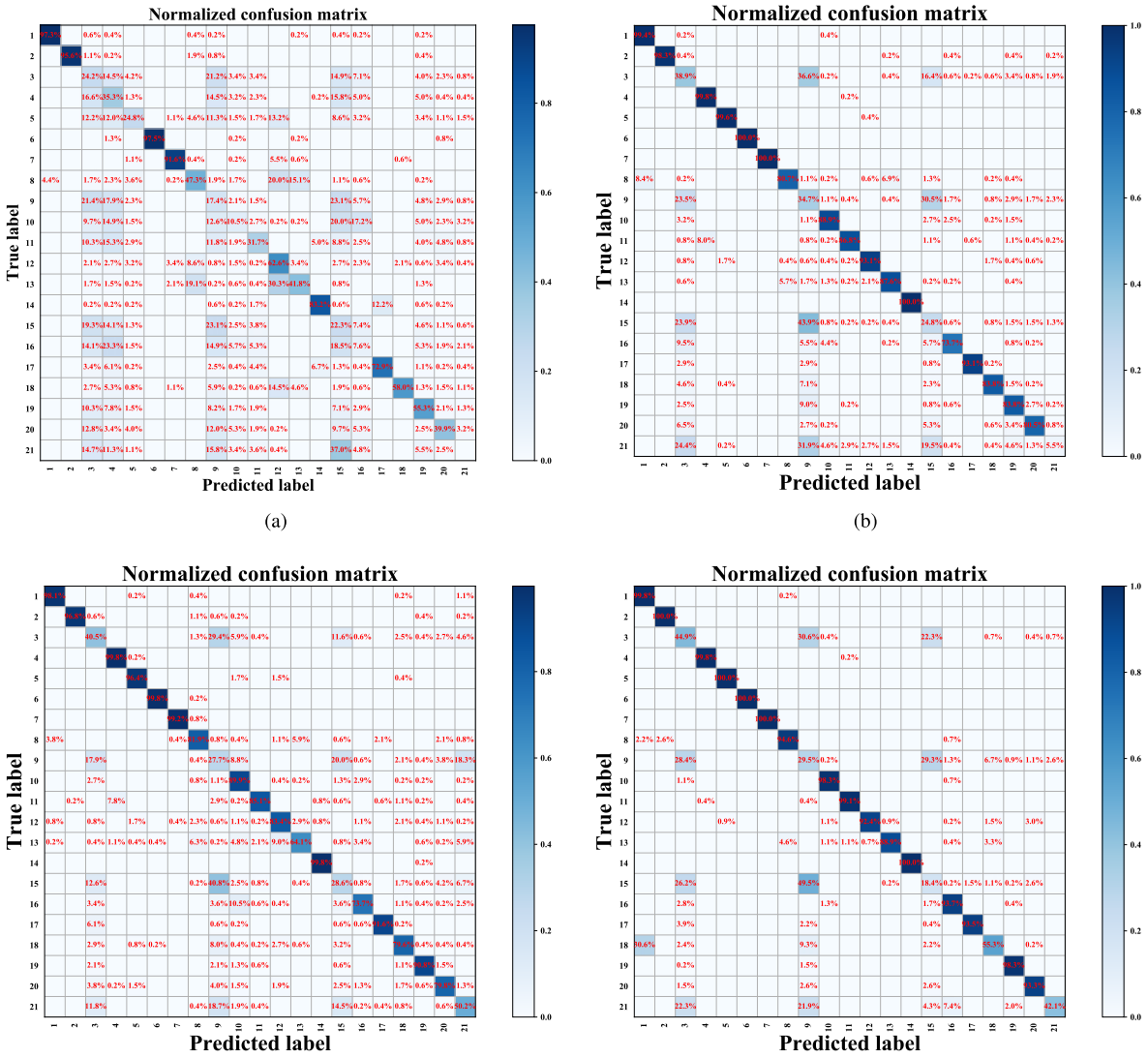



**FIGURE 3.** Confusion matrices of different methods over 21 fault modes. (a) PCA-SVM. (b) MLP. (c) Vanilla LSTM-BN. (d) TF-DNN.

4) Acquire the model output $\mathbf{y}_t^{\text{new}}$.
5) Classify test data to fault $\hat{c}_i = \arg\max\{(\mathbf{y}_t^{\text{new}})^{(i)}\}$.

## IV. APPLICATION STUDY TO TENNESSEE EASTMAN BENCHMARK PROCESS

In this section, a series of experiments were conducted to evaluate the performance of the proposed TF-DNN, vanilla LSTM with batch normalization (vanilla LSTM-BN), multilayer perceptron (MLP), and DPCA [22]-SVM in the fault detection and classification of the multivariate Tennessee Eastman (TE) sequential data. A brief description was firstly provided to the benchmark TE process, which is a benchmark process for various studies including process modeling and monitoring. Experiments were run on a computer with Inter Core i5-5257 2.7GHz CPU, 8GB memory and macOS High Sierra operating system. Programming language is Python 3.7 with deep learning package "Tensorflow".

### A. PROCESS DESCRIPTION

As a source of publicly available data, the TE process consisted of five major process units, which were a reboiled striper, a cooling condenser, a flash separator, an exothermic two-phase reactor and a recycle compressor. In this process, there were totally 52 measurements available (41 measurements for process variables and 11 measurements for manipulated variables), and a set of 21 programmed fault modes (namely, IDV(1)-IDV(21)) were defined in [23], as listed in Table 1. The simulation data of the TE process used in our study was downloaded from https://github.com/camaramm/tennessee-eastman-profBraatz. Each data set consisted of 500 samples, which contains a simulation run of 25 hours with a sampling interval of 3 minutes. In the case of faulty operation, each test data set for one fault mode (introduced at 160th sample) consisted of 960 samples. All variables were used for

