


Article

Industrial Fault Detection Based on Discriminant Enhanced Stacking Auto-Encoder Model

Bowen Liu ^{1,2} , Yi Chai ^{1,2,*}, Yutao Jiang ^{1,2} and Yiming Wang ^{1,2} ¹ College of Automation, Chongqing University, Chongqing 400044, China² State Key Laboratory of Power Transmission Equipment and System Security and New Technology, Chongqing University, Chongqing 400044, China

* Correspondence: chaiyi@cqu.edu.cn

Abstract: In the recent years, deep learning has been widely used in process monitoring due to its strong ability to extract features. However, with the increasing layers of the deep network, the compression of features by the deep model will lead to the loss of some valuable information and affect the model's performance. To solve this problem, a fault detection method based on a discriminant enhanced stacked auto-encoder is proposed. An enhanced stacked auto-encoder network structure is designed, and the original data is added to each hidden layer in the model pre-training process to solve the problem of information loss in the feature extraction process. Then the self-encoding network is combined with spectral regression kernel discriminant analysis. The fault category information is introduced into the features to optimize the features and enhance the discrimination of the extracted features. The Euclidean distance is used for fault detection based on the extracted features. From the Tennessee Eastman process experiment, it can be found that the detection accuracy of this method is about 9.4% higher than that of the traditional stacked auto-encoder method.

Keywords: deep learning; SAE; spectral regression kernel discriminant analysis; fault detection



Citation: Liu, B.; Chai, Y.; Jiang, Y.; Wang, Y. Industrial Fault Detection Based on Discriminant Enhanced Stacking Auto-Encoder Model. *Electronics* **2022**, *11*, 3993. <https://doi.org/10.3390/electronics11233993>

Academic Editor: Marcin Witeczak

Received: 7 October 2022

Accepted: 30 November 2022

Published: 2 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The improvement of science and continuous growth of the economy causes increasing scale of the process industry and closed coupling degree so that the abnormalities and faults of any equipment in system such as power system and industrial equipment will affect the system's normal operation [1–3]. Therefore, fault detection should run during the whole processing industry, it is an indispensable part of process monitoring and plays a vital role to ensure safety and stable operation of the whole process industrial system.

Fault detection methods are mainly divided into model-based methods, knowledge-based methods, and data-based methods. Among them, model-based methods and knowledge-based methods detect faults by establishing a system operation mechanism analysis model and an expert knowledge base. However, the complex structure and large scale of modern process industry increase the difficulty to acquire mechanism knowledge and establish a mechanism model, which brings significant challenges to detect faults of processing industry systems [4]. Therefore, traditional analytical model-based methods and knowledge-based methods are often unable to establish an accurate analytical model and expert knowledge model, and it is difficult to monitor the system in real-time effectively.

With the development of modern detection and sensing technology, sensor signals such as voltage, current, flow and pressure in industrial systems can be collected with large scale [5–8]. Therefore, fault detection methods based on data-driven have developed rapidly, such as principal component analysis (PCA), linear discriminant analysis (LDA), independent component analysis (ICA), and other multivariate statistical methods that have been widely investigated by different scholars [9–11]. However, variables of industrial

process often provide more complex correlations and nonlinearity. Using linear methods will lead to unsatisfactory results. Therefore, many researchers combined kernel methods with traditional multivariate statistical methods to deal with nonlinear problems, such as kernel principal component analysis (KDA) [12,13].

However, kernel method needs to artificially determine the kernel width, which means kernel methods are still unable to solve the nonlinear problem fundamentally. Moreover, the above multivariate statistical methods are relatively shallow while extracting features from large data. Some useful information will be submerged, which affects the detected performance [14]. Comparatively, deep learning provides more expressive performance, which can extract more useful nonlinear features for highly complex systems. Therefore, after the deep learning method was proposed in 2006, it received extensive attention from scholars [15,16]. Deep learning can deeply extract nonlinear features of data by using deep model composed of multiple processing layers and nonlinear activation functions. The typical methods include convolutional neural networks (CNN), deep belief networks (DBN), and stacked auto-encoder (SAE) [17–19]. These methods extract high-dimensional feature information from data layer by layer through its complex network structure, which can describe the original object more accurately.

In recent years, deep learning has been widely used in fault detection and diagnosis. Chen et al. redefined the objective function of nonlinear canonical correlation analysis (CCA) and gave a generalized solution. Then, the introduced method combined with a unilateral neural network. Consequently, a fault diagnosis method based on Single-Side CCA was proposed to improve the diagnostic performance [20]. Yuan et al. proposed an entropy-weighted nuisance attribute projection method to eliminate the interference information in the feature space and then combined it with neural network to diagnose fault of rolling bearings [21]. In addition, the researchers proposed a multivariate intrinsic multiscale entropy to analyze the irregularity and complexity of the Lorentz signal of bolted joints and constructed a convolutional neural network to achieve monitoring of bolted joints [22]. Liu et al. proposed a highly sensitive feature selection framework based on DBN and implemented fault detection on this framework. This method eliminated the redundant features in the DBN network and improved the fault detection performance [23]. Wang et al. proposed a stacked supervised auto-encoder to obtain deep features for fault classification and applied it to industrial process fault diagnosis [24]. To decrease the difficulty of label acquisition, Ma et al. proposed a consistency regularization auto-encoder framework based on an encoder-decoder network to realize fault classification [25]. Huang et al. proposed a distributed variational auto-encoder fault detection method. By establishing DVAE models for local and neighboring units of system to describe the complex relationship between each unit, the detected effect under missing data is improved [26]. Jiang et al. also established SAE to mine the relevant features between process variables, and detected the sample state through two-dimensional nonlinear and dynamic information to realize the fault diagnosis of nonlinear batch process [27]. Although deep learning can deeply extracts features that is applied to detect and diagnose faults, it still has some drawbacks. According to the information bottleneck theory, the depth of the network model becomes deeper, the model's compression of features will lead to the loss of information in the original data, resulting in a decline in the performance of the fault detection model [28].

This paper proposes a fault detection method based on discriminant enhance SAE (DESAE) to extract information in the original data effectively. Constructing an enhanced SAE (ESAE) network will reduce the loss of valuable information in the feature extraction process. In the pre-training process. This method combines original data with hidden layer's features of the network and stacks multiple enhanced AE networks to mine valuable information in the data layer by layer. Then, the introduced method combines ESAE with spectral regression kernel discriminant analysis (SRKDA), and adds the fault category information, into the features. additionally, and the features are optimized to enhance the discrimination of the extracted features. Finally, the Euclidean distance and sliding

window function are used for fault detection to eliminate the influence of noise on detection statistics. The main contributions of this paper are as follows:

1. SAE can learn the nonlinear features of data. However, when the deep model performs feature extraction, some valuable information in the original data may be lost due to the deeper network layers, which affects detected performance. In this paper, an ESAE network is constructed, which adds the original data into the hidden layer of the pre-training process and reduces the loss of valuable information in the feature extraction process.
2. Since SAE network is an unsupervised learning model, its training process only considers the reconstructed data's global error information and lacks the guidance of labels. In this paper, ESAE and SRKDA are combined to introduce fault category information into features and optimize features to enhance the discrimination of features and improve the performance of detection models.
3. Based on the characteristics obtained by the above model, Euclidean distance is used to measure the difference between normal data and fault data, and the sliding window function is applied to reduce the influence from noise interference on detection statistics. Considering the unknown distribution of actual industrial data, kernel density estimation (KDE) is implemented to design the control limitation of statistics.

