2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Incipient fault detection with feature ensemble based on one-class machine learning methods

Min Wang[1], Feiyang Cheng[2], Kai Chen[3], Jinhua Mi[4], Zhiwei Xu[5], and Gen Qiu[6]

***Abstract*— Considering production quality and process safety, incipient fault detection has drawn more and more attention. With the rapid development of machine learning, numerous researches for fault detection based on machine learning have been published. However, almost all machine learning methods used for fault detection need abnormal data to construct models. Unfortunately, it is difficult to obtain sufficient fault samples in practical industrial processes. In addition, the existing fault detection methods are based on single feature extraction strategy. Process monitoring methods with different working principles often extract and utilize different process information. Reasonable integration of features extracted by multiple methods can usually effectively improve the performance of incipient fault detection. Therefore, this paper proposes an one-class machine learning feature ensemble model (OCMLFEM) for incipient fault detection. In OCMLFEM, various one-class machine learning models are constructed as basic detectors. In order to effectively mine the features obtained by basic detectors, a feature ensemble strategy with the technologies of sliding window singular value and principal component analysis is adopted. Then, Tennessee Eastman process is utilized to verify the validity of the proposed detection model, which proves that OCMLFEM has significant superiority.**

***Index Terms*— fault detection, ensemble strategy, machine learning**

I. Introduction

With the continuous development of industrial technology, industrial processes are becoming more and more complex [1]–[3]. Industrial faults may cause hazardous abnormal events and property loss [1], [2], [4]. Therefore, timely and effective fault detection can not be ignored [5], [6]. However, incipient faults may transform and spread along the topology structure of the complex system, so that effective fault detection is very challenging. In recent years, the data-driven fault detection methods have drawn much more attention because they do not require complex structural information and sufficient expert experience [7]–[9].

Numbers of data-driven fault detection methods have been proposed. Among them, principal component analysis (PCA) is one of the most classical methods due to its simplified computation [10]. Then, dynamic PCA (DPCA) is proposed which considers the dynamic feature of process based on PCA [11]. Furthermore, independent component analysis (ICA) is used to detect faults by separating independent variables [12]. However, the performance of above methods is not as good as expected in certain situations. Taking Tennessee Eastman process (TEP) as an example, faults 3, 5, 9, etc., are difficult to be detected through these methods [13]. In addition, machine learning has attracted increasing attention in process monitoring due to its nonlinear transformation characteristics, which could extract features more adequately. Thus, fault detection models based on long short term memory network (LSTM) [14] and temporal convolutional network (TCN) [15] have been proposed recently. Moreover, the above machine learning methods performance effectively due to their its advantage in dealing with time series data. However, they require fault data for modeling. Usually, it is difficult to obtain sufficient abnormal data in the practical processes [9]. In order to solve this problem, one-class machine learning methods such as one-class support vector machine (OCSVM) [16] and isolation forest (iForest) [17] are introduced for process monitoring.

Furthermore, different fault detection methods extract and utilize different information because of their different working principles. If the process features mined by different methods are integrated in a reasonable and effective way, it may obviously improve the process monitoring performance. Classical ensemble strategies are simple to calculate which include but are not limited to averaging, voting, and Bayesian inference [18], [19]. However, above classical ensemble strategies are not effective enough. Recently, Liu et al. [20] have proposed a novel deep ensemble framework, which provides a new approach for ensemble learning.

In this paper, one-class machine learning feature ensemble model (OCMLFEM) is proposed for fault detection. The OCMLFEM includes two modules: basic detectors and ensemble strategy. In order to build a model trained without fault data, one-class approaches are adopted in the design of basic detectors. The one-class basic detectors constructed in this paper are OCSVM, iForest, Autoencoders (AE) [21] based on back propagation neural Network (BP) [22], LSTM and TCN. Then, the feature ensemble strategy is utilized to fuse the information mined by one-class basic detectors. Compared to the existing detectors, OCMLFEM is more efficient because it could extract sufficient information from the samples. In order to verify the validity of the proposed model, OCMLFEM is tested in TEP. Some classical ensemble strategies (averaging, voting and Bayesian inference)

This work was supported by the National Natural Science Foundation of China under Grant 62303090, U2230206, and Postdoctoral Program for Innovative Talents of Shandong Province under Grant SDBX2022001. (Corresponding author: Gen Qiu.)

[1]Min Wang, [2]Feiyang Cheng, [3]Kai Chen, [4]Jinhua Mi, [6]Gen Qiu are with School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (mwang@uestc.edu.cn, 2021060904025@std.uestc.edu.cn, kaichen@uestc.edu.cn, jinhuami@uestc.edu.cn, qgen615@uestc.edu.cn).

