Supplementary Material for " Deep Learning of Latent Variable Models for Industrial Process Monitoring"

Xiangyin Kong, Zhiqiang Ge, *Senior Member, IEEE*

I. A BRIEF OVERVIEW OF PCA AND ICA

1) PCA Model

Suppose $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ is the training dataset after preprocessing, where N is the number of samples, while M is the variables. The covariance matrix of $\boldsymbol{X}$ is $\mathrm{cov}(\boldsymbol{X}) = \frac{\boldsymbol{X}^T\boldsymbol{X}}{N-1}$. Define λ_i $(1 \leqslant i \leqslant M)$ as the eigenvalues of the covariance matrix, which are sorted in descending order, and their corresponding eigenvectors $\boldsymbol{p}_i \in \mathbb{R}^M$ are known as loading vectors. The score vector $\boldsymbol{t}_i \in \mathbb{R}^N$, can be obtained by $\boldsymbol{t}_i = \boldsymbol{X}\boldsymbol{p}_i$ $(1 \leqslant i \leqslant M)$.

Combine all loading vectors into a loading matrix $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_M] \in \mathbb{R}^{M\times M}$, the first K vectors $\boldsymbol{P}_p = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_K] \in \mathbb{R}^{M\times K}$ construct the principal component subspace (PCS) $\boldsymbol{T}_p = \boldsymbol{X}\boldsymbol{P}_p$, and the whole linear features of PCA are gained by $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{P}$. The PCA residual matrix $\boldsymbol{E} \in \mathbb{R}^{N\times M}$ is defined as

$$\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{T}_p\boldsymbol{P}_p^T \tag{1}$$

The number selection of K is still an open question, in our paper, we adopt the average eigenvalue method, which means K is equal to the numbers of principal component whose corresponding eigenvalue is larger than the average eigenvalue.

For a new test sample $\boldsymbol{x}_{new}$, two monitoring statistics can be built by

$$T^2 = \boldsymbol{t}_{new}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{t}_{new} \tag{2}$$

$$Q_T = \|\boldsymbol{x}_{new} - \hat{\boldsymbol{x}}_{new}\|_2^2 \tag{3}$$

where $\boldsymbol{t}_{new} = \boldsymbol{P}_p^T\boldsymbol{x}_{new}$, $\hat{\boldsymbol{x}}_{new} = \boldsymbol{P}_p\boldsymbol{t}_{new}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{K\times K}$ is a diagonal matrix, and its diagonal elements are λ_i $(1 \leqslant i \leqslant K)$. The T^2 statistic is designed to monitor the information in the PCS, while the Q_T statistic is built for monitoring the data variations in the residual subspace.

2) ICA Model

Given an original dataset $\boldsymbol{X} \in \mathbb{R}^{M\times N}$ with M variables and N samples, the ICA algorithm supposes the dataset $\boldsymbol{X}$ can be represented as linear combinations of several independent sources, i.e.

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} \tag{4}$$

where $\boldsymbol{S} \in \mathbb{R}^{M\times N}$ is the independent sources matrix, while $\boldsymbol{A} \in \mathbb{R}^{M\times M}$ is the mixing matrix.

Define the separating matrix $\boldsymbol{W} = \boldsymbol{A}^{-1}$, the estimation of source signal can be computed by

$$\hat{\boldsymbol{S}} = \boldsymbol{W}\boldsymbol{X} \tag{5}$$

The objective of ICA is to find a matrix $\boldsymbol{W}$ which makes the components in $\hat{\boldsymbol{S}}$ become as independent of each other as possible.

A very important preprocessing step of ICA is whitening. Given a vector $\boldsymbol{x}$ with its covariance matrix $E(\boldsymbol{x}\boldsymbol{x}^T)$, where $E(\cdot)$ stands for expectation. The eigenvalue decomposition of the covariance matrix is calculated by

$$E(\boldsymbol{x}\boldsymbol{x}^T) = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T \tag{6}$$

Define $\boldsymbol{V} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}^T$, the whitening transformation can be expressed as

$$\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x} \tag{7}$$

From various ICA algorithms, we adopt the FastICA algorithm which is based on maximizing the non-Gaussianity of variables. The classical measure of non-Gaussianity is kurtosis, which is easy to calculate, unfortunately, it is very sensitive to data outliers. A more robust measure of non-Gaussianity is the negentropy, which is based on the information-theoretic quantity of (differential) entropy. For more details about FastICA algorithm, please refer to [1].

