

Article

Nonlinear Dynamic Process Monitoring Based on Discriminative Denoising Autoencoder and Canonical Variate Analysis

Jun Liang, Daoguang Liu, Yinxiao Zhan  and Jiayu Fan *

College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China; jliang@zju.edu.cn (J.L.); lmeiiu@163.com (D.L.); zhanyinxiao@zju.edu.cn (Y.Z.)

* Correspondence: jiayufan_hsz@163.com; Tel.: +86-176-2595-1153

Abstract: Modern industrial processes are characterized by increasing complexity, often exhibiting pronounced dynamic behaviors and significant nonlinearity. Addressing these dynamic and nonlinear characteristics is essential for effective process monitoring. However, many existing methods for process monitoring and fault diagnosis are insufficient in handling these challenges. In this article, we present a novel process monitoring approach, CVA-DisDAE, which integrates an improved Denoising Autoencoder (DAE) with Canonical Variate Analysis (CVA) to address the challenges posed by dynamic behaviors and nonlinear relationships in industrial processes. First, CVA is employed to reduce data dimensionality and minimize information redundancy by maximizing correlations between past and future observations, thereby effectively capturing process dynamics. Following this, we introduce a discriminative DAE model (DisDAE) designed to serve as a semi-supervised denoising autoencoder for precise feature extraction. This is achieved by incorporating both between-class separability and within-class variability into the traditional DAE framework. The key distinction between the proposed DisDAE and the conventional DAE lies in the integration of a linear discriminant analysis (LDA) penalty into the DAE's loss function, resulting in extracted features that are more conducive to fault classification. Finally, we validate the effectiveness of the proposed semi-supervised dynamic process monitoring approach through its application to the Tennessee Eastman benchmark process, demonstrating its superior performance.

Keywords: process monitoring; fault diagnosis; canonical variate analysis (CVA); dynamics; denoising autoencoder (DAE); nonlinear process; linear discriminant analysis (LDA)



Citation: Liang, J.; Liu, D.; Zhan, Y.; Fan, J. Nonlinear Dynamic Process Monitoring Based on Discriminative Denoising Autoencoder and Canonical Variate Analysis. *Actuators* **2024**, *13*, 440. <https://doi.org/10.3390/act13110440>

Academic Editor: Giorgio Olmi

Received: 11 September 2024

Revised: 23 October 2024

Accepted: 31 October 2024

Published: 2 November 2024



Copyright: © 2024 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

Nonlinear dynamic process monitoring arises from the increasing complexity and nonlinearity of modern industrial systems, where traditional linear monitoring techniques fail to accurately capture dynamic variations and anomalies [1,2]. This necessitates the development of advanced methods capable of handling nonlinear dynamics, ensuring real-time detection of process deviations for enhanced operational safety and efficiency. In many industrial processes, dynamic behaviors are common, and both input and output variables often display autocorrelated data patterns. However, traditional data-driven monitoring methods frequently assume that data samples are collected independently, which can significantly degrade monitoring performance when dynamic information is not accounted for [3]. Over time, researchers have focused on three key strategies to address the challenges associated with monitoring dynamic industrial processes: dynamic multivariate statistical process monitoring (MSPM), time series analysis, and state-space models [2,4].

MSPM methods primarily utilize historical data collected during normal operations to define a baseline for normal conditions. Techniques like Principal Component Analysis (PCA), Independent Component Analysis (ICA), and other dimensionality reduction methods serve as the basis for MSPM. One of the earliest advancements in this area was

the dynamic PCA model introduced by Ku et al. [5], which improves the dataset by incorporating time-lagged samples for each variable, enabling the capture of autocorrelation patterns. Li and Chen et al. [6] introduced Principal Discriminant Component Analysis (PDCA), which is more efficient for detecting deviations between real-time monitoring data and normal operational datasets. Lou and Sheriff et al. [7,8] also achieved significant success by reducing high-dimensional process data, encompassing both input and output variables, into a smaller set of uncorrelated principal components (PCs), which are then tracked using statistical indices. However, PCA faces challenges in detecting faults related to key performance indicators (KPIs) as these may be hidden by interference from KPI-uncorrelated components.

Partial Least Squares (PLS) and Canonical Correlation Analysis (CCA) are commonly employed for monitoring KPI-related performance. In contrast to PCA, PLS and CCA focus on extracting KPI-relevant components from process data, enhancing fault detection. He et al. [9] proposed a modified PLS approach that more effectively decomposes data into KPI-related and KPI-independent components, facilitating the creation of two statistical indices for fault diagnosis. Si et al. [1] further enhanced the kernel partial least squares (KPLS) method by incorporating KPI-related information within the residual subspace, making it more applicable to KPI-focused process monitoring. By using correlation analysis for residual generation, a new data-driven monitoring method was developed that reduces communication costs and the risks of centralized monitoring without sacrificing performance [10]. PLS and CCA are particularly sensitive to anomalies in KPI-related components, and several methods have been proposed for process monitoring. Fazai and Harkat [11,12] combined PLS with generalized likelihood ratio tests to address the challenges posed by nonlinear and interval-characterized process data. Yin et al. [13] highlighted that traditional PLS approaches lose efficiency as the number of uncorrelated faults increases, prompting the development of the Modified Orthogonal Projection to Latent Structures (MOPLS), which improves the detection of quality-related faults.

Other methods, such as Canonical Variate Analysis (CVA), have also been utilized to reduce data dimensionality and redundancy. CVA is a data-driven technique that maximizes the correlation between two sets of variables [14], making it well-suited for capturing dynamic process behavior. A modified CVA based on the joint approximate diagonalization method was introduced to extract shared-mode features [15]. Shang et al. [16] proposed Recursive Ensemble Canonical Variate Analysis (RECVA), a method for detecting incipient faults in dynamic processes. To manage the nonlinear and dynamic nature of large-scale industrial data, Wang et al. [17] employed an autoregressive moving average model in conjunction with kernel PCA to monitor industrial processes characterized by complex dynamic and nonlinear properties.

Artificial Neural Networks (ANNs) have proven to be powerful tools for approximating complex nonlinear functions across various industrial applications. One notable unsupervised ANN variant, the autoencoder, autonomously learns features by minimizing reconstruction errors [18]. An autoencoder generally comprises two main parts: an encoder, which transforms input data into a latent space, and a decoder, which reconstructs the input from this latent space. In feature extraction, the autoencoder seeks to retain as much input information as possible by minimizing reconstruction error. Variations of autoencoders, such as denoising autoencoders (DAEs) [19] and stacked autoencoders (SAEs) [20], have been widely used in areas like process monitoring and fault detection. Nonetheless, the features obtained through traditional autoencoders may not always be optimal for downstream tasks [21], largely due to two key issues. The first is related to noise in the input data, which can distort the representations learned by conventional autoencoders. This is particularly problematic in industrial settings, where raw data is often noisy or contains irrelevant information. Standard autoencoders may produce representations that compromise diagnostic accuracy when using such data. To mitigate this, denoising autoencoders (DAEs) were introduced, incorporating artificial noise into the training process to enhance robustness [19,22]. The DAE architecture is similar to that of

a regular autoencoder but includes the additional step of corrupting the input data [23]. By reconstructing this corrupted input, the model learns more robust features, reducing overfitting and boosting performance [24,25]. The aim is to ensure that the reconstructed data remains faithful to the original, helping the model to derive clean representations from noisy inputs, thereby improving fault detection accuracy [26]. A second issue is the inability of unsupervised autoencoders to utilize label information, which can limit their capacity to extract meaningful features for diagnostic tasks. Newer approaches, such as supervised and semi-supervised autoencoders [18,27–29], have been proposed to overcome this limitation. One effective technique involves linear discriminant analysis (LDA) [30], which introduces a prior that encourages similar latent representations for data from the same class while separating representations from different classes, thus improving the model's ability to distinguish between them.

As a linear supervised classification method, LDA aims to find a set of characteristic directions that optimize class separability by maximizing between-class differences and minimizing within-class variability. Projecting data samples onto the feature subspace formed by the leading discriminant directions enables effective class separation while achieving significant dimensionality reduction [31]. The literature has shown that the monitoring and diagnostic performance of LDA surpasses that of PCA and discriminant PLS approaches [32,33]. However, despite its success in process monitoring, LDA has limitations when handling nonlinear data or data contaminated by noise. To address this, the paper proposes an enhanced strategy, integrating a DAE for feature extraction with LDA for supervised classification.

In this article, a novel process monitoring approach CVA-DisDAE is proposed to address problems of dynamic behaviors and nonlinear relations in industrial processes by combining improved DAE and CVA algorithms. The approach begins with the application of CVA to reduce data dimensionality and eliminate redundant information by maximizing the correlation between past and future observations. This allows the dynamic characteristics of the process data to be captured through a low-dimensional augmented data matrix containing several time-lagged samples. Next, a discriminative DAE model, named *DisDAE*, is introduced to achieve precise feature extraction in a semi-supervised denoising autoencoder framework. This is accomplished by incorporating class separability and within-class variability into the traditional DAE structure. The proposed *DisDAE* differs from conventional DAE in two key ways: first, it utilizes label information by embedding an LDA classification distance penalty in the DAE's loss function, with a weighting parameter allowing for an adjustable balance between DAE and LDA. Second, *DisDAE* undergoes both an unsupervised pre-training phase, similar to a traditional autoencoder, and a supervised parameter training phase. In the supervised phase, the LDA distance penalty optimizes the separation between class centers, allowing the autoencoder to extract more discriminative features for diagnosis. Finally, a Softmax classifier is employed in *DisDAE* for fault classification, with a weight decay term included to prevent overfitting during the training of the Softmax network [18]. To validate the effectiveness of the proposed *DisDAE*, comparative experiments were conducted using various autoencoders for process monitoring, including tests on the Tennessee Eastman dataset, to evaluate the monitoring performance of *DisDAE* against other commonly used autoencoder models. In summary, the contributions of this research can be summarized as follows:

- (1) We propose a novel method, CVA-DisDAE, which is specifically designed to extract nonlinear and dynamic features for fault diagnosis in industrial processes.
- (2) We introduce a new loss term that can maximize between-class separability while minimizing within-class variability, enhancing the traditional DAE's loss function.
- (3) A CVA-based algorithm is developed to capture process dynamics by leveraging past and future data, effectively reducing dimensionality and redundancy while enhancing the extraction of meaningful dynamics.