[5]Zhiwei Xu is with School of Control Science and Engineering, Shandong University, Jinan 250061, China (zwxu@email.sdu.edu.cn).

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

are compared with OCMLFEM under the same conditions, and the results illustrate the proposed approach has the superiority for process monitoring. The main contributions are summarized as follows:

(1) **This paper constructs a feature ensemble model based on one-class machine learning.** Several unsupervised machine learning models with nonlinearly mapping are introduced in proposed model. Thus, proposed model extracts nonlinear information fully and detects incipient faults effectively.

(2) **The proposed model gains superior detection performance in experiment.** Since nonlinear extraction ability is introduced, the proposed model has utilized the information of process sufficiently. Taking faults 3, 5 and 9 in TEP as examples, the model performs further positive than previous methods.

The remainder of this paper is structured as follows. Section II introduces the basic detector. In Section III, the integration strategy will be explained. Section IV is the simulation of the detect methods. Conclusions are drawn in Section V.

II. Basic Detector

In this section, the basic detectors utilized in this paper are described in detail.

A. OCSVM

One-class support vector machine (OCSVM) is adopted in this paper as a basic detector. In the theory of soft interval OCSVM [16] whose schematic diagram is shown in Fig. 1, a hyperplane could be found to classify normal and abnormal data. The hyperplane is obtained by:

$$\min_{\boldsymbol{\omega},\xi_i,\rho} \frac{1}{2}\left\|\boldsymbol{\omega}\right\|^2 + \frac{1}{vn}\sum_{i=1}^{n}\xi_i - \rho$$
$$s.t. \begin{cases} (\boldsymbol{\omega}\cdot\phi(\boldsymbol{x_i})) \geq \rho - \xi_i & i = 1,\ldots,n \\ \xi_i \geq 0 & i = 1,\ldots,n \end{cases} \tag{1}$$

where $\boldsymbol{\omega}$ and ρ are parameters of hyperplane in the point product space, n is the number of samples, $\boldsymbol{x}_i$ is the ith sample, ξ_i is the slack variable penalized in the object function, v controls the upper limit of the proportion of abnormal points in the training process and $\phi(\cdot)$ is the feature map from the original space to a point product space. The slack variable ξ_i is introduced to function $f(\boldsymbol{x_i}) = \boldsymbol{\omega}\cdot\phi(\boldsymbol{x_i})) - \rho + \xi_i$ be positive for most samples. So the slack variable ξ_i is expected to be non-negative in constraint condition [16].

Under the assumption that the normal process data is concentrated, the square distance between the sample point and the hyperplane will be taken as statistic $\boldsymbol{\alpha}^1$.

B. Isolation Forest

In the iForest theory [17], abnormal data samples are considered to be easily isolated from the normal data, because the probability of data occurring in the sparsely distributed areas is very low.

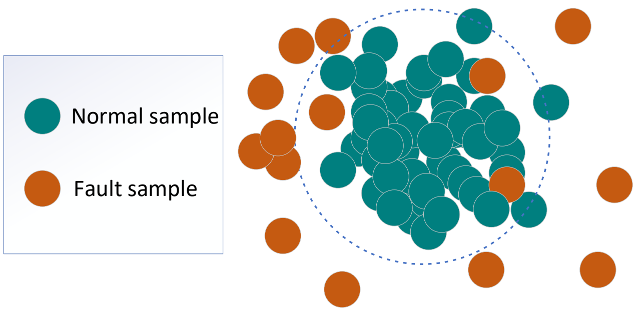


Fig. 1: Schematic diagram of OCSVM

Therefore, binary search tree can be constructed through multiple iterations named as isolation tree (iTree), and then a group of iTrees can be formed into an isolation forest. In each of the iTrees, the samples are placed into each final child node according to a set of rules. In practice process, various features and classification principles would be chosen randomly in training. The average distance in the forest between a sample and the root node is considered as the final statistic which is denoted as $\boldsymbol{\alpha}^2$. For example, an iTree shown in Fig. 2 is constructed by two variables, temperature (T) and air pressure (P), in which samples are classified according to the range of these two variables and put into different child nodes.

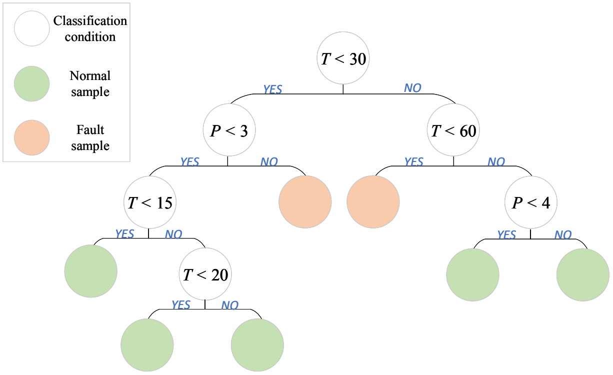


Fig. 2: Example of an iTree in the isolation forest

C. Autoencoder

Autoencoder based on BP is adopted to construct the one-class deep learning method [21], whose framework is shown in Fig. 3. In the encoder layer, the q-dimension feature is obtained from the p-dimension data. Then the p-dimension output is predicted by the decoder layer. The statistics $\boldsymbol{\alpha}^3$ is defined with the loss of output:

$$\mathbf{Loss} = ||\mathbf{X} - \mathbf{X}'||^2, \tag{2}$$

where $\mathbf{X}$ is the input, $\mathbf{X}'$ is the prediction of BP and $||\cdot||$ represents the 2-norm of the vector.

In addition to the BP network model, LSTM and TCN are also adopted as autoencoders due to their excellent

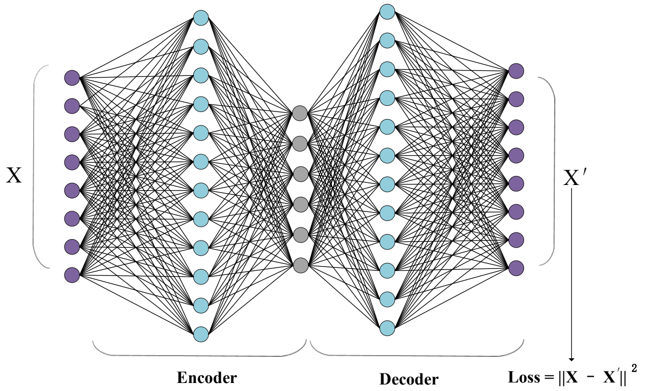


Fig. 3: Framework of Autoencoder

performance in dealing with time series. $\boldsymbol{\alpha}^4$ and $\boldsymbol{\alpha}^5$ are the statistics of these two models.

III. Ensemble Strategy

Based on the statistics obtained in the Section II, this section introduces five ensemble strategies.

A. Voting

After statistics vectors $\boldsymbol{\alpha}^j$ $(j = 1, 2, \cdots, m)$ have been obtained as well as the corresponding control limits δ^j $(j = 1, 2, \cdots, m)$ which are calculated by kernel density estimation method (KDE) [23]. Voting method follows the principle of minority obedience to majority. Through the ensemble of multiple features, the performance of the model can be improved. Ideally, the predictive effect of the voting method should be better than any basic models.

To build the voting integration model, the following statistics is defined:

$$H(x_i) = \sum_{j=1}^{m} h_j(x_i), \tag{3}$$

where $h_j(x_i)$ is the judgment of the jth model on the ith sample which is defined as:

$$h_j(x_i) = \begin{cases} 1, & \alpha_i^j > \delta^j \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where α_i^j represents the statistic for ith sample obtained by jth detector.

When $H(x_i) \leq m/2$, the process is under the normal operation status. Otherwise, the system is faulty.

B. Averaging

In order to eliminate the influence of various dimensions of statistics obtained by different models, the obtained statistics are first standardized as $\hat{\boldsymbol{\alpha}}^j$, which could be calculated by:

$$\hat{\boldsymbol{\alpha}}^j = \frac{\boldsymbol{\alpha}^j - \mu^j}{\sigma^j}, \tag{5}$$

where μ^j is the mean of $\boldsymbol{\alpha}^j$, σ^j is the standard deviation of $\boldsymbol{\alpha}^j$. The ensemble statistic $M(x_i)$ could be obtained by:

$$M(x_i) = \frac{1}{m} \sum_{j=1}^{m} \hat{\alpha}_i^j. \tag{6}$$

C. Bayesian Inference

To fuse the statistics, Bayesian inference [19] is adopted. For the ith sample $\boldsymbol{x}_i$, the posterior fault probability $P_{\alpha_i^j}(F \mid x_i)$ and prior fault probability $P_{\alpha_i^j}(x_i \mid F)$ could be obtained by:

$$P_{\alpha_i^j}(F \mid x_i) = \frac{P_{\alpha_i^j}(x_i \mid F) P_{\alpha_i^j}(F)}{P_{\alpha_i^j}(x_i)} \tag{7}$$

$$P_{\alpha_i^j}(x_i \mid F) = exp(-\gamma \frac{\delta^j}{\alpha_n^j}), \tag{8}$$

where F is the event that fault appears and γ is the tuning parameter designed to decrease the sensitivity to the outlier data (In this paper, γ is empirically chosen to be 1). $P_{\alpha^j}(F)$ is the fault probability with significance level $1 - \eta$.