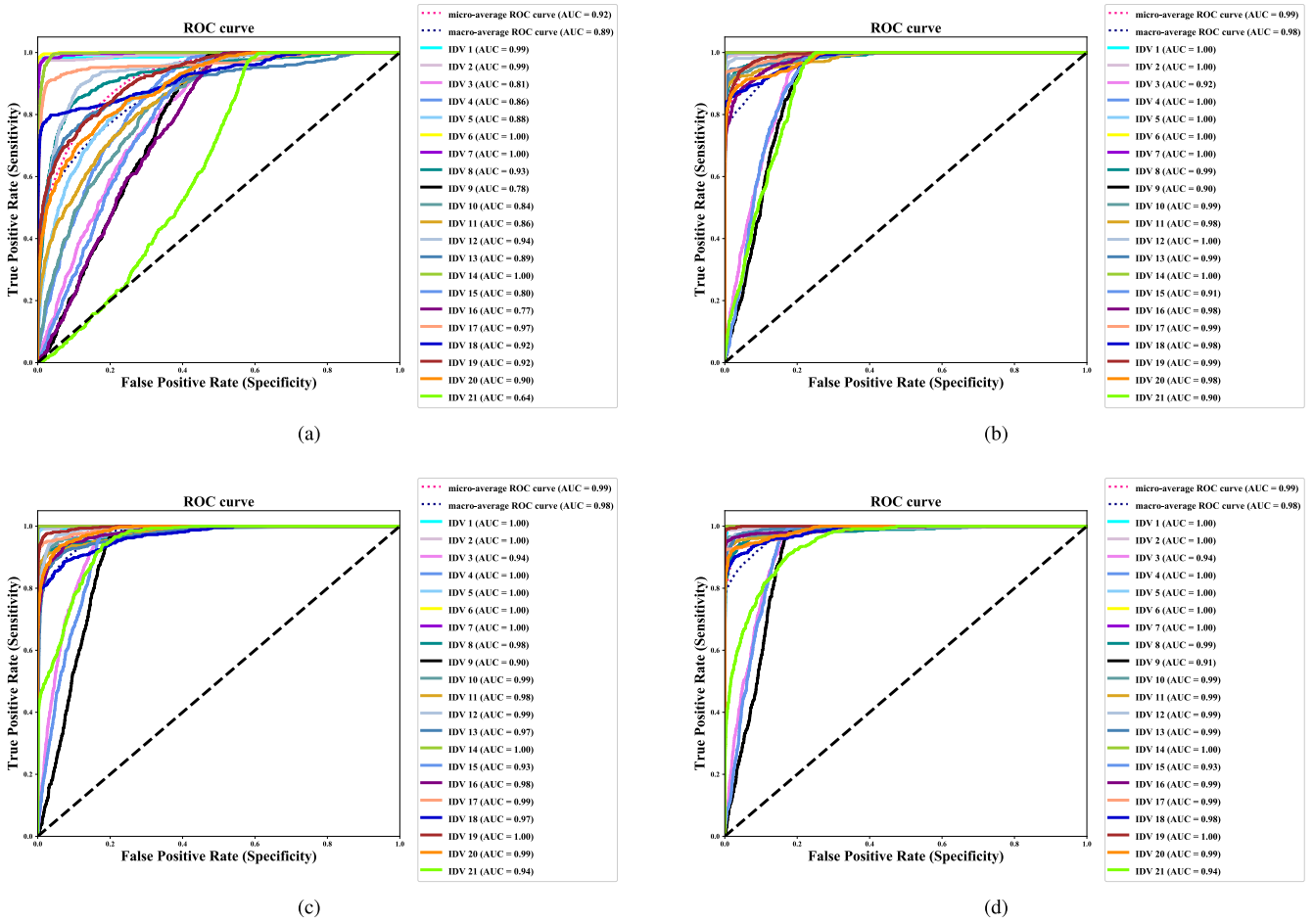



**FIGURE 4.** ROC curves of different methods over 21 fault modes. (a) PCA-SVM; (b) MLP; (c) Vanilla LSTM-BN; (d) TF-DNN. The micro and macro-averaging ROC curves are denoted by dashed pink and navy blue lines, respectively.

monitoring. The values of both the training samples and the testing samples were normalized to have zero mean and unit variance.

### B. PERFORMANCE METRICS

We investigated the multi-class classification performance using a total of 21 fault modes which involve all of the compositions, manipulated and measurement variables in the TE process. For an individual class IDV($i$), the performance was typically evaluated by a confusion matrix which consists of true positives ($\mathrm{TP}_i$), false positives ($\mathrm{FP}_i$), true negatives ($\mathrm{TN}_i$) and false negatives ($\mathrm{FN}_i$). The detail symbol representation is explained in Table 2. A set of changes in a confusion matrix should be considered to specific characteristics of data, i.e., recall, F1-score and precision.

For the multi-class setting, quality of the overall classification is usually averaged across all the classes in many possible ways. In this paper, we assessed the overall performance in the following two ways:

- *Micro-averaging:* calculate metrics globally by counting the total number of cumulative $\mathrm{TP}_i$, $\mathrm{FP}_i$, $\mathrm{TN}_i$, $\mathrm{FN}_i$.

**TABLE 2.** Confusion matrix for each class IDV($i$).

| | Counts of predicted label $i$ | Counts of predicted label other than $i$ |
|---|---|---|
| Counts of real label $i$ | $\mathrm{TP}_i$ | $\mathrm{TN}_i$ |
| Counts of real label other than $i$ | $\mathrm{FP}_i$ | $\mathrm{FN}_i$ |

The micro-averaged precision $P_{\mathrm{Micro}}$, recall $R_{\mathrm{Micro}}$ and F1-score $F1_{\mathrm{Micro}}$ are defined as,

$$P_{\mathrm{Micro}} = \frac{\sum_{i=1}^{l} \mathrm{TP}_i}{\sum_{i=1}^{l} (\mathrm{TP}_i + \mathrm{FP}_i)} \tag{33}$$

$$R_{\mathrm{Micro}} = \frac{\sum_{i=1}^{l} \mathrm{TP}_i}{\sum_{i=1}^{l} (\mathrm{TP}_i + \mathrm{FN}_i)} \tag{34}$$

$$F1_{\mathrm{Micro}} = \frac{2\, P_{\mathrm{Micro}} \cdot R_{\mathrm{Micro}}}{P_{\mathrm{Micro}} + R_{\mathrm{Micro}}} \tag{35}$$

- *Macro-averaging:* calculate metrics for each class independently, and then find their average value.