The structure of the article is as follows: Section 2 firstly provides a brief overview of CVA and LDA, and then the proposed CVA-DisDAE method for nonlinear dynamic process

monitoring is described in detail, covering the CVA-DisDAE framework, the development of the CVA-based algorithm for capturing process dynamics, the design of the DisDAE model for feature extraction and fault classification, as well as a flowchart outlining the off-line and on-line process monitoring procedures. Section 3 presents the Tennessee Eastman benchmark process as a case study to validate the effectiveness of the proposed approach. Finally, Section 4 offers the concluding remarks of the paper.

2. Research on the Process Monitoring Approach

In this section, the relevant methods used for process monitoring and fault diagnosis are first reviewed, and then the proposed method for industrial process monitoring in this paper is introduced. Specifically, a framework of the proposed method is first presented. Secondly, the CVA-based algorithm for process dynamic capturing is reported. Moreover, the DisDAE model for feature extraction and fault classification is developed in detail. Finally, the implementing flowchart of CVA-DisDAE is given.

2.1. Preliminaries

2.1.1. Canonical Variate Analysis (CVA)

CVA is an optimal technique for dimensionality reduction, designed to maximize the correlation statistic between two variable sets [14,34]. Let $x \in R^m$ and $y \in R^n$. Assume covariance matrices Σ_{xx} and Σ_{yy} , and cross-covariance matrix Σ_{xy} , there exist matrices $J \in R^{m \times m}$ and $L \in R^{n \times n}$ such that

$$J\Sigma_{xx}J^T = I_{\bar{m}} \quad (1)$$

$$L\Sigma_{yy}L^T = I_{\bar{n}} \quad (2)$$

and

$$J\Sigma_{xy}L^T = D = \text{diag}(\gamma_1, \dots, \gamma_r, 0, \dots, 0) \quad (3)$$

where $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_r$, $\bar{m} = \text{rank}(\Sigma_{xx})$, $\bar{n} = \text{rank}(\Sigma_{yy})$, the matrix D holds the canonical correlations γ_i , and the notation I_k represents a block diagonal matrix, with a $k \times k$ identity matrix in the first block and a zero matrix in the second. The canonical variable vector $c = Jx$ comprises a set of uncorrelated variables and has a covariance matrix [34]:

$$\Sigma_{cc} = J\Sigma_{xx}J^T = I_{\bar{m}} \quad (4)$$

Similarly, the canonical variable vector $d = Ly$ consists of uncorrelated variables, with a covariance matrix given by:

$$\Sigma_{dd} = L\Sigma_{yy}L^T = I_{\bar{n}} \quad (5)$$

The cross-covariance matrix between c and d is diagonal, meaning that the two vectors are correlated only in pairs, with the strength of the pairwise correlations defined by the canonical correlations γ_i , which can be ranked accordingly. In the common scenario where the covariance matrices are invertible, the projection matrices J and L , along with the canonical correlation matrix D , can be derived by performing the SVD:

$$\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2} = U\Sigma V^T \quad (6)$$

Here, $J = U^T\Sigma_{xx}^{-1/2}$, $L = V^T\Sigma_{yy}^{-1/2}$, and $D = \Sigma$. The scaling factors $\Sigma_{xx}^{-1/2}$ and $\Sigma_{yy}^{-1/2}$ ensure that the canonical variables are uncorrelated with unit variance, while the matrices U^T and V^T rotate them to ensure c and d are only correlated pairwise. The singular values indicate the degree of these pairwise correlations. The CVA method, being a data-driven approach, is well-suited for dynamic process monitoring by enhancing the correlations between past and future observations [34,35]. This is achieved by performing SVD on the covariance matrices that capture historical and future data, allowing the extraction of the appropriate projection and singular value matrices.

2.1.2. Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a discriminative technique that identifies a linear transformation to enhance the separation between different classes while reducing the variability within each class [30,36]. This transformation projects the data into a subspace with a maximum dimensionality of $L-1$, where L denotes the number of classes. Notably, LDA is distribution-free, meaning it does not assume any specific data distribution. Typically, LDA is integrated with a classifier operating in the transformed, lower-dimensional space [37].

Consider $X = [x(1), x(2), \dots, x(n)]^T \in R^{n \times d}$ as the desired low-dimensional representation of the input data $Y = [y(1), y(2), \dots, y(n)]^T \in R^{n \times m}$, where $d < L-1$ specifies the dimensionality of the reduced space. LDA strives to maximize the separability between classes while minimizing the intra-class scatter by optimizing

$$J(X) = \text{tr}(S_w^{-1} S_b) \quad (7)$$

or by minimizing

$$J(X) = \text{tr}(S_b^{-1} S_w) \quad (8)$$

where S_b and S_w are the between- and within-class scatter matrices:

$$S_b = \frac{1}{n} \sum_{i=1}^L n_i (m_i - m_0)(m_i - m_0)^T \quad (9)$$

$$S_w = \frac{1}{n} \sum_{i=1}^L \sum_{j=1}^{n_i} (x^{(i)}(j) - m_i)(x^{(i)}(j) - m_i)^T \quad (10)$$

where $X^{(i)} = [x^{(i)}(1), x^{(i)}(2), \dots, x^{(i)}(n_i)]^T \in R^{n_i \times d}$ are the n_i point in class i , m_i is the mean of the elements termed i , and m_0 is the mean of the data in all classes.

It is not trivial to ensure that $\text{tr}(S_w)$ measures the closeness of the vectors within the classes, and $\text{tr}(S_b)$ measures the separation between classes; that is, an optimal transformation would maximize $\text{tr}(S_b)$ and minimize $\text{tr}(S_w)$ simultaneously. In addition, the optimization problem in Equations (7) or (8) is equivalent to the following generalized eigenvalue problem: $S_b x = \lambda S_w x$, for $\lambda \neq 0$. The solution is obtained by using an SVD on the scatter matrices.

2.2. Framework of the Proposed Method

Data-driven process monitoring and fault diagnosis generally involve three steps: data processing, feature extraction, and classification. Data processing prepares the sampled data for dynamic monitoring, often requiring dimensionality reduction to manage computational load, especially when using techniques like DPCA [1]. Feature extraction identifies key attributes from large datasets to characterize faults, while classification determines the fault type and whether the system is operating correctly.

This paper presents a novel approach, CVA-DisDAE, which combines DAE for nonlinear data processing and CVA for capturing dynamic variations, as shown in Figure 1. The framework consists of two modules. First, CVA extracts latent dynamic information and reduces dimensionality. Then, this augmented data is processed by DisDAE for feature extraction and classification. By integrating an LDA penalty into DAE's loss function, DisDAE enhances class separation and minimizes intra-class variance, balancing unsupervised and supervised learning through a weighting parameter. A Softmax classifier further refines fault classification, with weight decay preventing overfitting during training. Detailed descriptions of the two modules are provided in the following subsections.

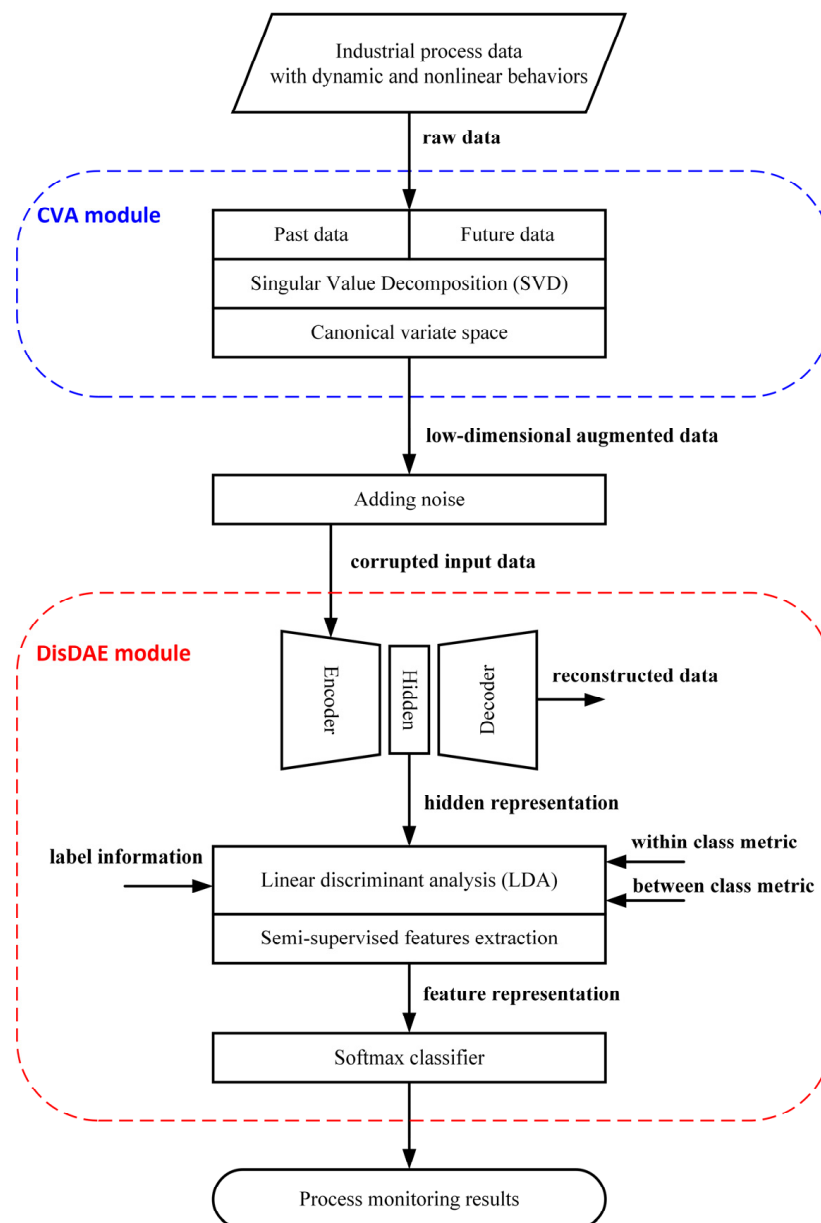


Figure 1. The framework of the proposed CVA-DisDAE. It includes two modules. In the first module, process raw data is analyzed by CVA to obtain the latent augmented data, while in the second module, the corrupted low-dimensional augmented data is fed into a DisDAE network for feature extraction and fault classification.