$P_{\alpha_i^j}(x_i)$ is the occurrence probability of x_i:

$$P_{\alpha_i^j}(x_i) = P_{\alpha_i^j}(x_i \mid F) P_{\alpha_i^j}(F) + P_{\alpha_i^j}(x_i \mid N) P_{\alpha_i^j}(N), \tag{9}$$

where N means the normal operation. And $P_{\alpha_i^j}(x_i \mid N)$ is the occurrence probability of x_i under the normal condition, calculated by

$$P_{\alpha_i^j}(x_i \mid N) = exp(-\gamma \frac{\alpha_i^j}{\delta^j}). \tag{10}$$

Then the final statistics $W\delta$ could be obtained by:

$$W\delta = \sum_{j=1}^{m} \frac{P_{\alpha_i^j}(F \mid x_i) P_{\alpha_i^j}(x_i \mid F)}{\sum_{j=1}^{m} P_{\alpha_i^j}(F \mid x_i)}. \tag{11}$$

When $W\delta \leq 1 - \eta$, the process is under the normal operation status. Otherwise, a fault is detected.

D. Weighted-Voting/Averaging

In order to improve the performance of the above two strategies (averaging, voting), the final statistics could be optimized by using the weights ω_j^i obtained by Bayesian inference, which is:

$$\omega_j^i = \frac{P_{\alpha_i^j}(F \mid x_i)}{\sum_{j=1}^{m} P_{\alpha_i^j}(F \mid x_i)}. \tag{12}$$

The weighted-voting strategy's statistic is

$$H'(x_i) = \sum_{j=1}^{m} \omega_j^i h_j(x_i) \tag{13}$$

The weighted-averaging strategy's statistic is

$$M'(x_i) = \sum_{j=1}^{m} \omega_j' \hat{\alpha}_i^j \tag{14}$$

Different from the direct use of voting strategy, the weight of average strategy ω_j' is the mean of jth detector's weights obtained during the training process ($\omega_j' = \frac{1}{n} \sum_{i=1}^{n} \omega_j^i$), because the real-time changing weight will make the control limit obtained by kernel density estimation unreliable.

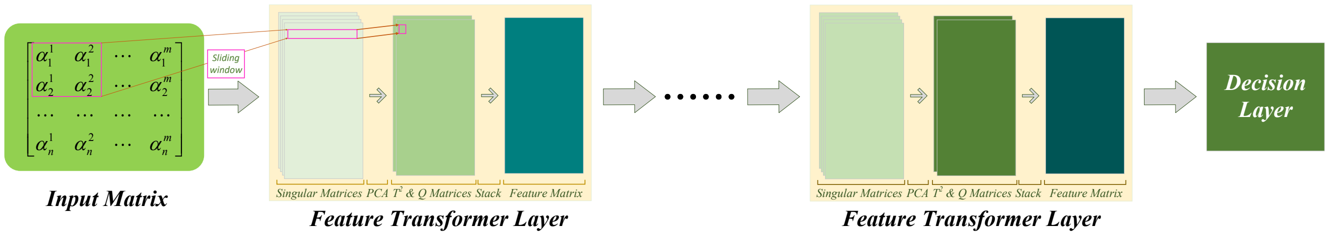


Fig. 4: Framework of FENet

E. FENet

FENet is a new integration strategy proposed recently [20], which extracts the deep features by using multiple feature transformer layers. Finally, a decision layer based on singular values is utilized to obtain the final statistics. The whole framework is Fig. 4.

The input matrix $\boldsymbol{S}$ is obtained by the base detectors and shown as:

$$\boldsymbol{S}=\begin{bmatrix}\alpha_1^1 & \alpha_1^2 & \cdots & \alpha_1^m\\ \alpha_2^1 & \alpha_2^2 & \cdots & \alpha_2^m\\ \cdots & \cdots & \cdots & \cdots\\ \alpha_n^1 & \alpha_n^2 & \cdots & \alpha_n^m\end{bmatrix}\in\mathbb{R}^{n\times m}. \tag{15}$$

In the lth transformer layer, the input matrix is denoted as $\boldsymbol{S}^l\in\mathbb{R}^{n_l\times m_l}$. For mining the hidden process features, sliding window with size $\omega\times h_l$ (in this paper, $\omega=8$ and $h_l=m_l-1$) is utilized. After choose h_l columns, the sliding window first slides through the column to obtain the feature matrix of samples. Then it will choose other combination of h_l columns sliding again until all the combinations occur. The sliding result of the qth sample is $\boldsymbol{S}^l_{q,u_l}$, where $u_l=1,...,C^{h_l}_{m_l}$ represents the u_lth combination of columns. Then, $\boldsymbol{S}^l_{q,u_l}$ is normalized as

$$\bar{\boldsymbol{S}}^l_{q,u_l}=(\boldsymbol{S}^l_{q,u_l}-\boldsymbol{1}_w\boldsymbol{\mu}^T_{0,u_l})\boldsymbol{\Sigma}^{-1}_{0,u_l} \tag{16}$$

where $\boldsymbol{1}_w=[1,1,\dots,1]^T\in\mathbb{R}^w$, $\boldsymbol{\mu}_{0,u_l}\in\mathbb{R}^{h_l}$, and $\boldsymbol{\Sigma}_{0,u_l}\in\mathbb{R}^{h_l\times h_l}$ are the mean and standard deviation of $\boldsymbol{S}^l_{u_l}$, respectively.