The remainder of paper structure is shown as follow: Section 2 simply introduced SAE. In Section 3 described monitoring process by using DESAE. Section 4 presented experiment that implementing introduced algorithm into TE process and concluded in Section 5.

2. Stacked Auto-Encoder

Auto-encoder (AE) is a standard feed-forward neural network. The network input to the hidden layer can be regarded as an encoding process. Moreover, from the hidden layer to the output layer is considered a decoding process. The encoded data will be reconstructed to the original data by decoding process. The AE minimizes the mean square error (MSE) of input and output by training network to achieve data feature extraction. The feature extraction of AE usually provides three different manifestations: Firstly, the nodes from hidden layer are less than from input layer and output layer, and the extracted features are the compressed dimension reduction representation of the training data; secondly, the hidden layer nodes are larger than the input layer and output layer, and the extracted features are the high-dimensional representation of training data; Thirdly, the hidden layer node is equal to the input node, the feature is an equal-dimensional representation of training data. The encoding and decoding process of AE is:

$$\begin{cases} \mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \\ \hat{\mathbf{x}} = \sigma(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2) \end{cases} \quad (1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_m]^T \in \mathbf{R}^m$ and $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m]^T \in \mathbf{R}^m$ are the input layer and output layer of the AE network respectively. $\mathbf{h} = [h_1, h_2, \dots, h_v] \in \mathbf{R}^v$ is the hidden layer of the network. $\mathbf{W}_1 \in \mathbf{R}^{v \times m}$ and $\mathbf{b}_1 \in \mathbf{R}^v$ are the weight matrix and offset vector of hidden layer. Similarly, $\mathbf{W}_2 \in \mathbf{R}^{m \times v}$ and $\mathbf{b}_2 \in \mathbf{R}^m$ are the weight matrix and bias vector of the output layer, respectively. $\sigma(\cdot)$ is the neuron activation function, generally using sigmoid and tanh functions.

SAE is a deep learning model stacked by auto-encoders. It deletes the decoding part after training the AE and inputs the hidden layer parameter obtained by the first AE into the second AE for training to obtain a new feature representation. After repeating the above steps several times, the desired SAE model can be obtained. The model structure is shown in Figure 1. After the network pre-training process, the corresponding hidden layers of each AE are stacked to form a deep auto-encoder network with multiple hidden layers. Similar to traditional AE, the parameters of the network are fine-tuned using the backpropagation method by calculating the error between the network output layer and the input layer. As

a typical deep learning network, SAE has better feature extraction ability than AE, and its pre-training steps can avoid problems such as over-fitting of network training.

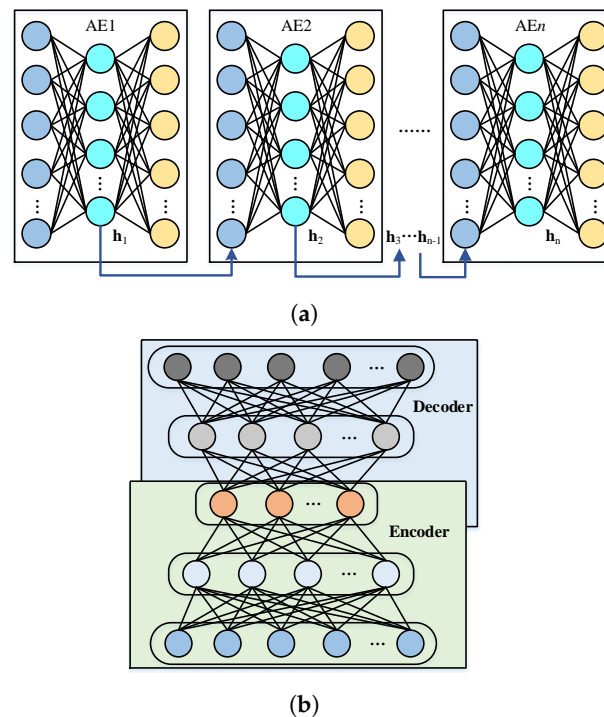


Figure 1. SAE Network Structure Diagram: (a) Schematic diagram of SAE pre-training stage; (b) Schematic diagram of SAE fine-tuning stage.

3. Process Monitoring Method Based on Discriminant Enhanced SAE

3.1. Enhanced SAE

Based on its underlying concept, SAE can effectively extract the deep features in the data and has the characteristics of fast convergence. However, according to the information bottleneck theory, ability of feature extraction by can still be improved. As the depth of the neural network increases, the relevant information of the features extracted by the network and the original data will decrease. Therefore, in order to retain more original data information in the process of feature extraction, this paper proposes an ESAE network. The network trains the original data as additional information of the hidden layer in the pre-training stage. It can make the original data fully participate in the coding process so that more information related to the original data can be retained in the feature extraction process. Its network structure is shown in Figure 2.

ESAE is similar to SAE in the training process, divided into pre-training and reverse fine-tuning. In the pre-training, the hidden layer data of each AE is used as the input for the next AE. Moreover, the original data is added to enhance the training process, as shown in the blue circle in Figure 2. In the pre-training process, ESAE first inputs the original data \mathbf{x}_{data} into AE1 for training to obtain the hidden layer data $\mathbf{h}^{(1)}$ of the AE1. The training method is shown in Equation (1). Then, the hidden layer $\mathbf{h}^{(1)}$ of AE1 is combined with the original data as the input $\mathbf{x}^{(2)} = [\mathbf{h}^{(1)}, \mathbf{x}_{data}]$ of AE2. The AE2 trained in the same way as AE1 to obtain the hidden layer $\mathbf{h}^{(2)}$. Repeat the above steps to reach the depth of the model set. In the ESAE reverse fine-tuning, the network parameters obtained in the network pre-training are used as initialization parameters to construct a deep network with multiple hidden layers. The MSE between the input layer and the output layer is calculated as the global parameter in the cost function inverse fine-tuning network.