The separating matrix $\boldsymbol{W}$ can be gained through iterations, after recovering the independent sources matrix $\boldsymbol{S}$, we need to select a few independent components as the "dominant components". Since the negentropy is used for the measurement of non-Gaussianity, we calculate the negentropy value of each component, and then choose the largest K of them. The selected K components and their corresponding separating matrix are written as $\boldsymbol{S}_K \in \mathbb{R}^{K\times N}$ and $\boldsymbol{W}_K \in \mathbb{R}^{K\times M}$, respectively.

For a sample vector $\boldsymbol{x}_{new} \in \mathbb{R}^M$, which is assumed to be standardized and whitened, similar to PCA, two monitoring statistics are constructed by

$$I^2 = \boldsymbol{s}_{newk}^T\boldsymbol{s}_{newk} \tag{8}$$

$$Q_I = \|\boldsymbol{x}_{new} - \hat{\boldsymbol{x}}_{new}\|_2^2 \tag{9}$$

where $\boldsymbol{s}_{newk} = \boldsymbol{W}_K\boldsymbol{x}_{new}$, $\hat{\boldsymbol{x}}_{new} = \boldsymbol{W}_K^T\boldsymbol{s}_{newk}$.

II. DETAILED MONITORING PROCEDURES

During the offline modeling stage, the normal condition data are collected to build the model, then the KDE method is used to determine the confidence limits. In the online monitoring stage, the new test sample is first inputted into the model, then the corresponding values of different monitoring statistics are calculated. Finally we make a comprehensive judgement about

the process state through fusing these statistics. The detailed procedures are as follows.

Offline modeling procedures

(1) Collect the data under normal condition, divide it into a training dataset $\boldsymbol{X}_1$ and a validating dataset $\boldsymbol{X}_2$.

(2) Acquire the mean and variance of the training dataset $\boldsymbol{X}_1$, normalize both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ with the above mean and variance.

(3) Perform the PCA method on the training dataset $\boldsymbol{X}_1$ to obtain the score matrix $\boldsymbol{T}^{(l)}$ and the loading matrix $\boldsymbol{P}^{(l)}$ at each layer.

(4) Use the validating dataset $\boldsymbol{X}_2$ to compute PCA monitoring statistics $T^{2(l)}$ and $Q_P^{(l)}$ at each layer, at the same time, obtain the PCA residual matrix $\boldsymbol{E}$ of validating dataset $\boldsymbol{X}_2$.

(5) Whiten the residual matrix $\boldsymbol{E}$, perform the ICA algorithm on $\boldsymbol{E}$ to gain the ICA monitoring statistics $I^{2(l)}$ and $Q_I^{(l)}$ at each layer,

(6) Determine the confidence limits of all statistics at each layer by utilizing the KDE method.

Online monitoring procedures

(1) For a new test sample, normalize it with the mean and variance of the training dataset $\boldsymbol{X}_1$.

(2) Follow the steps that we implement to the validating dataset in the offline modeling procedure to calculate the monitoring statistics at each layer of the new test sample.

(3) Transform the monitoring statistics at each layer into the probability-based index based on the fusion strategy.

(4) Transform the four probability-based indexes into an overall probability-based statistic, then monitor this statistic and make a comprehensive judgement about the process state based on it.

The online monitoring steps of the proposed model is shown in Fig. S1.

III. Figures and Tables

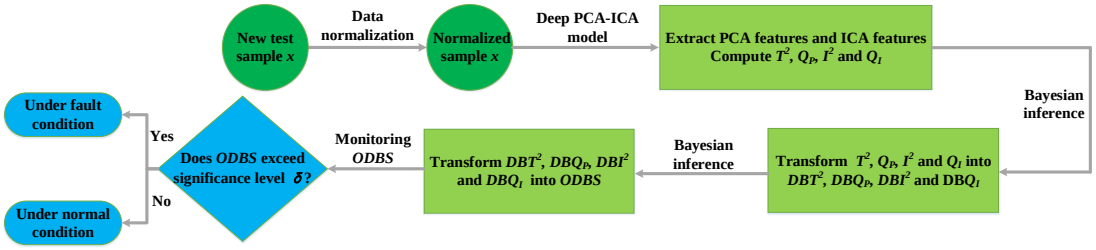


Fig.S 1. The online monitoring steps of the proposed deep PCA-ICA model.