Because the singular values are related to the proportion of fault samples [24], singular value decomposition (SVD) is used for $\bar{\boldsymbol{S}}^l_{q,u_l}$ to obtain singular values $\boldsymbol{\sigma}^l_{q,u_l}\in\mathbb{R}^{h_l}$ stacked into $\boldsymbol{V}^l_{q,u_l}\in\mathbb{R}^{(n_l-w+1)\times h_l}$. Applying PCA to $\boldsymbol{V}^l_{q,u_l}$, the statistics O (O represents T^2 and Q) are stacked into:

$$\boldsymbol{O}^l=\begin{bmatrix}O_{n-n_l+w,l,1} & \cdots & O_{n-n_l+w,l,C^{h_l}_{m_l}}\\ \vdots & \ddots & \vdots\\ O_{n,l,1} & \cdots & O_{n,l,C^{h_l}_{m_l}}\end{bmatrix}, \tag{17}$$

where O_{q,l,u_l} represents the corresponding statistics obtained from $\boldsymbol{V}^l_{q,u_l}$.

Next, the output feature matrix in the $(l+1)$th transformer layer is denoted by

$$\boldsymbol{S}^{l+1}=[\boldsymbol{T}^l,\boldsymbol{Q}^l]\in\mathbb{R}^{(n_l-w+1)\times 2h_l}. \tag{18}$$

In the output layer, the output feature matrix could be stacked by all feature matrixes generated from different sizes of the sliding window patches denoted as $\boldsymbol{S}^o\in\mathbb{R}^{n_o\times m_o}$.

In decision layer, the sliding window is employed to obtain $\boldsymbol{S}^o_q\in\mathbb{R}^{w\times m_o}$, where $q=n-n_o+w,n-n_o+w+1,\dots,n$. Furthermore, the $\boldsymbol{S}^o_q$ is scaled as $\bar{\boldsymbol{S}}^o_q$, and the singular values of $\bar{\boldsymbol{S}}^o_q$ are obtained as $\boldsymbol{\sigma}^o_q\in\mathbb{R}^h$, where h represents the number of sigular values.

Thus, the final statistic for qth sample is designed as

$$D_q=||\boldsymbol{\Phi}^{-1}(\boldsymbol{\sigma}^o_q-\boldsymbol{\kappa})||_2, \tag{19}$$

where $\boldsymbol{\kappa}\in\mathbb{R}^{m_o}$ and $\boldsymbol{\Phi}\in\mathbb{R}^{m_o\times m_o}$ are the mean and standard deviation of matrix $[\boldsymbol{\sigma}^o_{n-n_o+\omega},\dots,\boldsymbol{\sigma}^o_n]^T$.

IV. Simulation

The Tennessee Eastman process (TEP) is developed as an open simulation platform for chemical process, which simulates an practical chemical combination reaction process at Eastman chemical company in the United States. The simulation datasets are composed of several process variables, among which 31 variables are adopted in this paper. In the datasets, d00 is normal process data. Hence, it could be used to train models and its false detection rate (FDR) is used as false alarm rate (FAR) [25]. Datasets d01-d21 are fault data caused by various faults, which are used for detector testing. In each test data, the first 2000 samples are normal data, and the fault is introduced from 2001th sample. That is, the last 2000 samples are fault instances.

The test results are shown in TABLE I (for OCMLFEM, l^{max} is shortened as l). It is not difficult to find that basic detectors are not effective for detecting faults 3, 5, 9, 12, 15, 16, 18 and 21. The performances of basic detectors for fault 3 are shown in Fig. 6. In addition, the detection results of the ensemble strategies for fault 3 are depicted as Fig. 7. From the experimental results, most of the ensemble strategies have improved detection performance over the basic detector. Among them, OCMLFEM has a much more efficient detection. For faults 3,5,9 and 15, OCMLFEM can achieve 95% in the case of $l^{max}=3$, while the FDRs of other detectors are all lower than 30%.