2.3. CVA-Based Algorithm for Capturing Process Dynamics

Most process monitoring or fault diagnosis methods can only handle datasets sampled under stable conditions. However, when the correlation between variables is strong or the sampling interval is small, the sampled data necessarily contain a certain degree of dynamics. Therefore, both process monitoring and fault classification should take full account of the dynamics in the samples. As for data-driven methods, a common way to extract dynamics from data is to dynamically augment the sampled data to obtain a dynamic augmented matrix, which is then used as input to a process monitoring or fault diagnosis model. For example, let the original dataset sampled for the process

be $X = [x_1^T, x_2^T \cdots x_n^T]^T$, the corresponding dynamic augmented matrix for X can be defined as

$$X_D = \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-d+1} \\ x_2 & x_3 & \cdots & x_{n-d+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_d & x_{d+1} & \cdots & x_n \end{bmatrix} \quad (11)$$

where each column in X_D is augmented by containing d consecutive samples of data, and d is the width of the dynamic augmenting window. In this way, process dynamic information can be greatly preserved because d consecutive samples of data are combined and treated as a whole. The window width d is taken to mean that when the interval between some two samples exceeds d , the dynamic correlation between the two samples is weak enough not to be considered.

Nonetheless, this dynamic augmenting approach has some drawbacks. First, there are several samples in each sample combination that are identical to other combinations, resulting in information redundancy and overfitting of the model; second, X_D makes the input dimension of the model grow exponentially compared to X ; especially when the window width d is large, the computational effort of the model will be very large. To this end, CVA is used to reduce the dimensionality of X_D in this paper.

Let the process sample dataset X_0 be normalized to zero mean and unit standard deviation, $X_0 = [x_0(1)^T, x_0(2)^T, \cdots x_0(n)^T]^T \in R^{m_0 \times n}$, $x_0(i) \in R^{m_0}$, $i = 1, 2, \cdots n$, its past data matrix P and future data matrix F are constructed as

$$P = \begin{bmatrix} x_0(l_p) & x_0(l_p + 1) & \cdots & x_0(l_p + N - 1) \\ x_0(l_p - 1) & x_0(l_p) & \cdots & x_0(l_p + N - 2) \\ \vdots & \vdots & \ddots & \vdots \\ x_0(1) & x_0(2) & \cdots & x_0(N) \end{bmatrix} \in R^{m_0 l_p \times N} \quad (12)$$

$$F = \begin{bmatrix} x_0(l_p + 1) & x_0(l_p + 2) & \cdots & x_0(l_p + N) \\ x_0(l_p + 2) & x_0(l_p + 3) & \cdots & x_0(l_p + N + 1) \\ \vdots & \vdots & \ddots & \vdots \\ x_0(l_p + l_f) & x_0(l_p + l_f + 1) & \cdots & x_0(n) \end{bmatrix} \in R^{m_0 l_f \times N} \quad (13)$$

where the l_p and l_f are time lags, and $N = n - l_p - l_f + 1$ is the dimension of P and F . The covariance Σ_{PP} , Σ_{FF} and cross-covariance Σ_{FP} , Σ_{PF} can be estimated as

$$\begin{bmatrix} \Sigma_{PP} & \Sigma_{PF} \\ \Sigma_{FP} & \Sigma_{FF} \end{bmatrix} = \frac{1}{N-1} \begin{bmatrix} PP^T & PF^T \\ FP^T & FF^T \end{bmatrix} \quad (14)$$

In order to obtain the optimal linear combination of past and future observations, an SVD is performed on the following expression [38]:

$$H = \Sigma_{FF}^{-1/2} \Sigma_{FP} \Sigma_{PP}^{-1/2} = U \Sigma V^T \quad (15)$$

In this equation, U and V represent the left and right singular matrices of H , respectively. The diagonal matrix $\Sigma = \text{diag}(\gamma_1, \cdots \gamma_r, 0, \cdots 0)$ contains all singular values, where r is the rank of H . By arranging the singular values in decreasing order and reorganizing the corresponding singular vectors, the top m columns of V can be selected for their strong pairwise correlation with the corresponding columns of U [39]. Consequently, for the purpose of dimensionality reduction, a new matrix V_m with a reduced dimension (where $m < m_0 l_p$) has been derived. The transformation matrix J_p and J_R , used to convert the

$m_0 l_p$ -dimensional past matrices to the m -dimensional process variables and the residuals, respectively, are computed as follows:

$$J_P = V_m^T \Sigma_{PP}^{-1/2} \in R^{m \times m_0 l_p} \quad (16)$$

$$J_R = (I - V_m V_m^T) \Sigma_{PP}^{-1/2} \in R^{m_0 l_p \times m_0 l_p} \quad (17)$$

Finally, the process variables and the residuals are derived from the past vector matrix, respectively [38].

$$X = J_P P \in R^{m \times N} \quad (18)$$

$$E = J_R P \in R^{m_0 l_p \times N} \quad (19)$$

Hence, the canonical variables space can be partitioned into two orthogonal subspaces. The first m canonical variables (CVs) span the process subspace X , and the remaining CVs span the residual subspace E .

It is essential to select appropriate parameters, e.g., time lags and reduced dimensions, for process monitoring using this present approach. The time lags, l_p and l_f , play a key role in defining the canonical variable space. While increasing these lags can enhance the model's accuracy, it may also introduce overfitting and reduce the robustness of the monitoring technique. Furthermore, higher lags demand larger datasets, thus raising computational costs. On the other hand, if the time lags are too small, they fail to capture the correlation within the time series [40]. One common method for determining appropriate lags is the AIC criterion, which is used in model selection. However, variance and covariance only reflect the statistical properties of stochastic processes at individual time points and do not capture the inherent correlations within the process. Therefore, the autocorrelation function, a specific form of cross-correlation, is applied to determine l_p and l_f .

It is a crucial aspect of CVA-based process monitoring to select the reduced dimension, which determines the state order of the canonical space [40]. There are two common methods used to accomplish this. The first method is based on the singular values' magnitude, where the top m dominant canonical variables are selected as process variables based on a specific threshold. When the noise level is low compared to the monitoring variables, the first m singular values stand out significantly from the others. The second method involves minimizing the AIC to choose the state order. In scenarios where noise is significant and the singular values decay gradually, the AIC criterion is frequently applied in the literature to define the canonical space order.

2.4. Discriminative DAE Model for Feature Extraction and Fault Classification

The structure of the discriminative denoising autoencoder (*DisDAE*) is similar to that of a typical *DAE*, but it differs in its training methodology and loss function. As a semi-supervised model, *DisDAE* undergoes a two-stage training process, involving both unsupervised and supervised phases. By incorporating supervised learning, *DisDAE* is able to utilize label information, enabling it to produce a more meaningful representation than a conventional *DAE*.

2.4.1. Unsupervised Training Methods

A typical *DAE* comprises an encoder and a decoder [18,41], where the former maps the vector $x \in R^m$ to a representation $s \in R^l$, while the decoder attempts to rebuild the previous input vector by minimizing the difference between the vector x and the reconstructed $\hat{x} \in R^m$ [18]. When the reconstruction error is minimized sufficiently, the representation s retains the essential information about the input vector. To enhance the robustness of this intermediate representation against feature noise, the input redundancy is increased either by injecting Gaussian white noise into the data [41] or by employing a stochastic technique known as dropout [42]. As a result, the input is transformed into $\tilde{x} \in R^m$.

Encoder: The encoder is defined as:

$$s = f_{\text{encoder}}(W_e \tilde{x} + b_e) \quad (20)$$

where f_{encoder} is an activation function, such as sigmoid, tanh, or identity; W_e is a weight matrix of size $l \times m$; and b_e is a bias vector with dimension l [43].

Decoder: The decoder is given as:

$$\hat{x} = f_{\text{decoder}}(W_d s + b_d) \quad (21)$$

where f_{decoder} represents an activation function; W_d is a weight matrix of size $m \times l$; and b_d is a bias vector with dimension m .

The initial stage of the training process is the unsupervised phase. During this phase, the architecture and loss function of *DisDAE* are similar to those of a conventional *DAE*, with the model being trained using unlabeled data [18]. The objective is to optimize the parameters $\theta = (W_e, W_d, b_e, b_d)$ by minimizing the following loss function:

$$J(\theta) = L(x, \hat{x}) + \lambda \|W\|^2 = \sum_{i=1}^n \|x_i - \hat{x}_i\|^2 + \lambda (\|W_e\|^2 + \|W_d\|^2) \quad (22)$$

2.4.2. Supervised Training Methods

For the supervised training part, the parameters obtained from the unsupervised training part serve as initial values for further refinement. The loss function here differs from that of unsupervised training and is expressed as follows:

$$J_{\text{DisDAE}}(\theta) = L(x, \hat{x}) + \lambda \|W\|^2 + \text{penalty} \quad (23)$$

The “*penalty*” term, related to LDA, helps improve class separability by minimizing within-class variability and maximizing between-class distance. Based on Equation (8), the penalty term can be represented as follows:

$$\text{penalty} = \frac{1}{\sigma_d^2} \text{tr}(S_b^{(-1)} S_w) \quad (24)$$

where σ_d^2 is a scale parameter that influences the penalty’s weight. A large σ_d^2 retains more information from the data, while a small σ_d^2 enhances inter-class separability. Consequently, the supervised training loss function becomes [18]:

$$J_{\text{DisDAE}}(\theta) = L(x, \hat{x}) + \lambda \|W\|^2 + \frac{1}{\sigma_d^2} \text{tr}(S_b^{(-1)} S_w) \quad (25)$$

In this paper, the value of σ_d^2 strikes a balance between the discriminative power of *DisDAE* (when σ_d^2 is small) and its generalization capability (when σ_d^2 is large). In extreme cases ($\sigma_d^2 \rightarrow 0$ or $\sigma_d^2 \rightarrow \infty$), Equation (25) converges to LDA or *DAE*, respectively. Additionally, the distance metric in S_b, S_w is based on the Euclidean distance, but other metrics, such as Manhattan or Chebyshev distances, can also be employed [18].