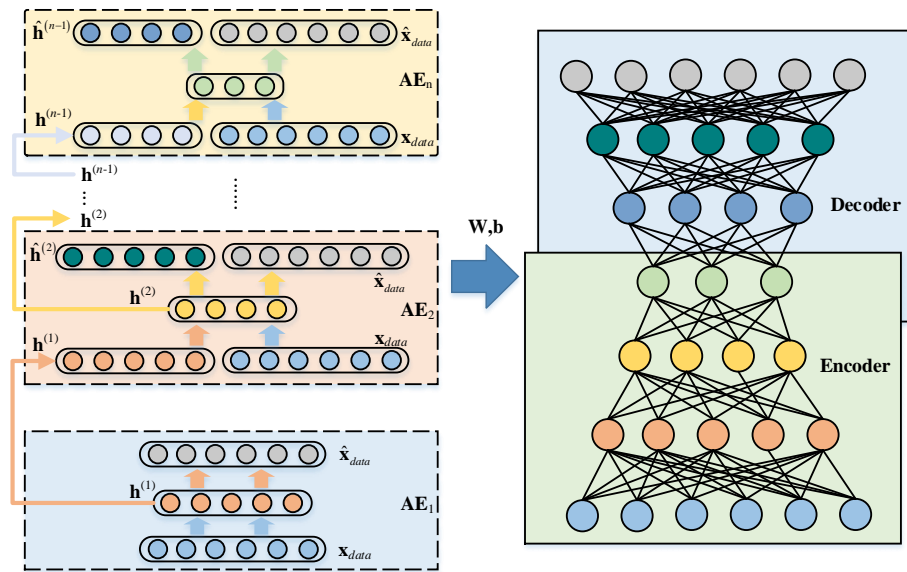


Figure 2. ESAE Network Structure Diagram.

3.2. Detection Model Based on Discriminant Enhanced SAE

ESAE can effectively extract the features of the data and freely select the feature dimension. However, since the algorithm is an unsupervised learning method, only the global reconstruction information of the data is considered, and the label information of the data is not involved. Therefore, this paper combines ESAE with SRKDA to effectively use the advantages of unsupervised learning and supervised methods to improve the feature extraction of process data.

Kernel discriminant analysis is a nonlinear extension of linear discriminant analysis. Its main purpose is to project linear discriminant analysis into high-dimensional feature space to solve the nonlinear problems. The feature extracted by ESAE is $\mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m] \in \mathbf{R}^n$, and there are c categories in the sample. The objective function of linear discriminant analysis is calculated as:

$$\mathbf{a}_{opt} = \arg \max \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}} \tag{2}$$

where S_b donates between-class covariance matrix and S_w denotes within-class covariance matrix in the feature space, which can be calculated as Equations (3) and (4) respectively.

$$S_b = \sum_{k=1}^N p \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu} \right) \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu} \right)^T \tag{3}$$

$$S_w = \sum_{k=1}^N p \left(\sum_{i=1}^{m_k} \left(\mathbf{h}_i^{(k)} - \boldsymbol{\mu}^{(k)} \right) \left(\mathbf{h}_i^{(k)} - \boldsymbol{\mu}^{(k)} \right)^T \right) \tag{4}$$

Maps the features $\mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m] \in \mathbf{R}^n$ to the feature space \mathcal{F} using nonlinear mapping ϕ :

$$\phi : \mathbf{R}^n \rightarrow \mathcal{F}$$

Then, between-class covariance matrix (S_b^ϕ), within-class covariance matrix (S_w^ϕ), and total scatter covariance matrix (S_t^ϕ) in the nonlinear feature space are calculated as:

$$S_b^\phi = \sum_{k=1}^c m_k \left(\boldsymbol{\mu}_\phi^{(k)} - \boldsymbol{\mu}_\phi \right) \left(\boldsymbol{\mu}_\phi^{(k)} - \boldsymbol{\mu}_\phi \right)^T \tag{5}$$

$$S_w^\phi = \sum_{k=1}^c \left(\sum_{i=1}^{m_k} \left(\phi \left(\mathbf{h}_i^{(k)} \right) - \boldsymbol{\mu}_\phi^{(k)} \right) \left(\phi \left(\mathbf{h}_i^{(k)} \right) - \boldsymbol{\mu}_\phi^{(k)} \right)^T \right) \tag{6}$$

$$S_t^\phi = \sum_{i=1}^m (\phi(\mathbf{h}_i) - \boldsymbol{\mu}_\phi) (\phi(\mathbf{h}_i) - \boldsymbol{\mu}_\phi)^T \tag{7}$$

where $\boldsymbol{\mu}_\phi$ is the mean vector of the whole sample in the feature space \mathcal{F} , $\boldsymbol{\mu}_\phi^{(k)}$ is the mean vector of the k-class sample in the feature space \mathcal{F} .

Let \mathbf{v} be the projection function of the feature space. The objective function in the feature space can be expressed as:

$$\mathbf{v}_{opt} = \arg \max \frac{\mathbf{v}^T S_b^\phi \mathbf{v}}{\mathbf{v}^T S_t^\phi \mathbf{v}} \tag{8}$$

Transform Equation (8) into solving eigenvalue problem:

$$S_b^\phi \mathbf{v} = \lambda S_t^\phi \mathbf{v} \tag{9}$$

Since the eigenvectors are linear combinations of $\phi(\mathbf{h}_i)$, there exist coefficient α_i such that

$$\mathbf{v} = \sum_{i=1}^m \alpha_i \phi(\mathbf{h}_i) \tag{10}$$

Let $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_m]^T$, the objective function can be rewritten as:

$$\boldsymbol{\alpha}_{opt} = \arg \max \frac{\boldsymbol{\alpha}^T \mathbf{K} \mathbf{W} \mathbf{K} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{K} \mathbf{K} \boldsymbol{\alpha}} \tag{11}$$

Similarly, transform Equation (11) into solving eigenvalue problem:

$$\mathbf{K} \mathbf{W} \mathbf{K} \boldsymbol{\alpha} = \lambda \mathbf{K} \mathbf{K} \boldsymbol{\alpha} \tag{12}$$

where $K_{ij} = k(\mathbf{h}_i, \mathbf{h}_j)$ is the kernel matrix. \mathbf{W} can be defined by the following equation:

$$W_{ij} = \begin{cases} \frac{1}{m_k}, & \text{if } \mathbf{h}_i \text{ and } \mathbf{h}_j \text{ belong to the } k\text{th class} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Simplify the eigenvalue problem in Equation (12) by using spectral regression technique, we have

$$\mathbf{W} \mathbf{y} = \lambda \mathbf{y} \tag{14}$$

Then

$$\mathbf{K} \mathbf{W} \mathbf{K} \boldsymbol{\alpha} = \mathbf{K} \mathbf{W} \mathbf{y} = \mathbf{K} \lambda \mathbf{y} = \lambda \mathbf{K} \mathbf{K} \boldsymbol{\alpha} \tag{15}$$

where $\mathbf{y} = (\mathbf{K} + \delta \mathbf{I}) \boldsymbol{\alpha}$, \mathbf{I} is the unit matrix, $\delta \geq 0$ is a regularization parameter. In the feature space, the projection function can be expressed as:

$$f(\mathbf{h}) = \langle \mathbf{v}, \phi(\mathbf{h}) \rangle = \sum_{i=1}^m \alpha_i \mathbf{K}(\mathbf{h}, \mathbf{h}_i) \tag{16}$$

After obtaining the projection matrix, the latent variable to the new sample \mathbf{x}_{new} are calculated as:

$$\mathbf{T}_{new} = \mathbf{K}(\mathbf{h}, \mathbf{h}_{new}) \cdot \mathbf{v} \tag{17}$$

In the projection space, the average value of normal samples is used as the reference value. Euclidean distance measures the difference between the test sample and the reference value as the detection statistics. The calculation method is as follows:

$$\mathbf{T}_d = \sqrt{\sum_{i=1}^k (\mathbf{T}_{new}^i - \mathbf{T}_{aver}^i)^2} \quad (18)$$

where \mathbf{T}_{new}^i is the i -dimensional feature of the new sample, and \mathbf{T}_{aver}^i represents the average value of the i -dimensional feature corresponding to the normal sample.