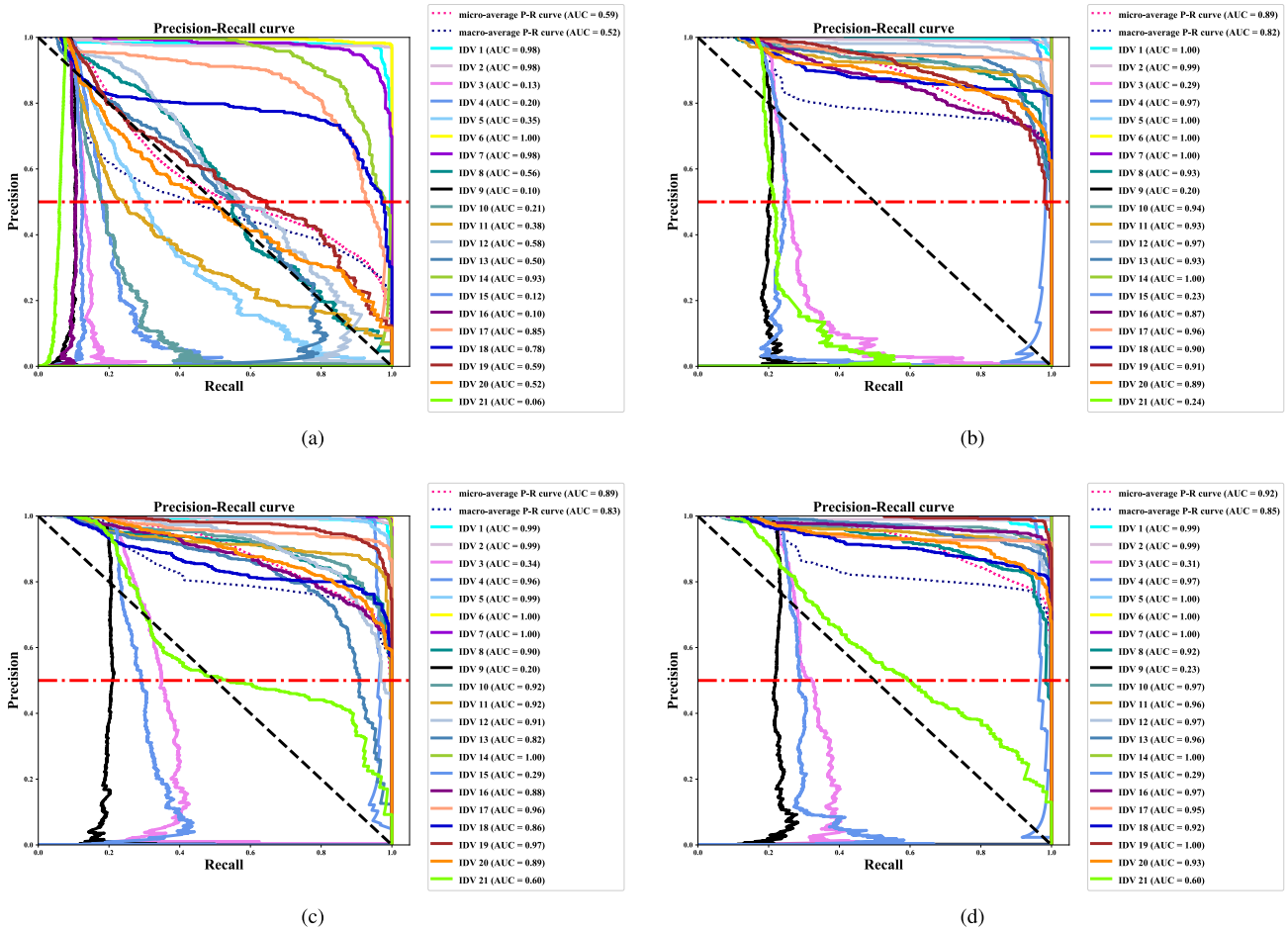



**FIGURE 5.** Precision-recall curves of different methods over 21 fault modes. (a) PCA-SVM; (b) MLP; (c) Vanilla LSTM-BN; (d) TF-DNN. The micro and macro-averaging precision-recall curves are denoted by dashed pink and navy blue lines, respectively. The horizontal line (red with dashes) represents the random performance level.

The macro-averaged precision $P_{\text{Macro}}$, recall $R_{\text{Macro}}$ and F1-score $F1_{\text{Macro}}$ are defined as,

$$P_{\text{Macro}} = \frac{1}{l}\sum_{i=1}^{l}\frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{36}$$

$$R_{\text{Macro}} = \frac{1}{l}\sum_{i=1}^{l}\frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{37}$$

$$F1_{\text{Macro}} = \frac{2\,P_{\text{Macro}} \cdot R_{\text{Macro}}}{P_{\text{Macro}} + R_{\text{Macro}}} \tag{38}$$

### C. FAULT CLASSIFICATION AND ANALYSIS

We compared the fault detection performance of the proposed TF-DNN method with that of classical shallow network support vector machine (SVM) and classical dynamic deep networks including MLP, vanilla LSTM-BN methods. For SVM, the classifier with RBF kernel was used for fault diagnosis after the feature extraction of PCA. The reduced dimension of PCA was selected to capture 97% of the variance within training set, and the parameter in the kernel function was chosen as the reciprocal of the number of principle components. In the following experiments, we adopted the feed forward neural network for MLP with 3 hidden layer and directly trained this network using rectified linear unit (ReLU) as the activation function. For fair comparison, the number of the hidden layer nodes in vanilla LSTM-BN and TF-DNN was set to 15, which is recommended by [14]. The parameters in these three deep networks were optimized by the Adam algorithm [24] with the initial learning rate of 0.001, where the size of a mini-batch is 128. In order to avoid overfitting, dropout strategy with a rate of 0.3 was used for each hidden layer of MLP and for the softmax layer of the vanilla LSTM-BN and TF-DNN.

Fig. 2 shows the overall accuracy curve for the training set and test set of TF-DNN, vanilla LSTM-BN and MLP during 150 training epochs. Experimental results in Fig. 2 show that there is no overfitting during training, and accuracy of TF-DNN is close to the value of 1 in the training set. At the same time, TF-DNN model has the higher accuracy in test set. It means that TF-DNN model has the more powerful feature representation abilities to improve the generalization of deep model.

For a more detailed analysis, we investigated the normalized confusion matrices of PCA-SVM, MLP, vanilla LSTM-BN and the proposed TF-DNN method in Fig. 3.