Let $x(k) \in R^m$, $k = 1, 2, \dots, N$ be the sample of canonical variables space obtained from Equation (18) by CVA, $\tilde{x}^{(i)}(k) \in R^m$ be the k th sample vector with noise “corrupted” in the i -th class of $x(k)$, which is also the input vector of the *DAE* encoder, $i = 1, 2, \dots, L$, $k = 1, 2, \dots, L_i$, $\sum_{i=1}^L L_i = N$, and N is the number of all samples. $\hat{x}^{(i)}(k) \in R^m$ is the output vector of the *DAE* decoder corresponding to $\tilde{x}^{(i)}(k)$. From Equations (20) and (21), we have

$$s^{(i)}(k) = f_{\text{encoder}}(W_e \tilde{x}^{(i)}(k) + b_e) \quad (26)$$

$$\hat{x}^{(i)}(k) = f_{\text{decoder}}(W_d s^{(i)}(k) + b_d) \quad (27)$$

where $s^{(i)}(k) \in R^{m_s}$ is the hidden layer vector corresponding to the k th input sample vector in the i -th class of samples, both activation function $f_{encoder}(\cdot)$ and $f_{decoder}(\cdot)$ are $\tanh(\cdot)$. From the between- and within-class scatter matrix defined in Equations (9) and (10), we have

$$S_b = \frac{1}{N} \sum_{i=1}^L L_i (\bar{s}^{(i)} - \bar{s})(\bar{s}^{(i)} - \bar{s})^T \quad (28)$$

$$S_w = \frac{1}{N} \sum_{i=1}^L \sum_{k=1}^{L_i} (s^{(i)}(k) - \bar{s}^{(i)})(s^{(i)}(k) - \bar{s}^{(i)})^T \quad (29)$$

where $\bar{s}^{(i)}$ and $\bar{s}$ are the within-class mean of the hidden vector corresponding to the sample of i th class and the mean of the hidden vector corresponding to all samples, respectively.

$$\bar{s}^{(i)} = \frac{1}{L_i} \sum_{k=1}^{L_i} s^{(i)}(k), i = 1, 2, \dots, L \quad (30)$$

$$\bar{s} = \frac{1}{N} \sum_{i=1}^L L_i \bar{s}^{(i)} \quad (31)$$

Therefore, according to Equation (25), the loss function of *DisDAE* is

$$J_{DisDAE}(\theta) = J_{DAE}(\theta) + J_R(\theta) + J_{Dis}(\theta) \quad (32)$$

where $\theta = (W_e, W_d, b_e, b_d)$ is the model parameter of *DAE*, $J_{DAE}(\theta)$ is the reconstruction error, $J_R(\theta)$ is the regularization term to prevent overfitting and $J_{Dis}(\theta)$ is the LDA penalty term,

$$J_{DAE}(\theta) = L(x, \hat{x}) = \sum_{i=1}^L \sum_{k=1}^{L_i} (x^{(i)}(k) - \hat{x}^{(i)}(k))(x^{(i)}(k) - \hat{x}^{(i)}(k))^T \quad (33)$$

$$J_R(\theta) = \lambda (\|W_e\|^2 + \|W_d\|^2) \quad (34)$$

$$J_{Dis}(\theta) = \frac{1}{\sigma_d^2} \text{tr}(S_b^{-1} S_w) = \frac{1}{\sigma_d^2} \text{tr} \left\{ \left[\sum_{i=1}^L L_i (\bar{s}^{(i)} - \bar{s})(\bar{s}^{(i)} - \bar{s})^T \right]^{-1} \left[\sum_{i=1}^L \sum_{k=1}^{L_i} (s^{(i)}(k) - \bar{s}^{(i)})(s^{(i)}(k) - \bar{s}^{(i)})^T \right] \right\} \quad (35)$$

By minimizing the errorless function, the network parameters $\theta = (W_e, W_d, b_e, b_d)$ of *DisDAE* are optimized as follows:

$$\theta^* = \arg \min_{\theta} J_{DisDAE}(\theta) = \arg \min_{\theta} \{J_{DAE}(\theta) + J_R(\theta) + J_{Dis}(\theta)\} = \arg$$

$$\min_{\theta} \left\{ \sum_{i=1}^L \sum_{k=1}^{L_i} (x^{(i)}(k) - \hat{x}^{(i)}(k))(x^{(i)}(k) - \hat{x}^{(i)}(k))^T + \lambda (\|W_e\|^2 + \|W_d\|^2) + \frac{1}{\sigma_d^2} \text{tr} \left\{ \left[\sum_{i=1}^L L_i (\bar{s}^{(i)} - \bar{s})(\bar{s}^{(i)} - \bar{s})^T \right]^{-1} \left[\sum_{i=1}^L \sum_{k=1}^{L_i} (s^{(i)}(k) - \bar{s}^{(i)})(s^{(i)}(k) - \bar{s}^{(i)})^T \right] \right\} \right\} \quad (36)$$

where σ_d^2 is the trade-off coefficient. Equation (36) typically represents a non-convex optimization problem. However, due to the smooth and continuously differentiable nature of the activation function commonly used in *DAE* networks, it can be efficiently solved using gradient descent methods. The necessary gradients can be readily computed through the backpropagation algorithm (BP algorithm). To enable fault classification, a Softmax classifier is incorporated into the *DisDAE* model. The hidden feature representation learned by *DisDAE* is fed into the Softmax classifier, which then produces the fault class predictions.

As a generalization of the Logistic regression model to the multi-classification problem, the Softmax classifier is a supervised learning algorithm. Now, consider the Softmax classification problem of the *DisDAE* hidden layer feature output $s(i) \in R^{m_s}$ of Equation (26). Given a labeled training dataset $\{(s(1), y(1)), (s(2), y(2)), \dots, (s(N), y(N))\}$ with N samples, where $s(i)$ is the input feature vector of Softmax, $y(i) \in \{1, 2, \dots, L\}$ is the label corresponding to $s(i)$, and L is the number of classes. For any input feature $s(i)$, a hypothe-

sis function is needed to estimate the conditional probability that $s(i)$ belongs to each class. Hence, the hypothesis function of Softmax is defined as

$$h_{\phi}(s(i)) = \begin{bmatrix} p(y(i) = 1|s(i); \phi) \\ p(y(i) = 2|s(i); \phi) \\ \vdots \\ p(y(i) = L|s(i); \phi) \end{bmatrix} = \frac{1}{\sum_{j=1}^L \exp(\phi_j^T s(i))} \begin{bmatrix} \exp(\phi_1^T s(i)) \\ \exp(\phi_2^T s(i)) \\ \vdots \\ \exp(\phi_L^T s(i)) \end{bmatrix} \quad (37)$$

where $h_{\phi}(s(i))$ is the vector of hypothesis, in which any element $p(y(i) = j|s(i); \phi)$ is the probability that the input $s(i)$ belongs to class j , and due to the normalization factor $1/\sum_{j=1}^L \exp(\phi_j^T s(i))$, the sum of elements in the vector is equal to 1. In addition, $\phi = [\phi_1 \ \phi_2 \ \cdots \ \phi_L]^T$ is the overall model parameter of Softmax and $\phi_1, \phi_2, \cdots, \phi_L \in R^{m_s}$ are the classifier parameters corresponding to each class.

The goal of Softmax is to enable the maximum likelihood classification of the input sample $s(i)$. To this end, the model parameter ϕ is solved by the maximum likelihood method with the minimization loss function of the maximum likelihood estimation. Further, in order to prevent the overfitting problem that tends to occur during the training process, a regularization term of the parameter vector is often added to the above loss function,

$$J(\phi) = -\frac{1}{N} \left\{ \sum_{i=1}^N \sum_{j=1}^L \mu\{y(i) = j\} \ln \left[\frac{\exp(\phi_j^T s(i))}{\sum_{q=1}^L \exp(\phi_q^T s(i))} \right] \right\} + \frac{\beta}{2} \sum_{j=1}^L \|\phi_j\|^2 \quad (38)$$

where $\mu\{\}$ is an indicator function that satisfies $\mu\{true\} = 1, \mu\{false\} = 0$. To address the strong nonlinearity, noise interference, and redundancy in industrial processes to achieve robust feature representation, the DisDAE model is established by integrating DAE and LDA together. The schematic of DisDAE is shown in Figure 2. Therefore, the DisDAE can both retain the robustness of DAE-based feature representation by adding noise or carrying dropout into input data, and improve classification accuracy by maximizing between-class separability and minimizing within-class variability with an LDA-based “penalty” term.

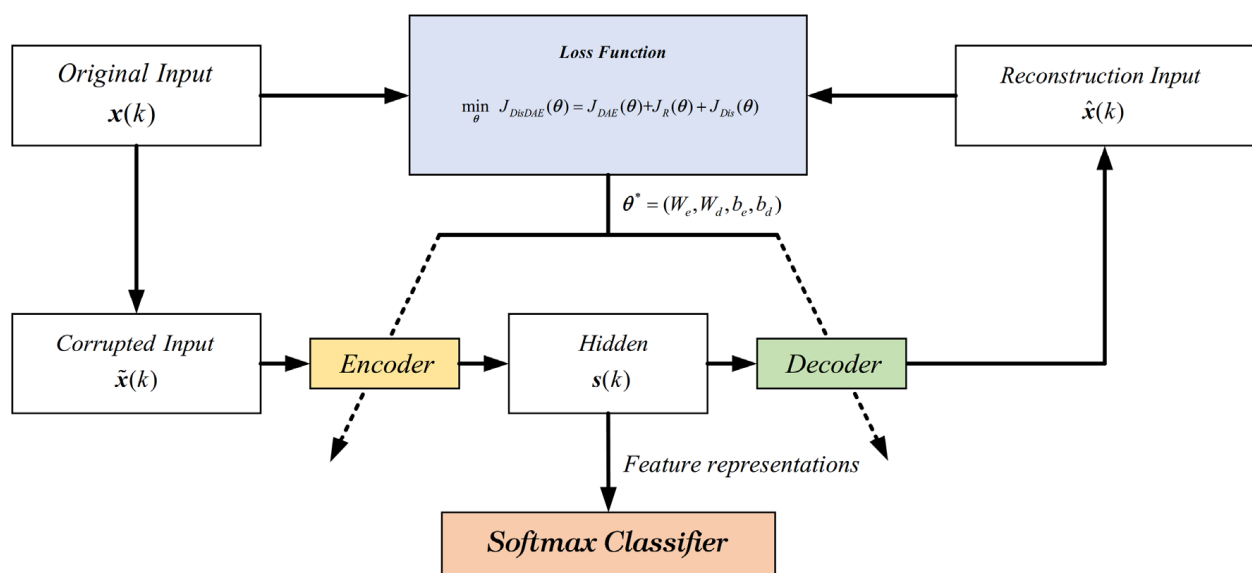


Figure 2. The schematic of DisDAE. The structure DisDAE shares a similar structure with a traditional DAE, but it differs in both its training process and loss function. DisDAE incorporates an LDA-based penalty to enhance between-class separability while reducing within-class variability. Additionally, a Softmax classifier is used to carry out fault classification.

2.5. The CVA-DisDAE Method

The flowchart of the proposed CVA-DisDAE for industrial process monitoring is shown in Figure 3. There are two phases in the practical application of the CVA-DisDAE method: offline modeling (training) and online monitoring (e.g., testing), described as follows.