Moving average window is often used in process control to avoid the impact of sudden noise in the measurement variables on the results as much as possible. In addition, the method can consider the influence of the previous state on the current state, smooth the statistics, and reflect the actual situation of the current statistics. Through the calculation of the moving average window, the detection statistics can be calculated as:

$$Statistics(n) = \frac{\mathbf{T}_d(n - N + 1) + \dots + \mathbf{T}_d(n - 1) + \mathbf{T}_d(n)}{N} \quad (19)$$

where N is the width of the sliding window.

Since the statistic is not limited by specific distribution, the control limit of the statistic is calculated by using KDE. The Gaussian function is selected as the kernel function, and the KDE function can be calculated as:

$$p(x) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^n \exp\left(-\frac{(x - x_i)^2}{2h^2}\right) \quad (20)$$

where $h = (4\hat{\sigma}^5/3n)^{1/5} \approx 1.06\hat{\sigma}n^{1/5}$, $\hat{\sigma}$ is the standard deviation of the sample. The cumulative distribution function can be calculated as:

$$F(a) = P(x \leq a) = \int_{-\infty}^a f(x)dx = 1 - \alpha \quad (21)$$

where α is the significance level. The control theory of fault detection can be computed by Equation (21). When the detection statistics of the detected sample exceed the control theory, the system is judged to be in fault state.

3.3. Detection Process

The method proposed in this paper mainly consists of offline training and online detection, as shown in Figure 3. Its main steps are as follows:

Offline training:

1. The historical data are collected as training samples and standardized.
2. Input standardized data into ESAE to extract representative features \mathbf{h} .
3. The feature \mathbf{h} is projected into the feature space \mathcal{F} to obtain the feature vector $\phi(\mathbf{h})$.
4. The S_b^ϕ , S_w^ϕ and S_t^ϕ of feature \mathbf{h} are calculated respectively.
5. The generalized feature solving problem is transformed into a regression framework to solve the projection function Equation (16).
6. The KDE is used to calculate the control limit of detection statistics.

Online detection:

1. Collect online data as the testing samples. The testing samples are standardized with the same mean and variance as the training samples.
2. Obtaining the features \mathbf{h}_{new} of standardized testing samples.
3. The feature \mathbf{h}_{new} is projected into the feature space \mathcal{F} to obtain the feature vector $\phi(\mathbf{h}_{new})$.
4. Calculate the Test set feature projection by Equation (17)
5. Calculate detection statistics by Euclidean distance and sliding window function.
6. Calculate the fault detection rate if a fault is detected.

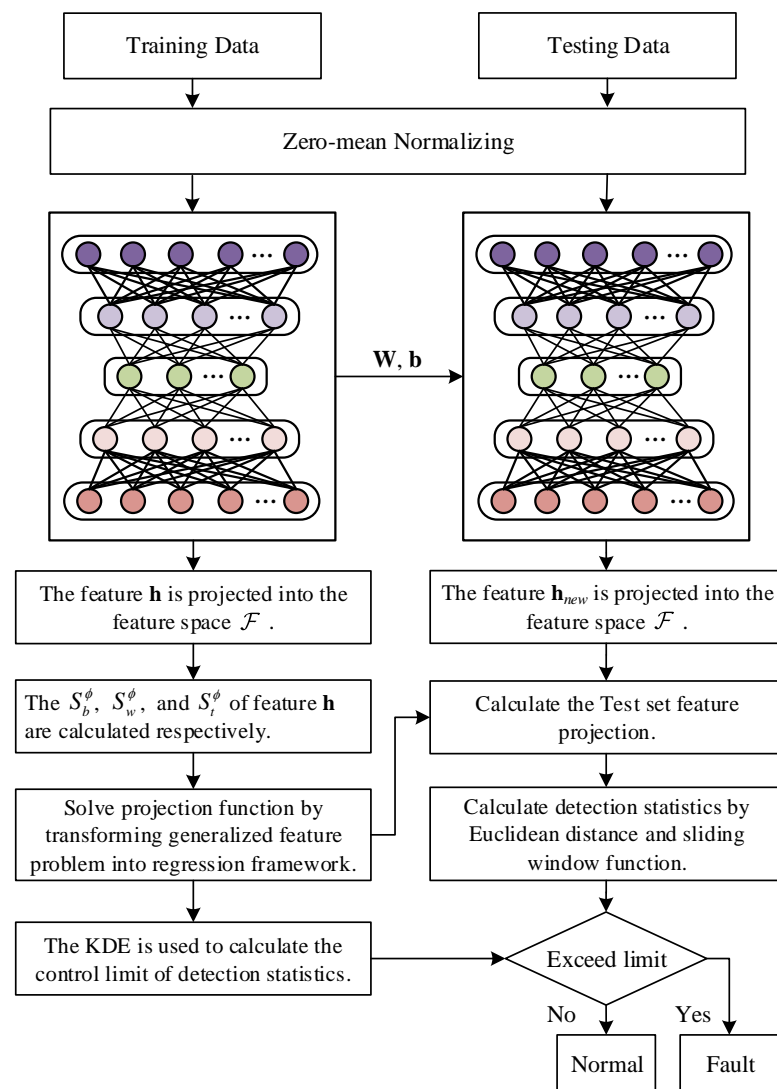


Figure 3. Detection flowchart of DESAE.

4. Case Study

4.1. TE Process

TE process, the simulated system of actual industrial system that was implemented, was applied to verify the effectiveness of proposed method. This system was widely employed in process control, multivariate statistical fault detection and diagnosis [29]. TE process mainly includes five central operating units such as condenser, compressor, separator, continuous stirring-type reactor, and stripping tower. Its process flow chart was shown on Figure 4.

TE process contained 41 process variable and 12 control variable (XMV1 to XMV12). The process variable contained 19 component variable and 22 measured variable (XMEAS1 to XMEAS22). When monitoring TE process with constant mixing speed at 50% and sampled data under normal working condition and fault working condition. Then, 21 fault datasets under 21 fault working conditions were obtained, as shown in Table 1. Every sampling dataset contained 960 samples. Besides, faults would be injected in the 161st sampling point. The fault detection rate (FDR) and average fault alarm rate (AvFAR) was regarded as evaluation standard, which were calculated by Equations (22) and (23) separately.