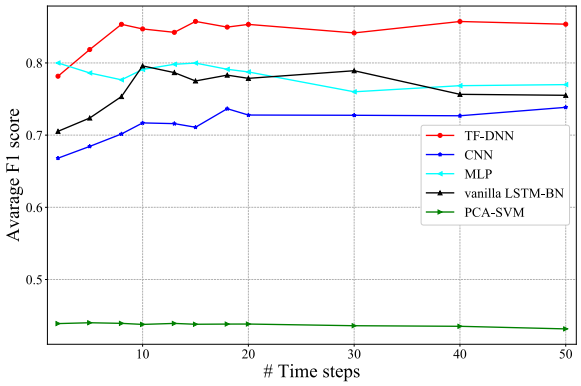



**FIGURE 6.** Average F1 scores of PCA-SVM, MLP, Vanilla LSTM- BN, CNN and TF-DNN with different time-steps.

It is a straightforward visualization that provides an intuitive summary of the relations between the ground truth and the classifier outputs from the adopted method. In the matrix, a row represents an instance of the predicted class, whereas a column represents an instance of the actual class. Consequently, the diagonal elements stand for the correct classification decisions. From Fig. 3, the deep learning methods MLP, vanilla LSTM-BN, and TF-DNN have obvious preponderance compared with traditional classification methods PCA-SVM. Clearly, TF-DNN provides a higher classification rates than the other two deep learning methods, MLP and vanilla LSTM-BN, for the most of IDVs. This phenomenon is due to the fact that promoting orthogonality in the layers of deep network improves computational efficiency in optimization, meanwhile enables efficient knowledge sharing across the spatiotemporal features in each fault class.

Moreover, the receiver operating characteristic (ROC) was utilized to evaluate the performance of a fault diagnosis method on each fault mode. In practice, ROC curve succinctly visualizes the relationship between true alarm rates ($TP_i$ ratio) and false alarm rates ($FP_i$ ratio). Informally, a point in the top left corner of ROC box is not only with a higher probability of correctly recognizing the faults, but also with a lower probability of incorrectly recognizing the normal operating condition (NOC) data. The ROC curves of PCA-SVM, MLP, vanilla LSTM-BN and TF-DNN methods over 21 fault modes are shown in Fig. 4. For the most of IDVs, deep learning methods greatly improve the micro- and macro-average area under curve (AUC). This demonstrates that the dynamic information can be effectively by through the hidden layers of deep networks. From Fig. 4(b) - 4(d), we can see that the AUC score of the ROCs provide similar performances for all fault modes and that the AUC score of the TF-DNN ROCs provides improvement in monitoring performance over IDVs (9), (11), (15) and (21).

To further evaluate the performance, precision-recall curve (P-R) was used for evaluating the classification performance. In the fault diagnosis scenario, precision (the y-axis of the curve) provides the information on the correct classification

**TABLE 3.** F1 scores using MLP, vanilla LSTM-BN, CNN and TF-DNN for the TE process.

| IDVs | MLP | vanilla LSTM-BN | CNN | TF-DNN |
|---|---|---|---|---|
| IDV(1) | 0.942 | 0.9689 | **0.9781** | 0.9570 |
| IDV(2) | 0.9936 | 0.9947 | 0.9860 | **0.9957** |
| IDV(3) | 0.304 | 0.3158 | 0.2218 | **0.4147** |
| IDV(4) | 0.9669 | 0.9655 | 0.9761 | **0.9853** |
| IDV(5) | 0.9812 | 0.9128 | 0.9503 | **0.9926** |
| IDV(6) | 0.9989 | 0.9979 | 0.9978 | **1** |
| IDV(7) | **1** | 0.9979 | 0.9958 | **1** |
| IDV(8) | 0.8874 | 0.7463 | 0.6412 | **0.8879** |
| IDV(9) | 0.2279 | 0.2653 | 0.1320 | **0.3133** |
| IDV(10) | 0.8778 | 0.8835 | 0.3013 | **0.9463** |
| IDV(11) | 0.9511 | 0.9093 | 0.9217 | **0.9603** |
| IDV(12) | **0.9484** | 0.8107 | 0.7631 | 0.9217 |
| IDV(13) | 0.8866 | 0.7237 | 0.7292 | **0.8966** |
| IDV(14) | **1** | 0.9915 | **1** | **1** |
| IDV(15) | 0.2548 | 0.3199 | 0.3176 | **0.3296** |
| IDV(16) | 0.6945 | 0.8049 | 0.2959 | **0.9463** |
| IDV(17) | 0.9605 | 0.94 | 0.9131 | **0.9617** |
| IDV(18) | 0.8562 | 0.8392 | 0.8622 | **0.8857** |
| IDV(19) | 0.8110 | 0.9513 | **0.9872** | 0.9736 |
| IDV(20) | 0.8643 | 0.8711 | 0.7141 | **0.8865** |
| IDV(21) | 0.2042 | 0.5311 | 0.5516 | **0.5931** |
| Macro-avarage | 0.7909 | 0.7959 | 0.7169 | **0.8470** |

of an actual fault, and recall (the x-axis of the curve) provides a proportion of a kind of sample that is correctly assessed. In the P-R curve, the upper right corner represents a classifier that obtains 100% precision and sensitivity. This is the ideal point which can be considered as a perfect fault diagnosis. Hence, we can expect the closer the P-R curve comes to the top right corner, the better the diagnosis is overall. Fig. 5 shows the P-R plots of PCA-SVM, MLP, vanilla LSTM-BN and TF-DNN over 21 fault modes. The red horizontal lines in Fig. 5 are the random performance level of a classifier [25]. It is clearly observed that TF-DNN presents a better quality of the overall performance in the P-R space, which is confirmed by the micro- and macro-average AUC (micro-average P-R curve $\mathrm{AUC}_{\mathrm{TF-DNN}} = 0.92 > \mathrm{AUC}_{\mathrm{vanillaLSTM-BN}} = \mathrm{AUC}_{\mathrm{MLP}} = 0.89 > \mathrm{AUC}_{\mathrm{PCA-SVM}} = 0.59$, macro-average P-R curve $\mathrm{AUC}_{\mathrm{TF-DNN}} = 0.85 > \mathrm{AUC}_{\mathrm{vanillaLSTM-BN}} = 0.83 > \mathrm{AUC}_{\mathrm{MLP}} = 0.82 > \mathrm{AUC}_{\mathrm{PCA-SVM}} = 0.52$). Inspecting the AUC values of each IDV in Fig. 5(b) - 5(d) finds that TF-DNN provides notably higher ranking of actual faults that are difficult to detect, IDVs (10), (11), (13), (16), (18), (19) and (20). Therefore, the TF-DNN method provides obviously the best performances focusing on the faulty cases. Moreover, the F1 scores were investigated to obtain the tradeoff between the precision and the recall, which are shown in Table 3. For that, the macro-average F1 score of TF-DNN is greatly higher than that of MLP and vanilla LSTM-BN, which further corroborates the benefit of the spatiotemporal features for fault diagnosis. In order to evaluate the TF-DNN network in the local representation of sequence, the baseline configurations with different time-steps $t = \{2, 5, 8, 10, 13, 15, 18, 20, 30, 40, 50\}$ was constructed, while each configuration was repeated for 20 times. Fig. 6 summarizes the results of F1 scores with the above 11 configurations in the test dataset. It can be seen that the