The offline modeling phase of CVA-DisDAE includes the following steps:

1. Collect sample data $\{x_t\}_{t=1}^n$, under both normal and fault conditions, along with the corresponding label variables $\{y_t\}_{t=1}^n$ as the training dataset. Here, $x_t \in R^{m_0}$ should be sampled at fixed intervals and must include critical measured and manipulated variables that reflect the characteristics of the monitored process. The label variables $y_t \in \{1, 2, \dots, L\}$ should encompass normal operating conditions as well as one or more fault conditions;
2. Apply normalization to the sample data $\{(x_t - \bar{x})/\sigma\}_{t=1}^n$ in the training datasets, where $\bar{x}$ and σ are the mean and standard deviation of all samples in the training dataset;
3. Construct an augmented data matrix $P = [p_{l_p+1}, p_{l_p+2}, \dots, p_{l_p+N}] \in R^{m_0 l_p \times N}$, $p_t = [x_{t-1}^T, x_{t-2}^T, \dots, x_{t-l_p}^T]^T$ containing past training samples and an augmented data matrix $F = [f_{l_p+1}, f_{l_p+2}, \dots, f_{l_p+N}] \in R^{m_0 l_f \times N}$, $f_t = x_t$ containing only the current sample. Here, the past time lag is l_p , the future time lag is $l_f = 1$, and $N = n - l_p - l_f + 1$;
4. Calculate the canonical variables of P and F according to the CVA algorithm and take the first m dimensions as the vectors of canonical variables $C = J_P P = [c_{l_p+1}, c_{l_p+2}, \dots, c_{l_p+N}] \in R^{m \times N}$, $J_P = V_m^T \Sigma_P^{-1/2} \in R^{m \times m_0 l_p}$;
5. Add the noise term to the vectors of canonical variables $\{c_t\}_{t=l_p+1}^{l_p+N}$ at each moment as the input $\{\tilde{c}_t\}_{t=l_p+1}^{l_p+N}$ to DisDAE and train the DisDAE model using the BP algorithm, in which the loss function $J_{DisDAE}(\theta) = J_{DAE}(\theta) + J_R(\theta) + J_{Dis}(\theta)$ includes the reconstruction error term, regularization term, and LDA penalty term;
6. Encode labels $\{y_t\}_{t=1}^n$ using a one-hot scheme and solve the hypothesis function $\phi = [\phi_1 \ \phi_2 \ \dots \ \phi_L]^T$ of the Softmax classifier by the gradient descent method.

The online monitoring phase of CVA-DisDAE includes the following steps:

1. Collect test sample x'_t in real-time, ensuring that the sampling period and variable composition remain consistent with those used in the offline phase. Additionally, cache historical data to facilitate the construction of augmented vectors;
2. Apply normalization to sample $(x'_t - \bar{x})/\sigma$, where $\bar{x}$ and σ are the same as in the offline phase;
3. Construct augmented vectors $p'_t = [x'_{t-1}^T, x'_{t-2}^T, \dots, x'_{t-l_p}^T]^T$ and $f'_t = x'_t$ of test samples, when $t \geq l_p + 1$;
4. Calculate the canonical variables of p'_t and f'_t according to the CVA algorithm and take the first m dimensions as the vector of canonical variables c'_t ;
5. Input the vector of canonical variables c'_t of the test samples at the current moment to the DisDAE model that has finished training in the offline phase and obtain the input feature vector $s'_t \in R^{m_s}$ of the Softmax classifier, i.e., the hidden layer vector of the DisDAE model;
6. Obtain the real-time monitoring result y'_t of the process using the Softmax classifier.
7. It should be noted that if the process data is not collected online, but is collected in advance, then the future time lag l_f of CVA will be able to take a value greater than 1. In that case, $f_t = x_t$ in the above online modeling phase should be modified to $f_t = [x_t^T, x_{t+1}^T, \dots, x_{t+l_f-1}^T]^T$, and $f'_t = x'_t$ in the testing phase should be modified to $f'_t = [x'^T_t, x'_{t+1}^T, \dots, x'_{t+l_f-1}^T]^T$.

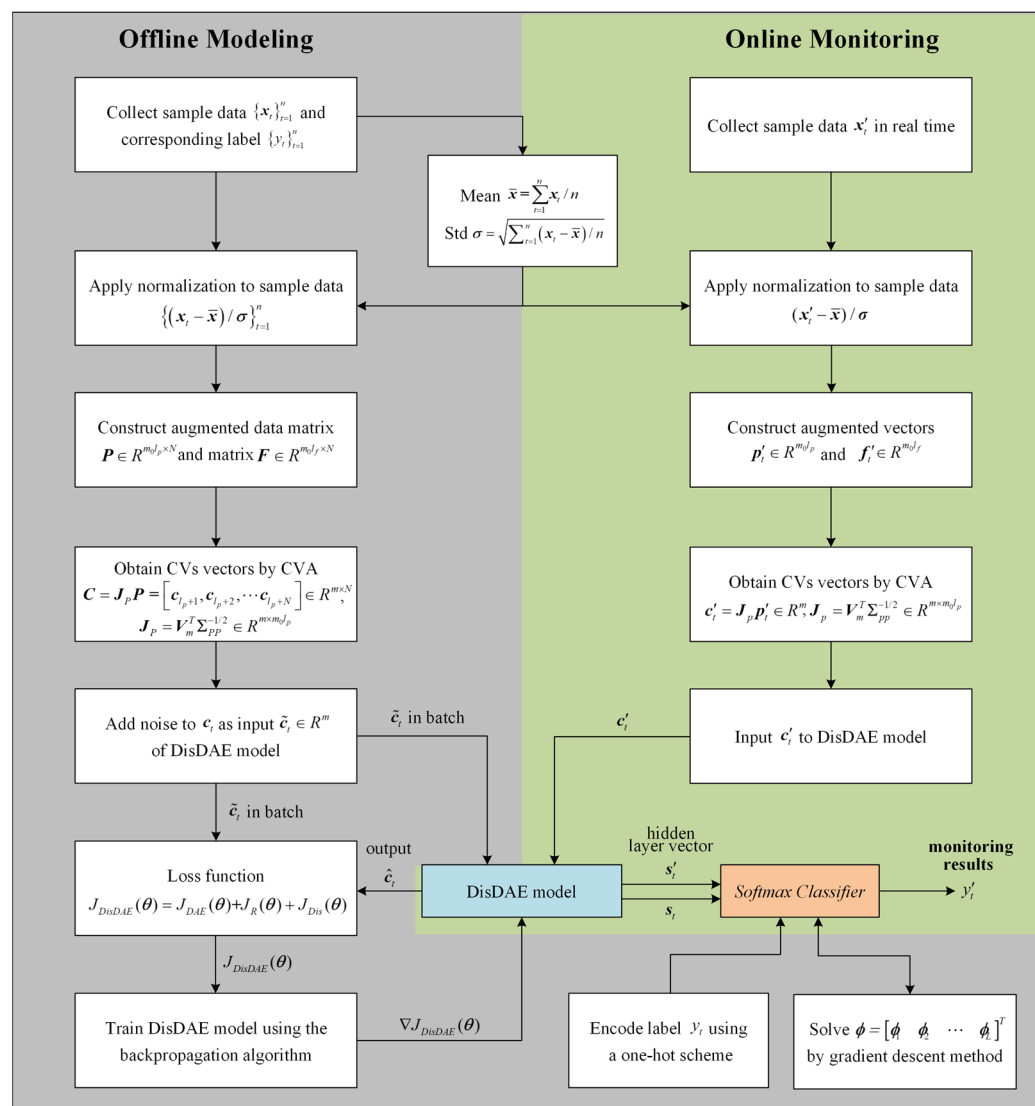


Figure 3. The flowchart of CVA-DisDAE method. There are two phases in the practical application of the CVA-DisDAE method. The DisDAE model and Softmax classifier should be trained in the offline modeling phase before diagnosing the real-time sample in the online monitoring phase.

3. Case Study

3.1. Tennessee Eastman Benchmark Process

This study utilizes the well-known Tennessee Eastman (TE) chemical process [44] for the case analysis. The TE process simulator, which can be accessed and downloaded online (<https://depts.washington.edu/control/LARRY/TE/download.html> (accessed on 31 October 2024)), includes 21 predefined process faults, known as IDVs [45]. In this context, IDV 0 indicates normal operation, while IDVs 1–21 represent various fault conditions. Each IDV scenario provides a training set of 480 samples and a test set of 800 samples, with data comprising 41 measured variables and 11 manipulated variables per sample. Of these, 33 variables are sampled every 3 min, while the remaining ones are recorded at 6 or 15 min intervals. To address data imbalance, this study focuses on the 33 variables sampled every 3 min as the feature variables, using a time interval of $\Delta t = 3$ min.

To evaluate the fault classification performance of the proposed model, several recurring IDV cases were combined into a single dataset, following a method similar to [46]. Table 1 presents three distinct datasets for training and testing, named TE1, TE2, and TE3. Each dataset contains 2400 training samples and 4000 test samples. TE1 and TE2 are used to assess the model's capability to classify various fault types, while TE3, which includes

both normal and faulty cases, is employed to evaluate the model's ability to detect faults without generating false alarms. In Table 1, n_{tr} denotes the number of training samples, n_{tst} indicates the number of test samples, and m_0 represents the number of feature variables in both the training and testing datasets.

Table 1. Configuration of datasets TE1, TE2, and TE3.

Dataset	IDVs	n_{tr}	n_{tst}	m_0
TE1	1, 2, 6, 7, 8	2400	4000	33
TE2	1, 2, 6, 8, 13	2400	4000	33
TE3	0, 1, 2, 4, 7	2400	4000	33

3.2. Model Optimization Strategy

To evaluate the performance of the proposed method and to verify the improvement in fault classification capability for each model component, the D-DAE, D-DisDAE, CVA-DAE, and CVA-DisDAE are trained and tested on the above three datasets, where D-DAE is a denoising autoencoder with a dynamic augmented matrix of input, D-DisDAE is a D-DAE with the LDA penalty term integrated into the loss function, and CVA-DAE is a denoising autoencoder for past data matrix and future data matrix processed by CVA algorithm as input. Note that all models should be fine-tuned to the optimum for performance comparisons. The hyperparameters of these models are listed in Table 2.

Table 2. The hyperparameters of D-DAE, D-DisDAE, CVA-DAE, and CVA-DisDAE.