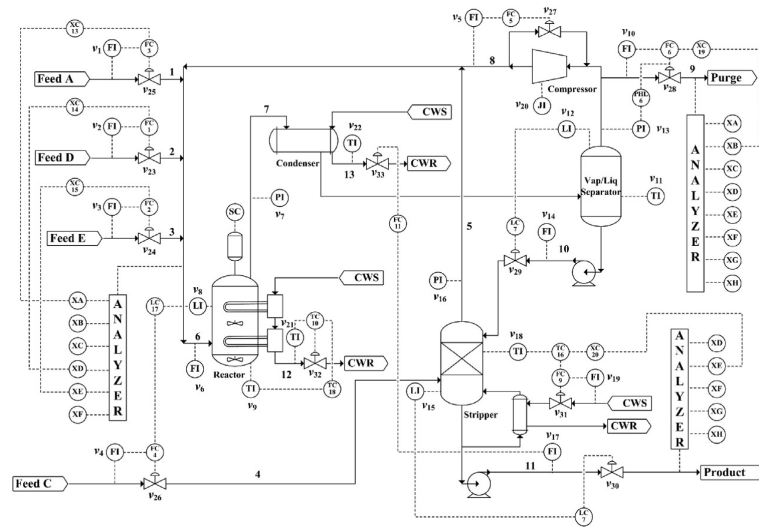


Figure 4. Tennessee Eastman process flowchart.

Table 1. Faults in the Tennessee Eastman process.

Number	Disturbances	Type
1	A/D feed ratio changes, B composition constant	Step
2	B feed ratio changes, A/D composition constant	Step
3	D feed temperature changes	Step
4	Reactor cooling water inlet temperature changes	Step
5	Condenser cooling water inlet temperature changes	Step
6	A feed loss	Step
7	C head pressure loss	Step
8	A/B/C composition changes	Random
9	D feed temperature changes	Random
10	C feed temperature changes	Random
11	Reactor cooling water inlet temperature changes	Random
12	Separator cooling water inlet temperature changes	Slow drift
13	Reactor dynamic constants changes	Slow drift
14	Reactor valve	Sticking
15	Separator valve	Sticking
16	Unknown	Unknown
17	Unknown	Unknown
18	Unknown	Unknown
19	Unknown	Unknown
20	Unknown	Unknown
21	Stable valve in stream 4	Constant

$$FDR(i) = \frac{\text{The number of faults data detected correctly in fault } i}{\text{The number of faults data}} \tag{22}$$

$$AvFAR(i) = \frac{\text{The total number of normal data detected wrongly}}{\text{The number of normal data}} \tag{23}$$

In the TE process. Fault3, 9 and 15 were ignored while analyzing the result of process monitoring, because their mean, variance and peak value had not changed obviously [30–32]. While monitoring process, confidence limitation of detection statistics was set to be 0.99. DESAE algorithm’s performance was verified by comparing performance of monitoring results by using PCA, KPCA, GLPP and SAE. The FDR was shown in Table 1 and AvFAR was shown in Table 2. According to Tables 2 and 3, using proposed method to monitor process provided higher FDR than using other compared methods, especially while detecting Fault6, 7, 11, 12, 17, 18, 19.

Table 2. FDR of PCA, KPCA, GLPP, SAE and DESAE on TE process.

	PCA		KPCA		GLPP		SAE	DESAE
	T^2	SPE	T^2	SPE	T^2	SPE		
Fault1	99.25	99.75	99.75	99.5	99.5	99.75	99.25	99.63
Fault2	98.5	98.63	98.38	98.5	97.62	98.75	98.75	98.63
Fault4	27.75	100	99.88	97.88	15.75	97.38	90.75	99.88
Fault5	23.75	33.25	31.5	29.13	100	32.13	100	99.87
Fault6	98.75	100	99.75	100	100	99.5	100	100
Fault7	100	100	100	100	54.62	100	100	100
Fault8	97.25	96.87	98.62	97.62	98	97.75	96.13	97.75
Fault10	25.37	44.5	53.87	56.5	92	54.87	22.25	77.5
Fault11	45	69.12	74.12	70.13	26	74.12	74	86.38
Fault12	98.37	95.13	99.5	98.88	99.87	99	97.13	99.75
Fault13	94.5	95.13	94.87	94.5	94.5	95.75	92.37	95.25
Fault14	98.88	99.88	100	100	94.25	100	99.88	99.87
Fault16	11	43.25	38.37	49.5	94.5	39.63	15.25	86.75
Fault17	75.88	95.63	95.5	92	85.12	93	88	96.5
Fault18	89	90.12	90.63	89.88	89.87	90.87	89.25	91.25
Fault19	6.25	20.38	27	17.25	89.88	20.38	13.5	95.38
Fault20	27.87	55.75	63.5	53.13	91	61	42.13	80.87
Fault21	39.5	49.75	43.63	47	58.37	42	31.38	45.25
avFDR	62.27	77.06	78.27	77.3	82.27	77.55	82.31	91.7

While considering the fault alarm rate, there is no obvious change in the data of Fault16 in the observation, detecting Fault16 is often less effective than detecting other faults [33,34]. Fault16 sacrifices part of the false alarm rate, thereby improving the detection rate. As shown in Table 2, the average false alarm rate of this algorithm is not optimal, but it is within the acceptable range.

Table 3. AvFAR of PCA, KPCA, GLPP, SAE and DESAE on TE process.

	PCA		KPCA		GLPP		SAE	DESAE
	T^2	SPE	T^2	SPE	T^2	SPE		
avFAR	0.42	3.44	3.64	1.42	2.3	3	2.08	1.5

Fault5 will happen if there exist step changes in cooling water's temperature of the condenser's entrance. Simultaneously, the impulse response will happen on the exit of the temperature isolator. The negative feedback system will cause some variables to return to normal if faults had happened several times. However, the temperature of the cooling water is still abnormal, which means faults had still existed. Fault5 was shown in Figure 5. There will exist faults in the 161st sampled data and some variables will return to the normal state in the 350th sampled data. According to Figure 5, PCA and GLPP methods in the figure cannot correctly identify faults under the system negative feedback mechanism. While the proposed method can effectively reflect the failure of the condenser cooling water inlet temperature after the negative feedback is stabilized.

If feed C was fed into the inlet of stripper, its temperature will influence change of working condition in the stripper. Consequently, Fault10 happened. Nevertheless, the temperature change will affect the production of the stripper and most methods can only monitor the change of temperature without considering other fault variables, which degrade the performance of detection. According to Figure 6 which demonstrates performance of detecting Fault10, PCA, GLPP and SAE methods all have obvious false positives in the detection process. While DESAE provided effective detection performance of detection results, there is no false alarm.