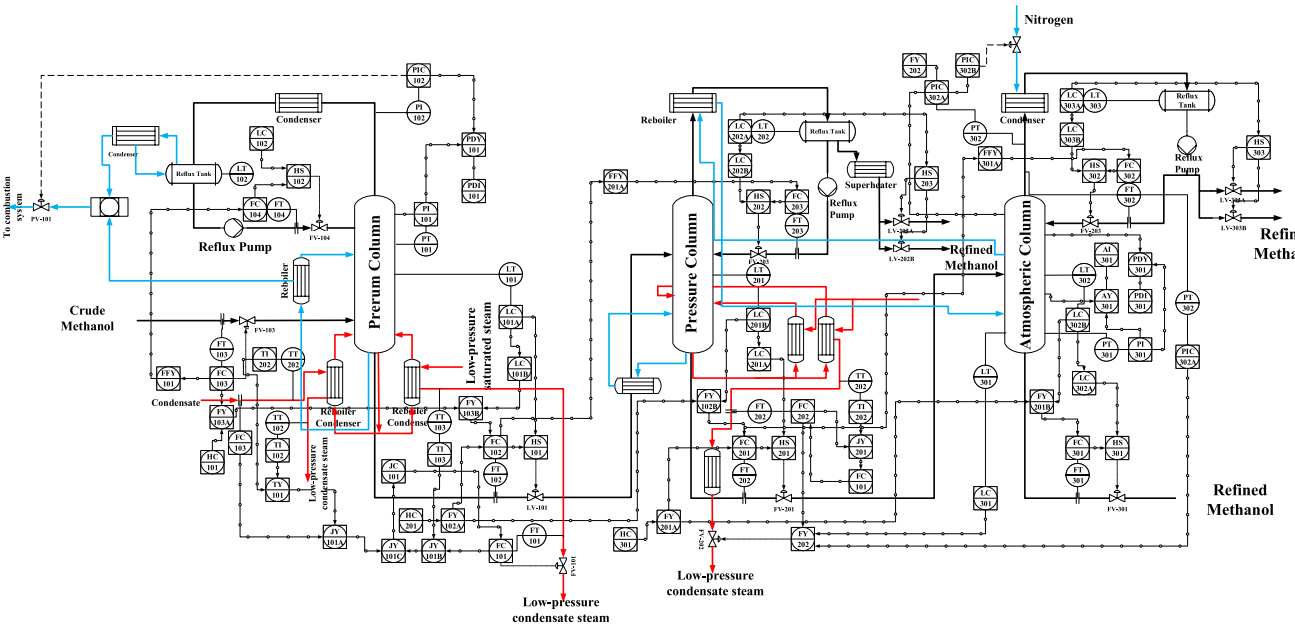



**FIGURE 7.** P&ID diagram of methanol refining unit with a 3 column configuration. The red line and blue line represents temperature and pressure control scheme, respectively. The line with dot is feed control scheme.

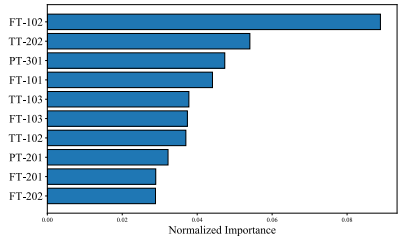



**FIGURE 8.** The sorting features according to the cumulative importance in the methanol plant.

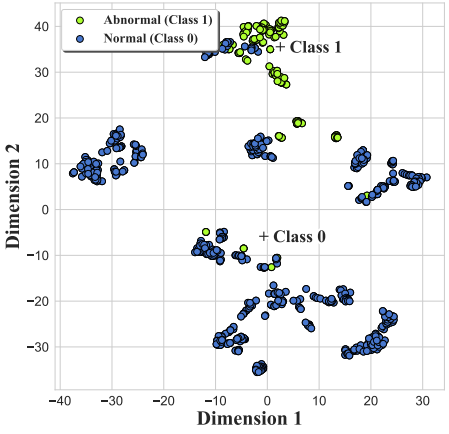



**FIGURE 9.** T-SNE visualization of normal and abnormal operating conditions during training stage. Blue and green circles indicate the normal and abnormal operating points, respectively. Symbols '+' indicates the median point in the corresponding category.

average F1 score of TF-DNN is around 90% with the different time-steps, which means that it is not sensitive for the time-steps.

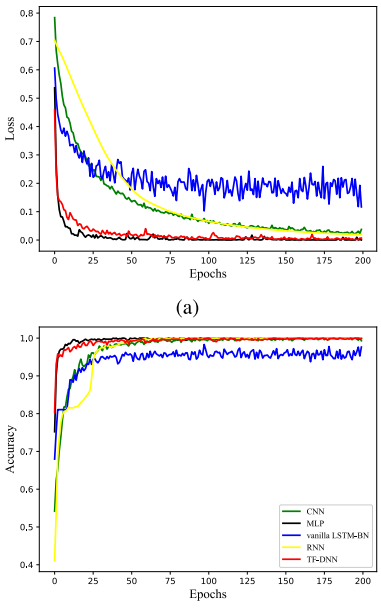



**FIGURE 10.** The overall loss and accuracy of TF-DNN, vanilla LSTM-BN and MLP on 21 faults during training stage.

## V. APPLICATION STUDY TO INDUSTRIAL METHANOL DISTILLATION

### A. PROCESS DESCRIPTION

The application of this work is to detect process upsets in order to protect parts of an industrial methanol plant owned by CNPC. As one of the major feedstock for the olefins production, the methanol section consists of a standard arrangement of three distillation columns. A process diagram of the industrial methanol distillation is shown in Fig. 7. The