V. Conclusion

In this paper, OCMLFEM has been proposed for incipient fault detection based on one-class machine learning

TABLE I: The FDRs(%) of various approaches

Fault	PCA		OCSVM	iForest	AE			Averaging	Voting	Weighted		Bayesian Inference	OCMLFEM		
	T^2	Q			BP	LSTM	TCN			Averaging	Voting		$l=1$	$l=2$	$l=3$
00	0.85	0.85	0.0	0.0	0.6	0.6	0.4	0.75	0.0	0.0	1.05	1.0	0.15	0.0	0.0
01	99.9	99.95	99.85	18.8	99.9	99.85	99.75	99.85	99.85	99.85	99.85	99.85	99.8	99.8	**99.8**
02	99.65	99.9	99.75	99.15	99.65	99.55	99.25	99.8	99.5	99.75	99.6	99.75	99.75	99.75	**99.75**
03	1.65	4.1	13.25	0.0	1.15	1.5	2.35	27.25	0.25	8.95	4.05	5.65	77.4	87.6	**95.0**
04	99.95	99.95	59.2	0.0	99.95	99.9	99.9	99.95	99.9	99.95	99.95	99.95	99.95	99.95	**99.95**
05	1.5	2.0	18.05	0.0	1.3	1.1	1.45	29.2	0.1	12.3	2.8	5.8	83.5	93.35	**98.05**
06	100.0	100.0	100.0	97.2	100.0	99.3	99.3	100.0	99.3	100.0	100.0	100.0	100.0	100.0	**100.0**
07	100.0	100.0	100.0	0.8	100.0	99.95	99.95	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
08	99.6	99.65	99.3	95.5	99.6	99.55	99.45	99.75	99.5	99.6	99.55	99.55	99.6	99.6	**99.6**
09	1.45	6.3	14.95	0.0	1.1	4.05	0.95	26.3	0.4	10.25	4.2	6.35	74.1	87.6	**95.95**
10	74.0	92.35	24.05	0.05	84.4	96.5	88.1	96.5	83.4	95.5	96.75	95.75	99.0	99.5	**99.55**
11	98.2	97.1	82.2	5.55	97.75	99.85	98.7	99.55	97.55	99.75	99.7	99.65	99.8	99.8	**99.8**
12	34.25	32.9	43.6	1.2	38.55	81.95	40.5	86.05	37.65	77.45	83.3	74.8	99.25	99.75	**99.75**
13	97.45	97.5	97.85	96.6	97.7	97.3	97.85	98.4	97.55	97.9	97.8	98.0	99.2	99.5	**99.55**
14	99.9	99.9	22.4	0.15	99.9	99.85	99.75	99.85	99.75	99.85	99.85	99.85	99.75	99.75	**99.75**
15	1.05	2.05	12.8	0.0	0.95	5.1	0.5	23.1	0.05	9.35	3.0	6.85	78.25	91.4	**98.0**
16	0.75	1.05	13.3	0.0	0.75	0.25	0.45	23.35	0.05	8.3	0.3	3.0	76.55	87.75	**95.4**
17	98.3	99.1	85.75	2.55	99.0	98.95	99.0	99.15	98.9	99.15	99.05	99.05	99.5	99.5	**99.55**
18	75.15	84.95	60.55	3.65	75.9	94.25	80.9	92.75	76.05	90.4	94.6	89.35	98.7	99.15	**99.55**
19	99.6	99.9	52.9	7.4	99.6	99.8	99.7	99.75	99.6	99.8	99.8	99.8	99.7	99.7	**99.7**
20	99.0	99.3	99.35	85.8	99.1	99.05	99.1	99.5	99.1	99.35	99.25	99.2	99.55	99.5	**99.55**
21	1.45	1.45	16.5	0.0	1.55	1.6	2.0	29.75	0.35	11.2	3.3	6.75	82.25	91.9	**97.35**