Model	m_s	σ^2	m	d	l_p	l_f	σ_d^2
D-DAE	✓	✓		✓			
D-DisDAE	✓	✓		✓			✓
CVA-DAE	✓	✓	✓		✓	✓	
CVA-DisDAE	✓	✓	✓		✓	✓	✓

Considering there are slight differences in the hyperparameters of the models, and that CVA-DAE and CVA-DisDAE have relatively more hyperparameters, it is difficult to optimize these parameters at the same time. As a result, the sequential optimization strategy is used in this case. Firstly, D-DAE with the fewest parameters is optimized, and then it is the turn of D-DisDAE, CVA-DAE, and CVA-DisDAE in order. Specifically, the five best-performing parameters/parameter combinations of the preceding model constitute a set, which is later combined with additional parameters of the subsequent model and used as the model configurations to be evaluated.

Note that the DAE-related hyperparameters (m_s and σ^2) of all models should be consistent because their influence on performance is beyond the scope of this paper. m_s and σ^2 were originally intended to be optimized using the sequential optimization strategy, just like the other parameters. However, to focus on evaluating the proposed model's core components, all autoencoders in these models are for single hidden layer structures, with the number of hidden layer nodes m_s of all models set to 100, and the covariance of Gaussian noise σ^2 set to 0.01. These values are commonly used in the literature [34,47] and have been found effective in initial experiments, indicating that they are appropriate choices for this study. In addition, D-DAE and D-DisDAE construct the augmented matrix X_D based on the window size d , refer to Equation (11), while CVA-DAE and CVA-DisDAE construct the past data matrix P and future data matrix F which represent past samples and future samples based on time lags l_p and l_f , respectively, (refer to Equations (12) and (13)). When $l_p = d$ and $l_f = 1$, matrices P and X_D contain the exact same samples, so the time lags l_p of CVA-DAE and CVA-DisDAE will be set to the optimal window size d of the D-DAE model, which maintains consistency in the inputs of all models. At the same time, another parameter l_f will be set to 1.

3.3. Evaluation of Classification Performance on TE1 and TE2

Instead of the AIC criterion, the grid search is adopted to determine the d of the D-DAE model. Given that the sample dimension m_0 of the TE process standard dataset at each moment is 33, the number of input layer nodes of D-DAE or D-DisDAE is $(33 \times d)$. If d is too large, the reasonable number of hidden layer nodes will also become correspondingly large which increases the model complexity. Accordingly, d is set to $P_d^{DAE} = [4, 30] \cap \mathbb{Z}$, $\mathbb{Z}$ is the integer set (if $d \leq 3$, $33 \times d < m_s$, there is no need for dimensionality reduction, if $d > 30$, $\Delta T = d \times \Delta t > 90$ min, the dynamic correlation between two samples can be considered weak enough), and the grid search approach is applied to decide the optimal parameter d for the D-DAE model when processing datasets TE1 and TE2.

The diagnostic accuracy (DA) of the D-DAE models assigned with different d are presented in Table 3. Among them, the best diagnostic accuracy in processing the TE1 and TE2 datasets is 97.26% and 95.16% with d of 18 and 27, respectively.

Table 3. Diagnostic accuracy of the D-DAE models assigned with different d on TE1 and TE2.

TE1				TE2			
DA	d	DA	d	DA	d	DA	d
0.940759494	9	0.957717250	28	0.909160305	13	0.934278351	23
0.950440252	4	0.959225806	24	0.918631179	10	0.939074550	21
0.951828499	6	0.960975610	20	0.921473952	12	0.940983607	6
0.951969504	12	0.961240310	25	0.926010101	7	0.942199488	17
0.952356688	14	0.961381074	17	0.928050314	4	0.943042913	30
0.953001277	16	0.962113402	23	0.928172589	11	0.943260591	20
0.954314721	11	0.962288687	30	0.928957529	22	0.943741935	24
0.954440154	22	0.963212435	27	0.929171975	14	0.945265888	28
0.954797980	7	0.964935065	29	0.929709229	8	0.945714286	29
0.954993679	8	0.966364812	26	0.929722922	5	0.945736434	25
0.955415617	5	0.966666667	19	0.930601793	18	0.946442432	26
0.955893536	10	0.969387755	15	0.930612245	15	0.951025641	19
0.956997455	13	0.972599232	18	0.932151899	9	0.951554404	27
0.957069409	21			0.933333333	16		

When optimizing D-DisDAE, the common parameters shared by D-DisDAE and D-DAE should be fixed. To ensure that the discriminant penalty term σ_d^2 is maintained at the same order of magnitude as the reconstruction error in the loss function, σ_d^2 will be set to $P_{\sigma_d^2}^{DisDAE} = \{10^1, 10^2, 10^3\}$, respectively. Based on the sequential optimization strategy, the sets $P_{d_{opt}}^{DAE_TE1} = \{29, 26, 19, 15, 18\}$ and $P_{d_{opt}}^{DAE_TE2} = \{29, 25, 26, 19, 27\}$ consisting of five parameters for best performance, separately, are combined with the $P_{\sigma_d^2}^{DisDAE}$ as the sets of parameters (d, σ_d^2) to be evaluated on TE1 and TE2, where $P_{(d, \sigma_d^2)}^{DisDAE_TE1} = P_{\sigma_d^2}^{DisDAE} \times P_{d_{opt}}^{DAE_TE1}$, $P_{(d, \sigma_d^2)}^{DisDAE_TE2} = P_{\sigma_d^2}^{DisDAE} \times P_{d_{opt}}^{DAE_TE2}$. For the sake of brevity, only the 10 groups of parameters with the best performance of D-DisDAE and the corresponding diagnostic accuracy are listed in Table 4.

The CVA-DAE is optimized in the same way as the D-DisDAE. Parameters (l_p, l_f) are set to $P_{(l_p, l_f)}^{CVA-DAE_TE1} = P_{d_{opt}}^{DAE_TE1} \times \{1\}$ and $P_{(l_p, l_f)}^{CVA-DAE_TE2} = P_{d_{opt}}^{DAE_TE2} \times \{1\}$, respectively. In the CVA algorithm involved in CVA-DAE, the singular values corresponding to canonical variables are positively correlated with the cross-correlation of canonical variables, they are sorted in descending order and the first m dimensions of them are preserved. The magnitude of m affects the dimensionality of the canonical variables vector extracted by CVA; that is, a small m can cause information omission, and a large m can cause information redundancy, which is contrary to the original intention of dimensionality reduction with CVA.

Table 4. Diagnostic accuracy of the D-DisDAE models with different settings on TE1 and TE2.

TE1			TE2		
DA	d	σ_d^2	DA	d	σ_d^2
0.968367347	15	10^3	0.938242894	25	10^2
0.969210867	26	10^3	0.939220779	29	10^3
0.970153061	15	10^1	0.940155440	27	10^2
0.970909091	29	10^2	0.942820181	26	10^2
0.971282051	19	10^1	0.943076923	19	10^2
0.972343150	18	10^2	0.944702842	25	10^3
0.972833118	26	10^2	0.946373057	27	10^1
0.972855314	18	10^1	0.951025641	19	10^1
0.973214286	15	10^2	0.952196382	25	10^1
0.974871795	19	10^2	0.954285714	29	10^1

According to [47], the m in CVA should be less than l_p , hence for the TE1 dataset, $m \leq \max(P_{d_{opt}}^{DAE_TE1}) = 29$, and the same grid search approach is applied to decide the optimal parameter combination (l_p, l_f, m) of CVA-DAE in the set $P_{(l_p, l_f, m)}^{CVA-DAE_TE1} = P_{(l_p, l_f)}^{CVA-DAE_TE1} \times P_m^{CVA-DAE_TE1}$, where $P_m^{CVA-DAE_TE1} = [2, 29] \cap \mathbb{Z}$. Likewise, the optimal parameter combination is obtained in $P_{(l_p, l_f, m)}^{CVA-DAE_TE2} = P_{(l_p, l_f)}^{CVA-DAE_TE2} \times P_m^{CVA-DAE_TE2}$ when processing the TE2 dataset, where $P_m^{CVA-DAE_TE2} = [2, 29] \cap \mathbb{Z}$. The 10 parameter combinations for optimal performance and corresponding rates for CVA-DAE are listed in Table 5.

Table 5. Diagnostic accuracy of the CVA-DAE models with different settings on TE1 and TE2.

TE1			TE2		
DA	l_p	m	DA	l_p	m
0.975160051	18	28	0.873056995	27	8
0.976666667	19	27	0.874352332	27	20
0.976952625	18	21	0.880519481	29	8
0.981112549	26	25	0.889230769	19	15
0.983699871	26	26	0.891591203	26	19
0.984734799	26	24	0.920000000	29	17
0.986804657	26	28	0.931007752	25	20
0.987792208	29	15	0.931606218	27	15
0.988051948	29	17	0.935751295	27	19
0.988571429	29	14	0.953488372	25	19

As for CVA-DisDAE, its parameters $(l_p, l_f, m, \sigma_d^2)$ are the union of parameters (l_p, l_f, m) and (d, σ_d^2) of CVA-DAE and D-DisDAE (d is equivalent to l_p, l_f). So, the combinations of parameters to be evaluated for CVA-DisDAE on TE1 and TE2 are $P_{(l_p, l_f, m, \sigma_d^2)}^{CVA-DisDAE_TE1} = P_{(l_p, l_f, m)_{opt}}^{CVA-DAE_TE1} \times P_{\sigma_d^2}^{DisDAE}$ and $P_{(l_p, l_f, m, \sigma_d^2)}^{CVA-DisDAE_TE2} = P_{(l_p, l_f, m)_{opt}}^{CVA-DAE_TE2} \times P_{\sigma_d^2}^{DisDAE}$, where $P_{(l_p, l_f, m)_{opt}}^{CVA-DAE_TE1} = P_{(l_p, l_f)_{opt}}^{CVA-DAE_TE1} \times P_{m_{opt}}^{CVA-DAE_TE1}$, $P_{(l_p, l_f, m)_{opt}}^{CVA-DAE_TE2} = P_{(l_p, l_f)_{opt}}^{CVA-DAE_TE2} \times P_{m_{opt}}^{CVA-DAE_TE2}$, $P_{\sigma_d^2}^{DisDAE} = [14, 28] \cap \mathbb{Z}$ and $P_{(l_p, l_f, m)_{opt}}^{CVA-DAE_TE2} = P_{(l_p, l_f)_{opt}}^{CVA-DAE_TE2} \times P_{m_{opt}}^{CVA-DAE_TE2}$, $P_{m_{opt}}^{CVA-DAE_TE2} = [15, 20] \cap \mathbb{Z}$. The 10 parameter combinations for optimal performance and corresponding diagnostic accuracy of CVA-DisDAE are listed in Table 6.