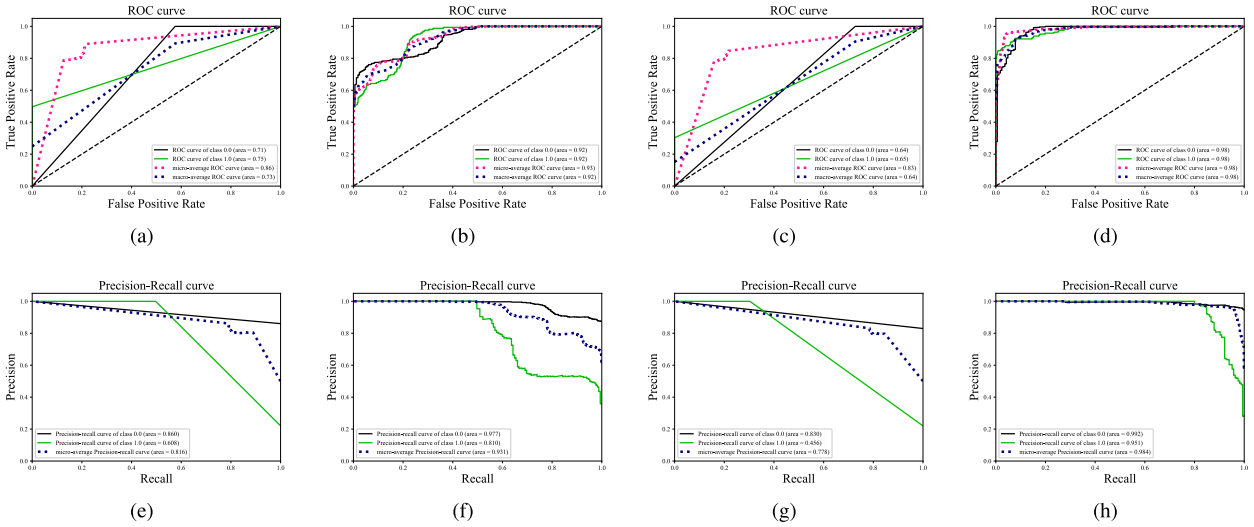



**FIGURE 11.** ROC and precision-recall curves of different methods on the methanol distillation process. (a) ROC curves, CNN; (b) ROC curves, Vanilla LSTM-BN; (c) ROC curves, MLP; (d) ROC curves, TF-DNN; (e) precision-recall curves, CNN; (f) precision-recall curves, Vanilla LSTM-BN; (g) precision-recall curves, MLP; (h) precision-recall curves, TF-DNN. The micro and macro-averaging curves are denoted by dashed pink and navy blue lines, respectively.

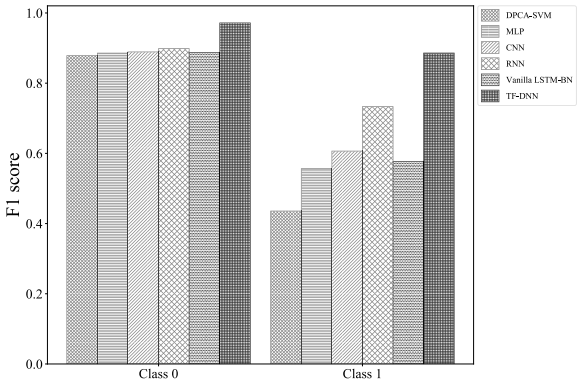



**FIGURE 12.** F1 scores of DPCA-SVM, MLP, CNN, RNN, Vanilla LSTM-BN and TF-DNN for normal and abnormal conditions in the methanol plant.

crude methanol is mainly produced using natural gas (CH4) as feedstock. It is then pumped into a topping column before feeding into refining column. The primary objective of the refining column is to make a given specification methanol while maintaining a high level recovery. In addition, an extra recovery column helps to further recover product methanol and achieves a split between fusel oil and methanol.

In the pressure control scheme, the blue lines in Fig. 7, the dissolved gases and other light substances are removed from the crude methanol by the valve PV-101. In order to monitor the bottoms methanol concentration, the pressure (PT-101, PT-201 and PT-301) and temperature (TT-101, TT-201 and TT-301) on the bottom streams are transmitted to the blocks (AY-101, AY201 and AY-301). To reduce the effects of the disturbances on the flow of condensate, the controller JC-101 and FC-101 are cascaded to maintain the temperature of the toping column, the red lines in Fig. 7. Similarly, the flow of condensate in the reboiler is controlled by the cascade scheme (JC-201 and FC-202). To balance the changes in the methanol content, feeding system is designed as an impulse control structure, which consists of inlet flow transmitter (FT-103, FT-102 and FT-201), level transmitter (LT-101, LT-201 and LT-302), adder block (FY-103B, FY-102B and FY-201B) and flow controller (FC-103, FC-102 and FC-201). The outlet flow transmitter (FT-102, FT-201 and FT-301), adder block (FY-103B, FY-102B and FY-201B) and flow controller (FC-102, FC-201 and FC-301) are composed of a cascade averaging control structure.

### B. DATA ACQUISITION AND PREPROCESSING

To monitor and analyze the operation of the distillation section of a methanol plant, 61 measurable process variables based on engineering requirements were continuously followed for more than 2 years. A total of 728 process samples using for the training datasets were collected from January 2017 to February 2018, while another 760 samples collected from February 2018 to May 2019 were required as the test datasets to demonstrate the effectiveness of the proposed method. From February 2018 to May 2019, there were two different operating conditions causing by the process changing, which was consulted by the experts and engineers.

After missing measurement imputations, outliers removing and z-score normalization, a gradient boosting machine (GBM) implemented by the Light-GBM were utilized to automatically estimate the importance of features from the process. The first 10 most important features on a normalized scale is shown as Fig. 8. In order to observe the distribution of the normal and abnormal conditions, the t-distributed stochastic neighbor embedding (t-SNE) [26] visualization of the methanol plant after feature selection are shown in Fig. 9. The dataset of the methanol plant can be downloaded from https://github.com/luolin1984/DeepLearn_ProcessMonitor/bl ob/master/methanolData021201.mat.