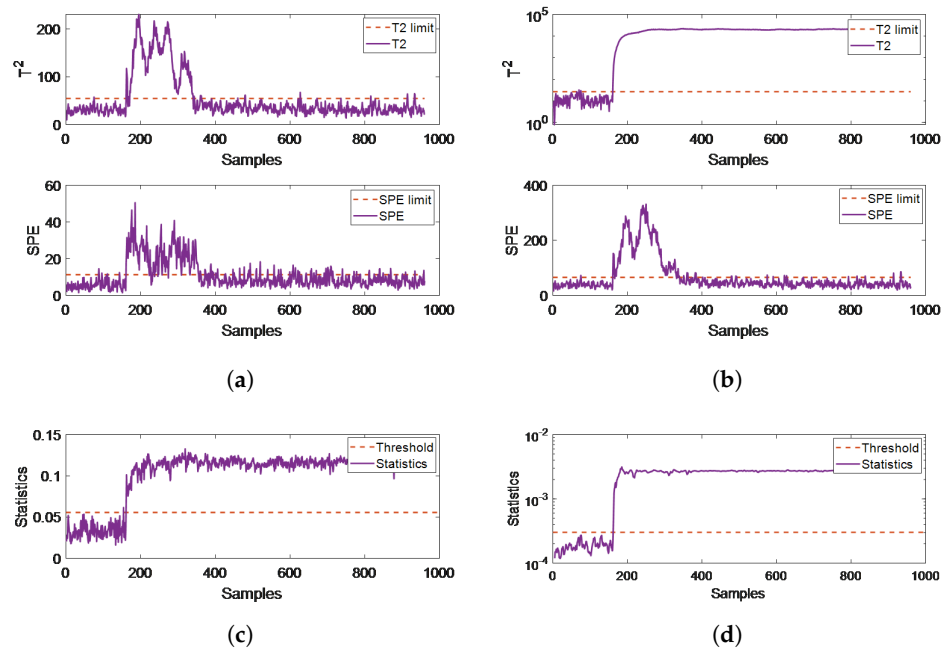


Figure 5. Comparison of detection results for Fault5: (a) Fault detection by PCA; (b) Fault detection by GLPP; (c) Fault detection by SAE; (d) Fault detection by DESAE.

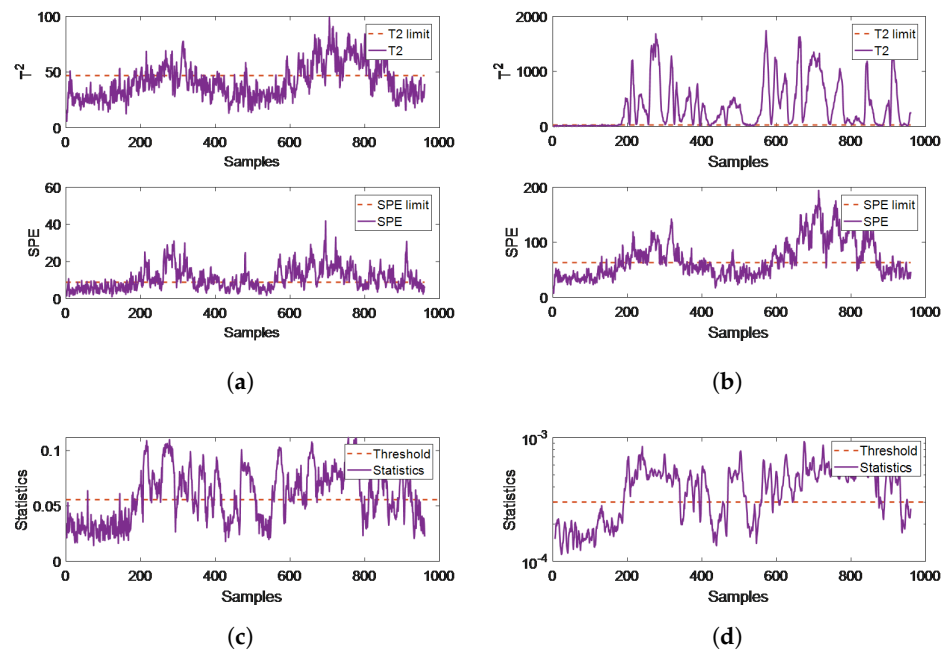


Figure 6. Comparison of detection results for Fault10: (a) Fault detection by PCA; (b) Fault detection by GLPP; (c) Fault detection by SAE; (d) Fault detection by DESAE.

Fault19 is an unknown fault which will not influence results of process variable. The adjusted threshold would allow the change of variable [35]. Figure 7 shows the detection results of Fault19. It can be seen from the figure that PCA, GLPP, and SAE algorithms do not extract some essential features of Fault19 well due to their unsupervised characteristics, so these algorithms cannot effectively detect Fault19. The DESAE proposed combines unsupervised and supervised characteristics to effectively extract the essential features so that the detection performance is further improved.

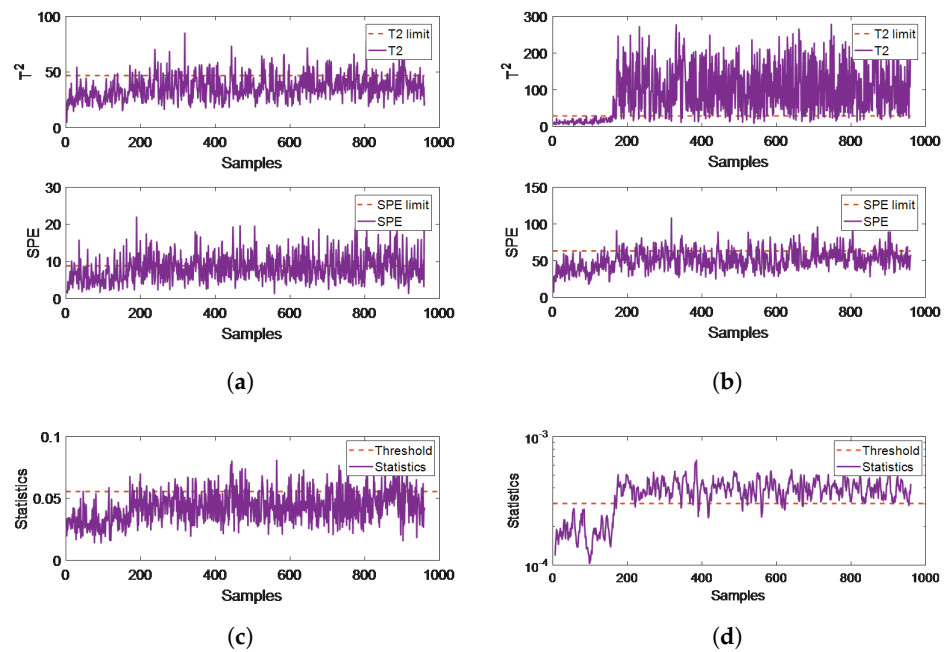


Figure 7. Comparison of detection results for Fault19: (a) Fault detection by PCA; (b) Fault detection by GLPP; (c) Fault detection by SAE; (d) Fault detection by DESAE.

4.2. Feature Visualization Analysis Based on t-SNE

Since the DESAE algorithm introduces label information to enhance the feature expression ability in the process of feature extraction. Therefore, this section will use t-distributed stochastic neighbor embedding (t-SNE) to visualize the features proposed by the algorithm to observe the enhancement effect of label information on features.