Table 6. Diagnostic accuracy of the CVA-DisDAE method with different settings on TE1 and TE2.

TE1				TE2			
DA	l_p	m	σ_d^2	DA	l_p	m	σ_d^2
0.988051948	29	14	10^2	0.947477361	26	8	10^1
0.988051948	29	24	10^3	0.948186528	27	15	10^3
0.988051948	29	20	10^2	0.948571429	29	20	10^3
0.988051948	29	17	10^2	0.951295337	27	20	10^3
0.988311688	29	19	10^2	0.951617076	26	8	10^2
0.98831169	29	15	10^2	0.952072539	27	19	10^1
0.989090909	29	25	10^3	0.952196382	25	20	10^3
0.989090909	29	24	10^2	0.957142857	29	19	10^1
0.990129870	29	26	10^2	0.958602846	26	8	10^3
0.990129870	29	22	10^2	0.959480519	29	15	10^3

3.4. Evaluation of False Alarm on TE3

The pursuit of the highest diagnostic accuracy is the priority goal of fault detection methods, but this may result in overfitting and cause the model to misclassify normal cases as faults, i.e., false alarms. Therefore, in order to avoid the illusion of high accuracy caused by model overfitting, and to verify the capability of each model to correctly distinguish faults while suppressing false alarms, the diagnostic accuracy and false alarm rates (FARs) of the above four models on the TE3 dataset were evaluated.

The same sequential optimization strategy is used in this part, but the criterion for evaluating the optimal hyperparameters is $R = 0.6 \times DA + 0.4 \times (-FAR)$, representing the weighted mean of the diagnostic accuracy and false alarm rate, which is different from the previous one. A portion of the performance of D-DAE, D-DisDAE, CVA-DAE, and CVA-DisDAE on TE3 is presented in Tables 7–10.

Table 7. Performance of the D-DAE models assigned with different d on TE3.

DA	FAR	d	R
0.963468869	0.113087675	12	0.532846252
0.972222222	0.125000000	7	0.533333333
0.970512821	0.115384615	19	0.536153846
0.967363752	0.108998733	10	0.536818758
0.974936548	0.091370558	11	0.548413706

Table 8. Performance of the D-DisDAE models with different settings on TE3.

DA	FAR	d	σ_d^2	R
0.885931559	0.044359949	10	10^1	0.513814956
0.889911168	0.036802030	11	10^1	0.519225888
0.969264892	0.141951838	10	10^3	0.524778200
0.909148666	0.030495553	12	10^1	0.533290978
0.985164141	0.143939394	7	10^3	0.533522727

Table 9. Performance of the CVA-DAE models with different settings on TE3.

DA	FAR	l_p	m	R
0.983270202	0.117424242	7	10	0.542992424
0.975285171	0.089987326	10	9	0.549176172
0.984935897	0.100000000	19	9	0.550961538
0.977474619	0.076142132	11	9	0.556027919
0.987373737	0.082070707	7	9	0.559595960

Table 10. Performance of the CVA-DisDAE method with different settings on TE3.

DA	FAR	l_p	m	σ_d^2	R
0.990799492	0.000000000	11	9	10^3	0.594479695
0.993011436	0.000000000	12	9	10^3	0.595806861
0.995265152	0.001262626	7	9	10^3	0.596654040
0.996192893	0.001269036	11	10	10^3	0.597208122
0.998729352	0.000000000	12	10	10^3	0.599237611

3.5. Comparison with LSTM on TE3

Long short-term memory (LSTM) networks are commonly used for sequence data processing tasks [48,49] due to their ability to capture long-term dependencies, allowing them to learn the dynamic characteristics of data. Additionally, neural networks are well-suited for handling nonlinear relationships and complex patterns, making them highly applicable for process monitoring and fault diagnosis in dynamic, nonlinear industrial processes. To demonstrate the superiority of CVA-DisDAE, this section compares the performance of the two methods on the TE3 dataset.

The LSTM network feeds the hidden state at the final time step of the sequence into a fully connected layer with the same number of neurons as the fault categories, which is then connected to a softmax classifier for fault diagnosis. To ensure a fair comparison, both models are configured with the same neural network size, input data volume, and training settings. Specifically, the number of hidden units in the LSTM's forget gate, input gate, output gate, and candidate memory cell is set to $m_s = 100$, and the input sequence length is $l_p + l_f = 13$. Both models are trained for 100 epochs, with a batch size of 20, and optimized using the Adam solver. Due to the randomness in neural network parameter initialization and input data order during training, both models were independently trained and tested 10 times. The average values of the performance metrics are shown in Table 11. Additionally, the confusion matrices with the highest R-value from the test sets of both models are provided to offer a more comprehensive view of their performance on the multiclass classification task in Figures 4 and 5.

Table 11. Performance of the CVA-DisDAE method and the LSTM method on TE3.

Model	DA	FAR	R
LSTM	0.973697586	0.001016518	0.583811944
CVA-DisDAE	0.998017789	0.000127064	0.598759847

True Labels	IDV0	787				
	IDV1		787	13		
	IDV2		11	789		
	IDV4			4	757	39
	IDV7				25	723
	IDV0	IDV1	IDV2	IDV4	IDV7	
	Predicted Labels					

Figure 4. Confusion matrix of the LSTM method on TE3.

IDV0	787				
IDV1		787			
IDV2			787		
IDV4				787	
IDV7	4				783
	IDV0	IDV1	IDV2	IDV4	IDV7

Predicted Labels

Figure 5. Confusion matrix of the CVA-DisDAE method on TE3.

3.6. Discussion

Provided that all models are given optimal parameters, comparing the diagnostic accuracy of D-DAE and D-DisDAE on the TE1 and TE2 datasets in Tables 3 and 4, the average disparities are 0.2% and 0.3%. It can be concluded that if the adjusting parameter of the penalty term is taken as reasonable, the discriminant penalty term in *DisDAE* can impel *DAE* to extract the feature vectors from the input vectors that are more conducive to accurate classification.

A comparison of diagnostic accuracy among D-DAE, CVA-DAE, D-DisDAE, and CVA-DisDAE reveals that with an appropriately chosen value of m , the CVA algorithm not only reduces the dimensionality of the augmented matrix, thereby lowering the complexity of the *DAE* and *DisDAE* models, but also enhances their fault detection performance. Analyzing the diagnostic accuracy across all models on the TE1 and TE2 datasets shows that the CVA-DisDAE method achieves the highest average diagnostic accuracy. This confirms that the integration of the CVA method and LDA penalty term significantly boosts fault classification ability, leading to substantial performance improvements in CVA-DisDAE compared to D-DAE.

Evaluating all models on the TE3 dataset demonstrates that while D-DAE achieves a diagnostic accuracy of 97.5%, it also has a high false alarm rate of 9.1%. D-DisDAE improves upon D-DAE in terms of accuracy but also increases the false alarm rate. CVA-DAE, on the other hand, improves both diagnostic accuracy and false alarm rate. Among all models, CVA-DisDAE exhibits the strongest fault detection capability while also minimizing false alarms, outperforming the other three models in both aspects.

Furthermore, the performance comparison on the TE3 dataset shows that CVA-DisDAE outperforms LSTM in DA and FAR, with the sample counts in the confusion matrix more concentrated along the diagonal, indicating stronger fault diagnosis capabilities.

4. Conclusions

In this article, a novel process monitoring approach called CVA-DisDAE based on improved *DAE* and CVA has been proposed to address problems of dynamic behaviors and nonlinear relations that are inherent in industrial processes. The proposed CVA-DisDAE differs from the traditional approach in two aspects. First, CVA has been employed to reduce data dimensionality and information redundancy by maximizing the correlations between past and future observations, so that the process dynamics can be captured maximally. Second, a discriminative strategy has been used to achieve a semi-supervised denoising autoencoder for exact feature extraction by incorporating an LDA penalty into *DAE*'s loss function, so that it can maximize the between-class separability and minimize the within-class variability simultaneously, and the tradeoff between unsupervised learning

and supervised learning can be adjusted by a weighting parameter. In addition, a Softmax classifier was adopted in *DisDAE* to perform fault classification, and a weight decay term was added to avoid over-fitting in the training process of Softmax.

In the case study part, the proposed CVA-*DisDAE* has been applied to the Tennessee Eastman benchmark process, demonstrating its superior performance. An ablation experiment comparing CVA-*DisDAE* with D-*DAE*, D-*DisDAE*, and CVA-*DAE* was conducted, which validated that the introduction of the CVA algorithm and the LDA penalty term is beneficial for enhancing fault classification ability. The combination of the two techniques can greatly improve the performance of the autoencoder. In addition, the false alarm evaluation shows the CVA-*DisDAE* can achieve high diagnostic accuracy while keeping the false alarm rate at a minimal level. In comparison, with the state-of-the-art LSTM method, CVA-*DisDAE* demonstrated superior fault detection performance under the same neural network size, input data volume, and training settings.

Future research work will focus on extending the proposed *DisDAE* to its stacked network version and how an adaptive algorithm can be established to find the optimal parameters of CVA-*DisDAE*, such as time lags of CVA, tradeoff weights of LDA, anti-overfitting factor of Softmax, and so on.

Author Contributions: Conceptualization, J.L. and D.L.; Methodology, J.L., D.L., Y.Z. and J.F.; Software, J.L., D.L. and Y.Z.; Validation, J.L., D.L., Y.Z. and J.F.; Formal Analysis, J.L. and D.L.; Investigation, Y.Z. and J.F.; Resources, J.L. and D.L.; Data Curation, J.L., D.L. and Y.Z.; Writing—Original Draft Preparation, J.L., D.L., Y.Z. and J.F.; Writing—Review and Editing, J.L., D.L., Y.Z. and J.F.; Visualization, D.L. and Y.Z.; Supervision, J.L. and J.F.; Project Administration, D.L. and Y.Z.; Funding Acquisition, J.L., D.L., Y.Z. and J.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the State Key Laboratory of Nuclear Power Safety Monitoring Technology Equipment and the Fundamental Research Funds for the Central Universities (Zhejiang University NGICS Platform).

Data Availability Statement: The data presented in this study is available at [[https://doi.org/10.1016/0098-1354\(94\)00057-U](https://doi.org/10.1016/0098-1354(94)00057-U)] in [44]. These data were derived from the following resources available in the public domain: <https://depts.washington.edu/control/LARRY/TE/download.html> (accessed on 15 October 2024).