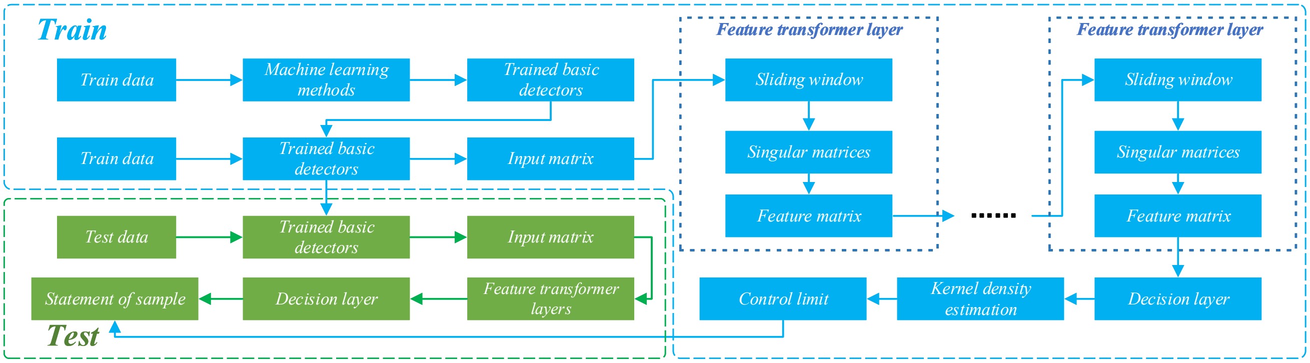
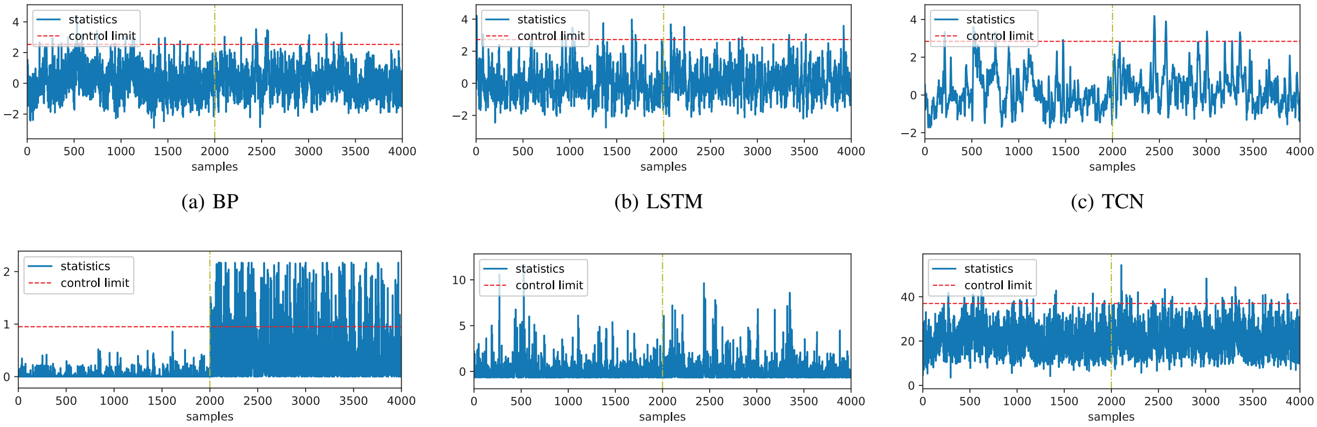


Fig. 5: The flowchart of OCMLFEM in experiment



(a) BP

(b) LSTM

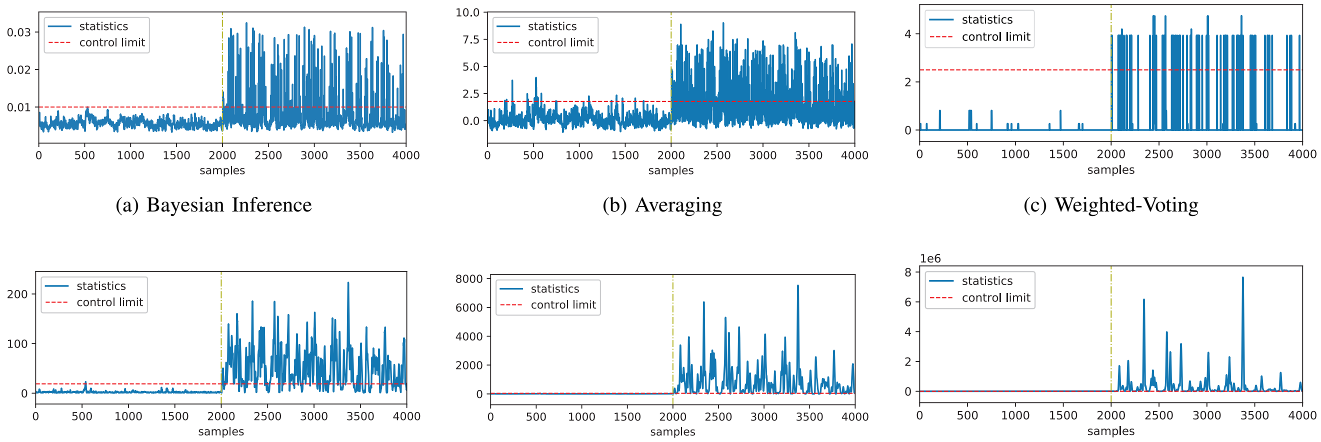
(c) TCN

(d) OCSVM

(e) iForest

(f) PCA(T^2)

Fig. 6: Detection performances of basic detectors for fault 3



(a) Bayesian Inference

(b) Averaging

(c) Weighted-Voting

(d) OCMLFEM $l^{max} = 1$

(e) OCMLFEM $l^{max} = 2$

(f) OCMLFEM $l^{max} = 3$

Fig. 7: Detection performance of ensemble detectors for fault 3

methods and a feature ensemble strategy. Five one-class machine learning models are adopted to construct the basic detectors without fault data and applied to original data for feature extraction. Then, a deep feature ensemble strategy is utilized to integrate the information obtained by basic detectors, which achieves better performance than any basic detector. OCMLFEM not only inherits the characteristics of nonlinear transformation from machine learning, but also mines sufficient process information through deep feature ensemble. Finally, the validity and superiority of OCMLFEM is demonstrated by TEP, such as faults 3, 9 and 15. It's worth to notice that the feature transformer layer of OCMLFEM is simple. Thus, the extraction method or feature transformation structure can be improved to form further effective model in future work.