**TABLE 4.** The candidate structures and parameters for MLP, CNN, RNN, vanilla LSTM-BN and TF-DNN.

| Algorithm | Architecture | Optimizer | Learning rate | Number of epochs | Batch size | Activation | Dropout |
|---|---|---|---|---|---|---|---|
| MLP | {10, 64,128, 256,128, 2} | Adam | 0.001 | 100 | 64 | Relu | 0.3 |
| CNN | {10, 32, 64,128, 2} | Adam | 0.001 | 100 | 64 | Relu | 0.25(MaxPooling)<br>0.5(Dense) |
| RNN | {10, 10, 2} | Adam | 0.001 | 100 | 64 | Relu | – |
| vanilla LSTM-BN | {10, 10, 2} | Adam | 0.001 | 100 | 64 | Relu | – |
| TF-DNN | {10, [32, 10], [10, 32], 2}] | Adam | 0.001 | 100 | 64 | Relu | 0.3 |

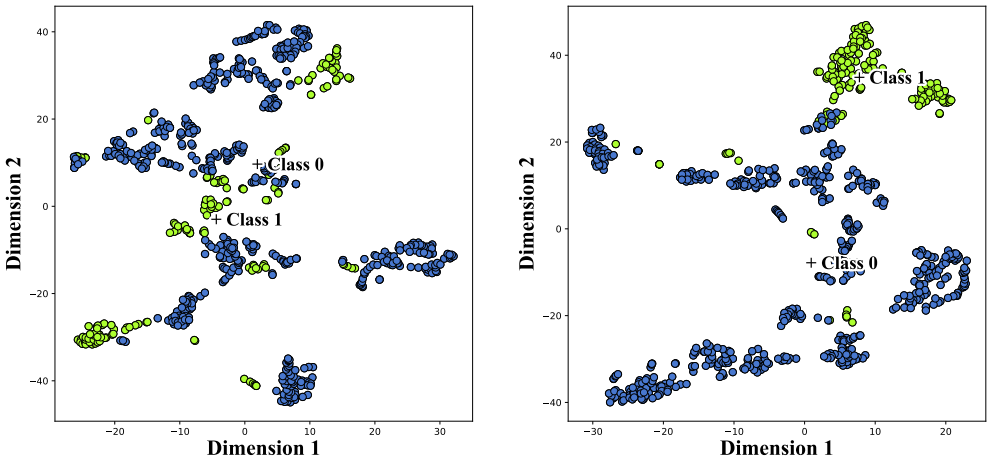



**FIGURE 13.** Feature visualization via t-SNE using the last hidden states of (a) RNN and (b) TF-DNN for the test dataset.

### C. DIAGNOSIS RESULTS COMPARED WITH THE OTHER METHODS

To show the fault classification performance on the industrial methanol plant, the proposed TF-DNN-based method was compared with the other five methods, which are DPCA-SVM, MLP, convolutional neural network (CNN), RNN and vanilla LSTM-BN. The candidate structures and parameters for these methods was listed in Table 4, where the structure of the entire network is the number of neurons in input, hidden and output layers. Fig. 10 shows the overall loss and accuracy curve for the training set of TF-DNN, CNN, vanilla LSTM-BN, RNN and MLP during 200 training epochs. Experimental results in Fig. 10 show that there is no overfitting during training, and accuracy of TF-DNN is close to the value of 1 in the training set.

For a more detailed analysis, the ROC and P-R curves of CNN, vanilla LSTM-BN, MLP and TF-DNN over the fault mode are shown in Fig. 11. From Fig. 11(a) - 11(d), we can see that the AUC score of the TF-DNN ROCs provides improvement in monitoring performance when the failure presents (macro-average ROC curve $\mathrm{AUC}_{\mathrm{TF-DNN}} = 0.98 > \mathrm{AUC}_{\mathrm{vanillaLSTM-BN}} = 0.93 > \mathrm{AUC}_{\mathrm{MLP}} = 0.86 > \mathrm{AUC}_{\mathrm{PCA-SVM}} = 0.83$). Another performance metric, precision-recall curve (P-R), was used for further evaluating the classification performance. Fig. 11(e) - 11(h) show the P-R plots of CNN, vanilla LSTM-BN, RNN, MLP and TF-DNN over the fault mode. It is clearly observed that TF-DNN presents a better quality of the overall performance in the P-R space, which is confirmed by the micro- and macro-average AUC (micro-average P-R curve $\mathrm{AUC}_{\mathrm{TF-DNN}} = 0.984 > \mathrm{AUC}_{\mathrm{vanillaLSTM-BN}} = 0.931 > \mathrm{AUC}_{\mathrm{MLP}} = 0.89 > \mathrm{AUC}_{\mathrm{CNN}} = 0.816 > \mathrm{AUC}_{\mathrm{MLP}} = 0.778$). Therefore, the TF-DNN method provides obviously the best performances focusing on the faulty cases.

Fig. 12 summarizes the results of F1 scores with the above six methods in the test dataset. It can be seen that TF-DNN gives the best simultaneous fault diagnosis results, of which the F1 score reaches nearly 90 %. It is particularly noteworthy that the F1 score of TF-DNN method under abnormal conditions is comparatively high with that of RNN, which means that TF-DNN achieves the promised diagnosis to the abnormal operating condition. We believe this phenomenon is due to the fact that the dynamic information is effectively captured by the hidden layers in the TF-DNN.

To provide insight on how the deep model structure improves fault detection performance, we investigated the features at the last hidden states. The t-SNE was used to reduce the dimensions of layers to two dimensions since high-dimension lies in each layer. Fig. 13 shows the distribution of features at the last hidden states of RNN and TF-DNN for the test data set, where "+" indicates that the median point in the corresponding category. From Fig. 13(a), it can be seen that features for modeling different operating conditions are grouped together. However, as shown in Fig. 13(b), it is clear that many points from the same conditions are almost grouped together, while most of the samples from different operating conditions are evidently separated. This makes TF-DNN more efficient to separate out different failure

events. This further corroborates the benefit of TF-DNN for the feature learning of industrial data.