Table S I. 21 TE Process Faults

Fault	Variable	Fault type
1	A/C feed ratio, B composition constant (stream 4)	step
2	B composition, A/C ratio constant (stream 4)	step
3	D feed temperature (stream 2)	step
4	reactor cooling water inlet temperature	step
5	condenser cooling water inlet temperature	step
6	A feed loss (stream 1)	step
7	C header pressure loss-reduced availability (stream 4)	step
8	A, B, C feed composition (stream 4)	random
9	D feed temperature (stream 2)	random
10	C feed temperature (stream 4)	random
11	reactor cooling water inlet temperature	random
12	condenser cooling water inlet temperature	random
13	reaction kinetics	slow drift
14	reactor cooling water	sticking
15	condenser cooling water	sticking
16	unknown	unknown
17	unknown	unknown
18	unknown	unknown
19	unknown	unknown
20	unknown	unknown
21	valve position constant (stream 4)	sticking

Table S II. Detailed MDR (%) On 21 TE Process Faults Obtained By Several Methods

Fault	PCA		ICA		DAE	CAE	RCAE	ARCAE	DePCA	DPI-2L	DPI-3L
	T^2	Q_P	I^2	Q_I	Q_D	Q_C	Q_R	Q_A	PT^2	$ODBS$	$ODBS$
1	0.500	0.250	0.375	0.250	**0**	0.500	0.625	0.750	0.250	0.250	0.250
2	1.750	1.250	1.250	1.500	1.625	1.375	1.625	2.000	1.250	1.250	**1.125**
3	92.000	92.875	95.625	86.625	89.750	89.500	92.625	87.000	93.125	91.500	90.875
4	72.000	0.000	0.500	28.625	0.000	26.000	1.125	1.125	0.000	0.000	**0.000**
5	69.375	69.375	0.000	0.000	0.000	65.750	26.125	3.750	0.000	0.000	**0.000**
6	0.750	0.000	0.000	0.000	0.000	0.000	2.500	0.125	0.000	0.000	**0.000**
7	0.125	0.000	0.625	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**
8	2.500	2.125	2.125	2.250	1.750	**1.375**	3.625	2.000	2.000	2.000	2.000
9	92.625	94.625	96.625	86.250	89.250	90.500	94.000	87.375	95.125	93.000	92.250
10	50.875	47.875	20.750	16.250	19.375	42.375	42.750	34.625	9.000	7.750	**7.250**
11	53.125	27.500	32.125	37.125	17.250	37.875	37.750	**1.625**	22.250	15.000	14.125
12	1.000	2.375	0.375	0.250	0.375	0.625	5.100	**0.000**	0.125	0.125	0.125
13	5.000	4.625	4.875	4.625	**4.250**	4.750	5.625	5.875	4.500	4.500	4.500
14	1.000	0.000	0.000	0.000	0.000	0.000	2.125	0.125	0.000	0.000	**0.000**
15	87.875	91.875	94.250	86.625	87.000	83.125	91.125	79.750	80.875	79.750	77.500
16	68.000	53.875	17.750	19.125	18.625	54.375	50.500	37.725	6.000	4.750	**4.000**
17	18.625	6.5008	7.875	7.000	**2.375**	10.000	24.500	3.125	3.500	2.500	2.500
18	10.500	8.625	9.500	9.750	8.625	8.875	**8.500**	8.875	9.625	9.000	8.750
19	91.750	72.125	26.250	41.875	20.125	94.250	74.125	56.625	8.875	4.625	**3.125**
20	53.625	43.750	14.250	18.625	27.250	41.750	42.125	24.875	9.375	9.000	**8.625**
21	61.250	50.000	49.250	49.875	42.375	56.250	46.875	52.875	38.000	35.875	**33.875**
Average	39.726	31.887	22.590	23.649	20.476	33.774	31.112	23.344	18.196	17.185	**16.708**

Note: Preferred model for each fault is in **bold**. It should be noted that, the MDRs of faults 3, 9 and 15 obtained by different models are all very high. In fact, the above three faults are extremely hard to detect by most existing data-driven monitoring methods, because there is no obvious data variation in the process of these three faults, and these faults have almost no effect on the overall TE benchmark [2].

Table S III. Network architecture of the convolutional autoencoders (CAE)

Input (1×52)
Conv(1×3)-16
Dropout(0.5)/BN/Relu
Conv(1×3)-32
Dropout(0.5)/BN/Relu
Conv(1×3)-64
Dropout(0.5)/BN/Relu
ConvTrans(1×3)-32
Dropout(0.5)/BN/Relu
ConvTrans(1×3)-16
Dropout(0.5)/BN/Relu
ConvTrans(1×3)-1
Dropout(0.5)/BN/Relu
Output (1×52)

Note: BN: batch normalization. ConvTrans: deconvolution. "Conv(1×3)-16" means kernel size is 1×3 and output channels are 16.