References

[1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 293–311, 2003.

[2] M. Wang, D. Zhou, and M. Chen, "Hybrid variable monitoring:an unsupervised process monitoring framework with binary and continuous variables," *Automatica*, vol. 147, p. 110670, 2023.

[3] M. Wang, D. Zhou, and M. Chen, "Hybrid variable monitoring mixture model for anomaly detection in industrial processes," *IEEE T. Cybern.*, early access, 2023.

[4] D. Miljkovi, "Fault detection methods: A literature survey," in *2011 Proc. 34th Int. Conv. MIPRO*, 2011, pp. 750–755.

[5] M. Wang, D. Zhou, and M. Chen, "Adjustable multimode monitoring with hybrid variables and its application in a thermal power plant," *IEEE Trans. Ind. Inform.*, vol. 19, no. 2, pp. 1425–1435, 2023.

[6] Y. Yang, S. X. Ding, and L. Li, "Parameterization of nonlinear observer-based fault detection systems," *IEEE Trans. Automat. Contr.*, vol. 61, no. 11, pp. 3687–3692, 2016.

[7] M. Wang, D. Zhou, and M. Chen, "Anomaly monitoring of nonstationary processes with continuous and two-valued variables," *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 53, no. 1, pp. 49–58, 2023.

[8] M. Zhong, T. Xue, Y. Song, S. X. Ding, and E. L. Ding, "Parity space vector machine approach to robust fault detection for linear discrete-time systems," *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 51, no. 7, pp. 4251–4261, 2021.

[9] M. Wang, D. Zhou, and M. Chen, "Recursive hybrid variable monitoring for fault detection in nonstationary industrial processes," *IEEE Trans. Ind. Inform.*, vol. 18, no. 10, pp. 7296–7304, 2022.

[10] S. Joe Qin, "Statistical process monitoring: basics and beyond," *J. Chemom.*, vol. 17, no. 8-9, pp. 480–502, 2003.

[11] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 30, no. 1, pp. 179–196, 1995.

[12] J.-M. Lee, C. Yoo, and I.-B. Lee, "Statistical process monitoring with independent component analysis," *J. Process Control*, vol. 14, no. 5, pp. 467–485, 2004.

[13] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *J. Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[14] M. Xue, H. Yan, M. Wang, H. Shen, and K. Shi, "Lstm-based intelligent fault detection for fuzzy markov jump systems and its application to tunnel diode circuits," *IEEE Trans. Circuits Syst. II-Express Briefs*, vol. 69, no. 3, pp. 1099–1103, 2022.

[15] H. Zhang, B. Ge, and B. Han, "Real-time motor fault diagnosis based on tcn and attention," *Machines*, vol. 10, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2075-1702/10/4/249

[16] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt *et al.*, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, no. 3, pp. 582–588, 2000.

[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 1–39, 2012.

[18] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.

[19] W. Shao and X. Tian, "Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models," *Chem. Eng. Res. Des.*, vol. 95, pp. 113–132, 2015.

[20] D. Liu, M. Wang, and M. Chen, "Feature ensemble net: A deep framework for detecting incipient faults in dynamical processes," *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8618–8628, 2022.

[21] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE-ASME Trans. Mechatron.*, vol. 23, no. 1, pp. 89–100, 2017.

[22] J. Li, J.-h. Cheng, J.-y. Shi, and F. Huang, "Brief introduction of back propagation (bp) neural network algorithm and its improvement," in *Advances in Computer Science and Information Engineering: Volume 2*. Springer, 2012, pp. 553–558.

[23] P. Phaladiganon, S. B. Kim, V. C. Chen, and W. Jiang, "Principal component analysis-based control charts for multivariate nonnormal distributions," *Expert Syst. Appl.*, vol. 40, no. 8, pp. 3044–3054, 2013.

[24] I. Ionita-Laza, K. McCallum, B. Xu, and J. D. Buxbaum, "A spectral approach integrating functional genomic annotations for coding and noncoding variants," *Nature Genet.*, vol. 48, no. 2, pp. 214–220, 2016.

[25] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *J. Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.