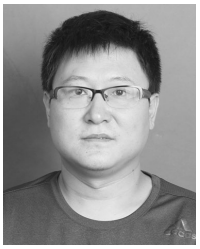
## VI. CONCLUSION

In this paper, a deep network based on tensor factorization layer has been proposed for fault detection and diagnosis in industrial process. The developed method has the following notable features: (1) TF layer not only preserves the advantages of FC layer but is also suitable for extracting the spatiotemporal features of fault data. (2) By stacking the multiple TF layers, TF-DNN is trained with an end-to-end manner which synchronously updates the parameters of the feature extraction and classifier. 3) Extensive experiments in the benchmark TE process and an industrial methanol plant between TF-DNN and the state-of-the-art classification methods show the proposed TF layer mechanism not only improves the performance of deep network structure but also enhances its feature expressive ability.

In the future work, the proposed TF layer can also be used for the other deep architecture, i.e., LSTM, BiLSTM CNN, CNN-LSTM etc. Moreover, the temporal dependencies across different time-steps should be preserved to improve the generalization capability. Hence, the attention augmented TF-DNN should be a further direction to the FDD applications.

## REFERENCES

[1] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.
[2] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometric Intell. Lab. Syst.*, vol. 171, pp. 16–25, Dec. 2017.
[3] Q. Jiang, X. Yan, and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes," *Ind. Eng. Chem. Res.*, vol. 58, no. 29, pp. 12899–12912, Jul. 2019.
[4] S. Bezergianni and A. Kalogianni, "Application of principal component analysis for monitoring and disturbance detection of a hydrotreating process," *Ind. Eng. Chem. Res.*, vol. 47, no. 18, pp. 6972–6982, Sep. 2008.
[5] D. Zhou, G. Li, and S. J. Qin, "Total projection to latent structures for process monitoring," *AIChE J.*, vol. 56, no. 1, pp. 168–178, 2010.
[6] Z. Ge and Z. Song, "Process monitoring based on independent component analysis-principal component analysis (ICA- PCA) and similarity factors," *Ind. Eng. Chem. Res.*, vol. 46, no. 7, pp. 2054–2063, 2007.
[7] C. Shang, F. Yang, D. Huang, and W. Lyu, "Data-driven soft sensor development based on deep learning technique," *J. Process Control*, vol. 24, no. 3, pp. 223–233, Mar. 2014.
[8] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring with kernel principal component analysis," *Ind. Eng. Chem. Res.*, vol. 49, no. 17, pp. 7849–7857, Sep. 2010.
[9] Y. Zhang, H. Zhou, S. J. Qin, and T. Chai, "Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 3–10, Feb. 2010.
[10] L. Wang and H. Shi, "Multivariate statistical process monitoring using an improved independent component analysis," *Chem. Eng. Res. Design*, vol. 88, no. 4, pp. 403–414, Apr. 2010.
[11] Q. Jiang and X. Yan, "Parallel PCA–KPCA for nonlinear process monitoring," *Control Eng. Pract.*, vol. 80, pp. 17–25, Nov. 2018.
[12] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019.
[13] Z. Zhang and J. Zhao, "A deep belief network based fault diagnosis model for complex chemical processes," *Comput. Chem. Eng.*, vol. 107, pp. 395–407, Dec. 2017.
[14] H. Zhao, S. Sun, and B. Jin, "Sequential fault diagnosis based on LSTM neural network," *IEEE Access*, vol. 6, pp. 12929–12939, 2018.
[15] H. Wu and J. Zhao, "Deep convolutional neural network model based chemical process fault diagnosis," *Comput. Chem. Eng.*, vol. 115, pp. 185–197, Jul. 2018.
[16] F. Zhou, S. Yang, H. Fujita, D. Chen, and C. Wen, "Deep learning fault diagnosis method based on global optimization GAN for unbalanced data," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104837.
[17] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[18] R. Pasricha, E. Gujral, and E. E. Papalexakis, "Identifying and alleviating concept drift in streaming tensor decomposition," in *Proc. ECML PKDD*, Dublin, Ireland, 2018, pp. 327–343.
[19] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, *Multilinear Subspace Learning: Dimensionality Reduction Multidimensional Data*. New York, NY, USA: Chapman & Hall, 2013.
[20] Q. Li and D. Schonfeld, "Multilinear discriminant analysis for higher-order tensor data classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2524–2537, Dec. 2014.
[21] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, Fort Lauderdale, FL, USA, 2011, pp. 315–323.
[22] Y. Dong and S. J. Qin, "A novel dynamic PCA algorithm for dynamic data modeling and process monitoring," *J. Process Control*, vol. 67, pp. 1–11, Jul. 2018.
[23] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. London, U.K.: Springer, 2001.
[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 7–9.
[25] A. Tharwat, "Classification assessment methods," *Appl. Comput. Inform.*, Aug. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210832718301546, doi: 10.1016/j.aci.2018.08.003.
[26] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.



**LIN LUO** received the B.Eng. and M.Eng. degrees from Liaoning Shihua University, Fushun, China, in 2007 and 2010, respectively, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2015.

In 2016, he became a Lecturer with the Faculty of Electrical and Control Engineering, Liaoning Technical University. Since 2017, he has been with the Department of Information and Control Engineering, Liaoning Shihua University. His research interests include monitoring, optimization and control of industrial process, and soft sensor.

**LEI XIE** received the B.S. and Ph.D. degrees from Zhejiang University, China, in 2000 and 2005, respectively. From 2005 to 2006, he was a Postdoctoral Researcher with the Berlin University of Technology. He was an Assistant Professor, from 2005 to 2008. He is currently a Professor with the Department of Control Science and Engineering, Zhejiang University. His research activities culminated in over 30 articles that are published in internationally renowned journals and conferences, three book chapters, and a book in the area of applied multivariate statistics and modeling. His research interests include interdisciplinary area of statistics and system control theory.

**HONGYE SU** (Senior Member, IEEE) was born in 1969. He received the B.S. degree in industrial automation from the Nanjing University of Chemical Technology, Jiangsu, China, in 1990, and the M.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1993 and 1995, respectively. He was a Lecturer with the Department of Chemical Engineering, Zhejiang University, from 1995 to 1997. He was an Associate Professor with the Institute of Advanced Process Control, Zhejiang University, from 1998 to 2000. He is currently a Professor with the Institute of Cyber-Systems and Control, Zhejiang University. His current research interests include the robust control, time-delay systems, and advanced process control theory and applications.