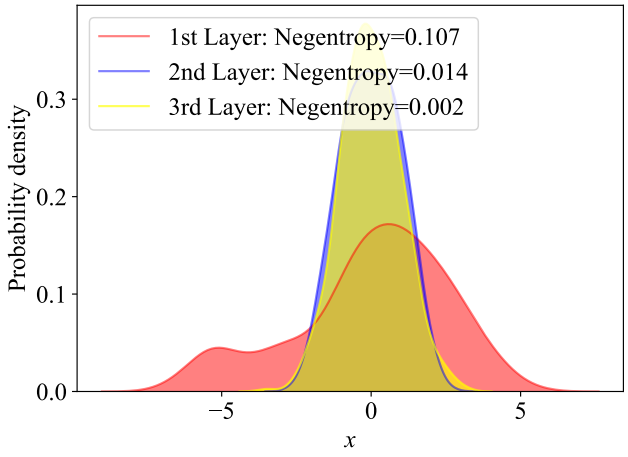


Fig.S 2. Probability density function of the fist principal component at three different layers.

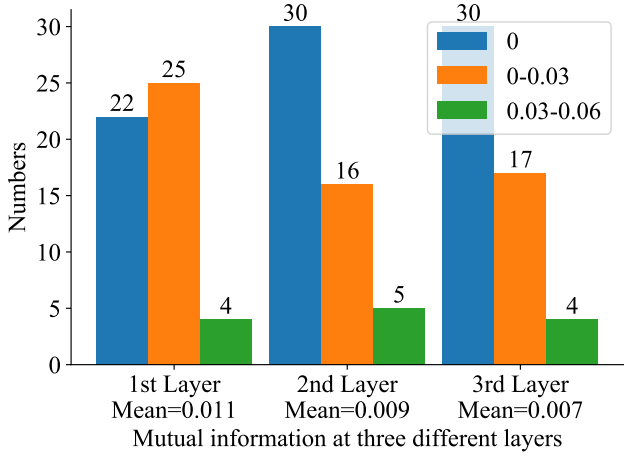
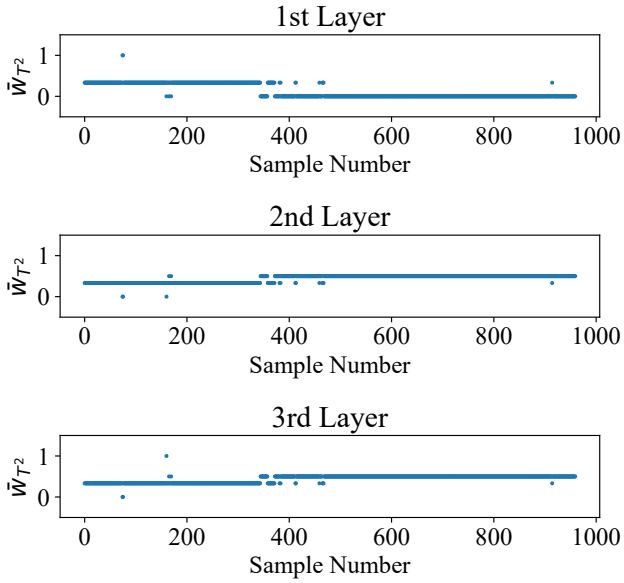
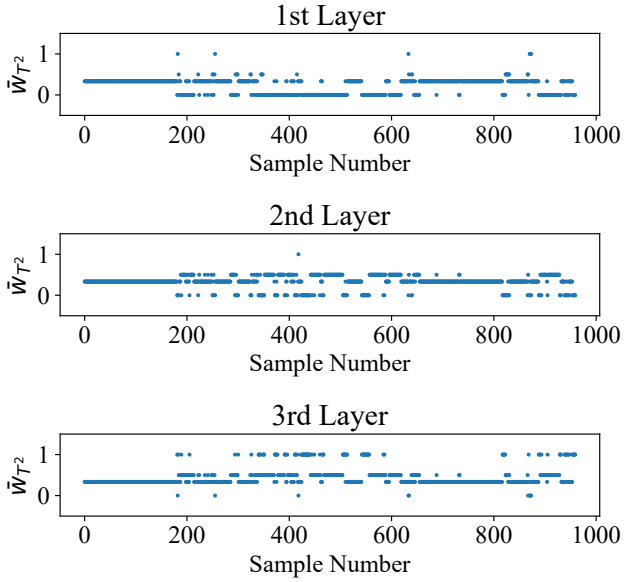


Fig.S 3. Mutual information between the first feature and the remaining 51 features at different layers.



(a) Fault 5



(b) Fault 10

Fig.S 4. Standardized weighting coefficients of T^2 at each layer for fault 5 and 10.

REFERENCES

[1] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[2] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.