According to different fault types such as step, slow change, viscosity and unknown faults, Fault1, 2, 5, 6, 7, 13, 14, 19 are selected for the experiment. The visualization result of extracting features by t-SNE is shown in Figures 8 and 9. According to Figure 8, original data is mixed up between the variables without feature extraction. In addition, visualization result of extracting features by PCA, GLPP, SAE and the proposed method are shown in Figure 9a–d. As shown in Figure 9, the visualization result of extracting features by compared method provided better performance of separation than result of original data. However, fault types cannot be shown in the features for the reason that PCA, GLPP and SAE are unsupervised learning algorithm. Consequently, the visualization results are still mixed. However, features of fault type were extracted by adding supervised part into the DESAE algorithm, which provides better performance of monitoring.

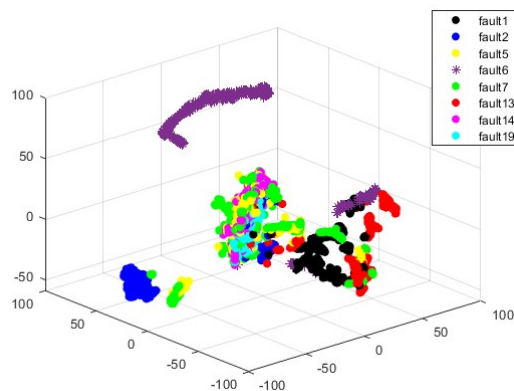


Figure 8. The t-SNE visualization results of Original data.

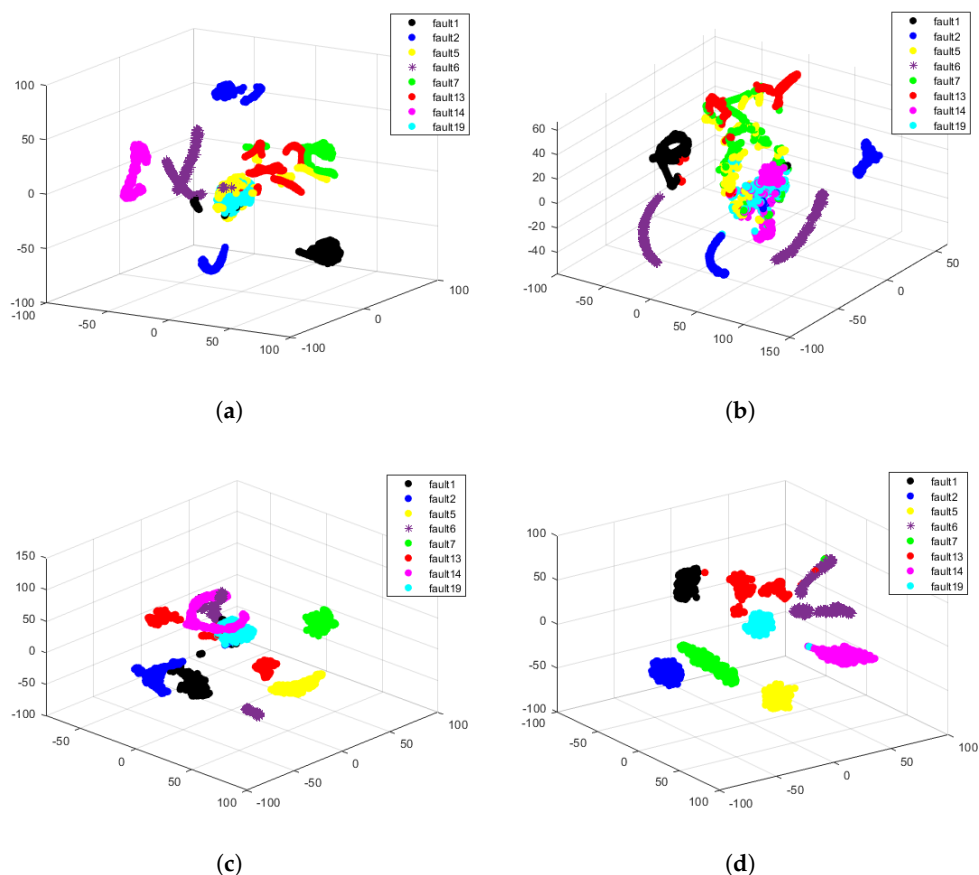


Figure 9. Comparison of t-SNE visualization results of fault features: (a) Visualization results of PCA extracted features; (b) Visualization results of GLPP extracted features; (c) Visualization results of SAE extracted features; (d) Visualization results of DESAE extracted features.

5. Conclusions

This paper considers the problem that nonlinear features of monitoring data in the process industry are difficult to extract and some valuable information is lost. An industrial fault detection method based on DESAE is proposed. During the process of pre-training model, the features in the hidden layer of AE network are combined with the original data as the input of the next AE network, and the deep features in the data are extracted layer by layer to solve the problem of valuable information loss in the feature extraction process. In addition, the proposed method uses the SRKDA method to optimize features extracted by ESAE using label information to make the features discriminative. Finally, the Euclidean distance between the features that needs to be measured and normal features are calculated to detect faults. The sliding window function is applied to eliminate noise interference on the monitoring statistics, which improves the detected effect.

The DESAE method is applied to the TE process for verification, and the average fault detection rate is 91%. The discrimination of features is verified by t-SNE visualization. The experimental results show that the detection rate of this method is 9.4% higher than that of traditional SAE. In addition, because DESAE can extract high-order features of monitoring data, this method can also detect other types of equipment in the process industry, such as open circuit and short circuit fault detection of converters in power systems.

Author Contributions: Funding acquisition, Y.C.; project administration, B.L., Y.C.; conceptualization, B.L., Y.C., Y.J.; validation, Y.W.; formal analysis, B.L., Y.J.; investigation, B.L., Y.W.; data curation, Y.J., Y.W.; writing—original draft preparation, B.L.; writing—review and editing, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Key Research and Development Project (2019YFB2006603). This work was also funded by the National Natural Science Foundation of China (U2034209). This research was supported by Shandong Provincial Natural Science Foundation (ZR2022QF031).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xu, S.; Tao, S.; Zheng, W.; Chai, Y.; Ma, M.; Ding, L. Multiple open-circuit fault diagnosis for back-to-back converter of PMSG wind generation system based on instantaneous amplitude estimation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [\[CrossRef\]](#)
2. Xu, S.; Huang, W.; Wang, H.; Zheng, W.; Wang, J.; Chai, Y.; Ma, M. A Simultaneous Diagnosis Method for Power Switch and Current Sensor Faults in Grid-Connected Three-Level NPC Inverters. *IEEE Trans. Power Electron.* **2022**, *38*, 1104–1118. [\[CrossRef\]](#)
3. Pecina Sánchez, J.A.; Campos-Delgado, D.U.; Espinoza-Trejo, D.R.; Valdez-Fernández, A.A.; De Angelo, C.H. Fault diagnosis in grid-connected PV NPC inverters by a model-based and data processing combined approach. *IET Power Electron.* **2019**, *12*, 3254–3264. [\[CrossRef\]](#)
4. Tang, Q.; Chai, Y.; Qu, J.; Ren, H. Fisher discriminative sparse representation based on DBN for fault diagnosis of complex system. *Appl. Sci.* **2018**, *8*, 795. [\[CrossRef\]](#)
5. Li, Z.; Lv, Y.; Yuan, R.; Zhang, Q. An intelligent fault diagnosis method of rolling bearings via variational mode decomposition and common spatial pattern-based feature extraction. *IEEE Sens. J.* **2022**, *22*, 15169–15177. [\[CrossRef\]](#)
6. Yang, D.; Lv, Y.; Yuan, R.; Yang, K.; Zhong, H. A novel vibro-acoustic fault diagnosis method of rolling bearings via entropy-weighted nuisance attribute projection and orthogonal locality preserving projections under various operating conditions. *Appl. Acoust.* **2022**, *196*, 108889. [\[CrossRef\]](#)
7. Balta, S.; Zavrak, S.; Eken, S. Real-Time Monitoring and Scalable Messaging of SCADA Networks Data: A Case Study on Cyber-Physical Attack Detection in Water Distribution System. In *International Congress of Electrical and Computer Engineering*; Springer: Cham, Switzerland, 2022.
8. Eken, S. An exploratory teaching program in big data analysis for undergraduate students. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 4285–4304. [\[CrossRef\]](#)
9. Chen, H.; Jiang, B.; Ding, S.X.; Huang, B. Data-driven fault diagnosis for traction systems in high-speed trains: A survey, challenges, and perspectives. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1700–1716. [\[CrossRef\]](#)
10. Chiang, L.H.; Kotanchek, M.E.; Kordon, A.K. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Comput. Chem. Eng.* **2004**, *28*, 1389–1401. [\[CrossRef\]](#)
11. Lee, J.M.; Yoo, C.; Lee, I.B. Statistical process monitoring with independent component analysis. *J. Process. Control* **2004**, *14*, 467–485. [\[CrossRef\]](#)
12. Lee, J.M.; Yoo, C.; Choi, S.W.; Vanrolleghem, P.A.; Lee, I.B. Nonlinear process monitoring using kernel principal component analysis. *Chem. Eng. Sci.* **2004**, *59*, 223–234. [\[CrossRef\]](#)
13. Baudat, G.; Anouar, F. Generalized discriminant analysis using a kernel approach. *Neural Comput.* **2000**, *12*, 2385–2404. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Zhu, J.; Ge, Z.; Song, Z.; Gao, F. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. *Annu. Rev. Control* **2018**, *46*, 107–133. [\[CrossRef\]](#)
15. Erdem, T.; Eken, S. Layer-Wise Relevance Propagation for Smart-Grid Stability Prediction. In *Mediterranean Conference on Pattern Recognition and Artificial Intelligence*; Springer: Cham, Switzerland, 2022.
16. Breviglieri, P.; Erdem, T.; Eken, S. Predicting Smart Grid Stability with Optimized Deep Models. *SN Comput. Sci.* **2021**, *2*, 1–12. [\[CrossRef\]](#)
17. Chenglin, W.E.N.; Feiya, L.Ü. Review on deep learning based fault diagnosis. *J. Electron. Inf. Technol.* **2020**, *42*, 234–248.
18. Ren, H.; Qu, J.F.; Chai, Y.; Tang, Q.; Ye, X. Deep learning for fault diagnosis: The state of the art and challenge. *Control Decis.* **2017**, *32*, 1345–1358.
19. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [\[CrossRef\]](#)
20. Chen, H.; Chen, Z.; Chai, Z.; Jiang, B.; Huang, B. A single-side neural network-aided canonical correlation analysis with applications to fault diagnosis. *IEEE Trans. Cybern.* **2021**, *52*, 9454–9466. [\[CrossRef\]](#)
21. Yang, D.; Lv, Y.; Yuan, R.; Li, H.; Zhu, W. Robust fault diagnosis of rolling bearings via entropy-weighted nuisance attribute projection and neural network under various operating conditions. *Struct. Health Monit.* **2022**, *21*, 14759217221077414. [\[CrossRef\]](#)
22. Yuan, R.; Lv, Y.; Wang, T.; Li, S.; Li, H. Looseness monitoring of multiple M1 bolt joints using multivariate intrinsic multiscale entropy analysis and Lorentz signal-enhanced piezoelectric active sensing. *Struct. Health Monit.* **2022**, *21*, 14759217221088492. [\[CrossRef\]](#)

23. Liu, B.; Chai, Y.; Liu, Y.; Huang, C.; Wang, Y.; Tang, Q. Industrial process fault detection based on deep highly-sensitive feature capture. *J. Process. Control* **2021**, *102*, 54–65. [[CrossRef](#)]
24. Wang, Y.; Yang, H.; Yuan, X.; Shardt, Y.A.; Yang, C.; Gui, W. Deep learning for fault-relevant feature extraction and fault classification with stacked supervised auto-encoder. *J. Process. Control* **2020**, *92*, 79–89. [[CrossRef](#)]
25. Ma, Y.; Shi, H.; Tan, S.; Tao, Y.; Song, B. Consistency regularization auto-encoder network for semi-supervised process fault diagnosis. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3184346. [[CrossRef](#)]
26. Huang, C.; Chai, Y.; Zhu, Z.; Liu, B.; Tang, Q. A Novel Distributed Fault Detection Approach Based on the Variational Autoencoder Model. *ACS Omega* **2022**, *7*, 2996–3006. [[CrossRef](#)]
27. Jiang, Q.; Yan, S.; Yan, X.; Yi, H.; Gao, F. Data-driven two-dimensional deep correlated representation learning for nonlinear batch process monitoring. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2839–2848. [[CrossRef](#)]
28. Naftali, T.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Jerusalem, Israel, 26 April–1 May 2015.
29. McAvoy, T.J.; Ye, N. Base control for the Tennessee Eastman problem. *Comput. Chem. Eng.* **1994**, *18*, 383–413. [[CrossRef](#)]
30. Yu, J. Hidden Markov models combining local and global information for nonlinear and multimodal process monitoring. *J. Process. Control* **2010**, *20*, 344–359. [[CrossRef](#)]
31. Lee, J.M.; Qin, S.J.; Lee, I.B. Fault detection of non-linear processes using kernel independent component analysis. *Can. J. Chem. Eng.* **2007**, *85*, 526–536. [[CrossRef](#)]
32. Huang, C.; Chai, Y.; Liu, B.; Tang, Q.; Qi, F. Industrial process fault detection based on KGLPP model with Cam weighted distance. *J. Process. Control* **2021**, *106*, 110–121. [[CrossRef](#)]
33. Bounoua, W.; Benkara, A.B.; Kouadri, A.; Bakdi, A. Online monitoring scheme using principal component analysis through Kullback-Leibler divergence analysis technique for fault detection. *Trans. Inst. Meas. Control* **2020**, *42*, 1225–1238. [[CrossRef](#)]
34. Liu, B.; Chai, Y.; Huang, C.; Fang, X.; Tang, Q.; Wang, Y. Industrial process monitoring based on optimal active relative entropy components. *Measurement* **2022**, *197*, 111160. [[CrossRef](#)]
35. Lau, C.K.; Ghosh, K.; Hussain, M.A.; Hassan, C.C. Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS. *Chemom. Intell. Lab. Syst.* **2013**, *120*, 1–14. [[CrossRef](#)]