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

References

- Si, Y.; Wang, Y.; Zhou, Y. Key-performance-indicator- related process monitoring based on improved kernel partial least squares. *IEEE Trans. Ind. Electron.* **2021**, *68*, 2626. [[CrossRef](#)]
- Ma, X.; Si, Y.; Yuan, Z.; Qin, Y.; Wang, Y. Multi- step dynamic slow feature analysis for industrial process monitoring. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 9535. [[CrossRef](#)]
- Ge, Z.; Song, Z.; Gao, F. Review of Recent Research on Data-Based Process Monitoring. *Ind. Eng. Chem. Res.* **2013**, *52*, 3534. [[CrossRef](#)]
- Lou, Z.; Wang, Y.; Si, Y.; Lu, S. A novel multivariate statistical process monitoring algorithm: Orthonormal subspace analysis. *Automatica* **2022**, *138*, 110148. [[CrossRef](#)]
- Ku, W.; Storer, R.H.; Georgakis, C. Disturbance detection and isolation by dynamic principal component analysis. *Chemom. Intell. Lab. Syst.* **1995**, *30*, 179. [[CrossRef](#)]
- Li, S.; Chen, Y.; Tong, C. Multivariate statistical process monitoring based on principal discriminative component analysis. *J. Frankl. Inst.* **2021**, *15*, 7900. [[CrossRef](#)]
- Lou, Z.; Wang, Y. Multimode continuous processes monitoring based on hidden semi-markov model and principal component analysis. *Ind. Eng. Chem. Res.* **2017**, *46*, 13800. [[CrossRef](#)]
- Sheriffa, M.Z.; Mansouric, M.; Karima, N.; Nounouc, H.; Nounoub, M. Fault detection using multiscale pca-based moving window GLRT. *J. Process Control* **2017**, *54*, 47. [[CrossRef](#)]
- He, S.; Wang, Y.; Liu, C. Modified partial least square for diagnosing key-performance-indicator-related faults. *Can. J. Chem. Eng.* **2018**, *2*, 444. [[CrossRef](#)]
- Peng, X.; Ding, S.; Du, W.; Zhong, W.; Qian, F. Distributed process monitoring based on canonical correlation analysis with partly-connected topology. *Control Eng. Pract.* **2020**, *101*, 104500. [[CrossRef](#)]
- Fazai, R.; Mansouri, M.; Abodayeh, K.; Nounou, H.; Nounou, M. Online reduced kernel pls combined with glrt for fault detection in chemical systems. *Process Saf. Environ. Prot.* **2019**, *128*, 228. [[CrossRef](#)]

12. Harkat, M.; Mansouri, M.; Nounou, N. Fault detection of uncertain chemical processes using interval partial least squares-based generalized likelihood ratio test. *Inf. Sci.* **2019**, *490*, 265.
13. Yin, S.; Wang, G.; Gao, H. Data-driven process monitoring based on modified orthogonal projections to latent structures. *IEEE Trans. Control Syst. Technol.* **2015**, *4*, 1480. [\[CrossRef\]](#)
14. Ji, H.; Sheng, N.; Liu, H.; Huang, K. Fault detection and isolation for dynamic non-stationary processes with stationary subspace-based canonical variate analysis. *Chem. Eng. Sci.* **2024**, *295*, 120196. [\[CrossRef\]](#)
15. Zhang, S.; Bao, X.; Wang, S. Common canonical variate analysis (CCVA) based modeling and monitoring for multimode processes. *Chem. Eng. Sci.* **2023**, *271*, 118581. [\[CrossRef\]](#)
16. Shang, L.; Gu, Y.; Tang, Y.; Fu, H.; Hua, L. Recursive ensemble canonical variate analysis for online incipient fault detection in dynamic processes. *Measurement* **2023**, *220*, 113411. [\[CrossRef\]](#)
17. Wang, J.; Yu, H.; Li, X. Efficient Iterative Dynamic Kernel Principal Component Analysis Monitoring Method for the Batch Process with Super-large-scale Data Sets. *ACS Omega* **2021**, *6*, 9989. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Luo, X.; Li, X.; Wang, Z.; Liang, J. Discriminant autoencoder for feature extraction in fault diagnosis. *Chemom. Intell. Lab. Syst.* **2019**, *192*, 103814. [\[CrossRef\]](#)
19. Alonso, S.; Morán, A.; Pérez, D.; Prada, M.; Fuertes, J.; Domínguez, M. Gap imputation in related multivariate time series through recurrent neural network-based denoising autoencode. *Integr. Comput.-Aid Eng.* **2024**, *31*, 157. [\[CrossRef\]](#)
20. Zhang, X.; Zhang, H.; Song, Z. Feature-Aligned Stacked Autoencoder: A Novel Semisupervised Deep Learning Model for Pattern Classification of Industrial Faults. *IEEE Trans. Artif. Intell.* **2023**, *4*, 592. [\[CrossRef\]](#)
21. Snoek, J.; Adams, R.P.; Edu, H.L. Nonparametric Guidance of Autoencoder Representations using Label Information. *J. Mach. Learn. Res.* **2013**, *13*, 2567.
22. Zhang, Z.; Jiang, T.; Li, S.; Yang, Y. Automated feature learning for nonlinear process monitoring – An approach using stacked denoising autoencoder and k-nearest neighbor rule. *J. Process Control* **2018**, *64*, 49. [\[CrossRef\]](#)
23. Hu, Z.; Zhao, H.; Peng, J. Low-rank reconstruction-based autoencoder for robust fault detection. *Control Eng. Pract.* **2022**, *123*, 105156. [\[CrossRef\]](#)
24. Yu, W.; Zhao, C. Robust Monitoring and Fault Isolation of Nonlinear Industrial Processes Using Denoising Autoencoder and Elastic Net. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 1083. [\[CrossRef\]](#)
25. Zhang, C.; Zhou, L.; Zhao, Y.; Zhu, S.; Liu, F.; He, Y. Noise reduction in the spectral domain of hyperspectral images using denoising autoencoder methods. *Chemom. Intell. Lab. Syst.* **2020**, *203*, 104063. [\[CrossRef\]](#)
26. Yu, J.; Yan, X. Multiscale intelligent fault detection system based on agglomerative hierarchical clustering using stacked denoising autoencoder with temporal information. *Appl. Soft Comput.* **2020**, *95*, 106525. [\[CrossRef\]](#)
27. Fu, X.; Liu, W.; Xu, Y.; Cui, L. Combine HowNet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis. *Neurocomputing* **2017**, *241*, 18. [\[CrossRef\]](#)
28. Gao, S.; Zhang, Y.; Jia, K.; Lu, J.; Zhang, Y. Single sample face recognition via learning deep supervised autoencoders. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2108. [\[CrossRef\]](#)
29. Huang, R.; Liu, C.; Li, G.; Zhou, J. Adaptive Deep Supervised Autoencoder Based Image Reconstruction for Face Recognition. *Math. Probl. Eng.* **2016**, *2016*, 6795352. [\[CrossRef\]](#)
30. Martinez, M.; Kak, A.C. PCA versus LDA. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 228. [\[CrossRef\]](#)
31. Yu, J. Localized Fisher discriminant analysis based complex chemical process monitoring. *AIChE J.* **2011**, *57*, 1817. [\[CrossRef\]](#)
32. Lu, P.; Jiang, X.; Zhang, Y.; Liu, X.; Cai, Z.; Jiang, J.; Plaza, A. Unsupervised Linear Discriminant Analysis for Feature Extraction and Classification of Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1.
33. Wu, B.; Tang, W.; Zhou, J.; Jia, H.; Shen, H.; Qi, Z. Near-infrared spectroscopy combined with fuzzy fast pseudoinverse linear discriminant analysis to discriminate mee tea grades. *Heliyon* **2024**, *5*, 27732. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Evan, L.; Russell, L.; Chiang, H.; Richard, D. Braatz, Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemom. Intell. Lab. Syst.* **2000**, *51*, 81.
35. Jiang, B.; Huang, D.; Zhu, X.; Yang, F.; Braatz, R.D. Canonical variate analysis-based contributions for fault identification. *J. Process. Control* **2015**, *26*, 17. [\[CrossRef\]](#)
36. Benson, S.Y.; Choy, S.K.; Yu, K.W. Linear discriminant analysis with trimmed and difference distribution modeling. *Knowl.-Based Syst.* **2024**, *299*, 112093.
37. Urtasun, R.; Darrell, T. Discriminative Gaussian process latent variable model for classification. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; p. 927.
38. Wang, X.; Wu, P. Nonlinear Dynamic Process Monitoring Based on Ensemble Kernel Canonical Variate Analysis and Bayesian Inference. *ACS Omega* **2022**, *7*, 18904–18921. [\[CrossRef\]](#)
39. Samuel, R.T.; Cao, Y. Kernel Canonical Variate Analysis for Nonlinear Dynamic Process Monitoring. *IFAC-PapersOnLine* **2015**, *48*, 605. [\[CrossRef\]](#)
40. Tang, Q.; Liu, Y.; Chai, Y.; Huang, C.; Liu, B. Dynamic process monitoring based on canonical global and local preserving projection analysis. *J. Process. Control* **2021**, *106*, 221. [\[CrossRef\]](#)
41. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and Composing Robust Features with Denoising Autoencoders. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, New York, NY, USA, 5–9 July 2008; Association for Computing Machinery: Melbourne, Australia; p. 1096.

42. Liang, J.; Liu, R. Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. In Proceedings of the 2015 8th International Congress on Image and Signal Processing (CISP), Shenyang, China, 14–16 October 2015; p. 697.
43. Zhang, M.; Li, X.; Wang, R. Incipient Fault Diagnosis of Batch Process Based on Deep Time Series Feature Extraction. *Arab. J. Sci. Eng.* **2021**, *46*, 10125. [[CrossRef](#)]
44. Lyman, P.R.; Georgakis, C. Plant-wide control of the Tennessee Eastman problem. *Comput. Chem. Eng.* **1995**, *19*, 321.
45. Downs, J.J.; Vogel, E.F. A plant-wide industrial process control problem. *Comput. Chem. Eng.* **1993**, *17*, 245. [[CrossRef](#)]
46. Zhu, D.Y.; Shen, J.Y.; Huang, W.P.; Liang, J. Fault classification based on modified active learning and weighted SVM. *J. Zhejiang Univ. (Eng. Sci.)* **2017**, *51*, 697.
47. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [[CrossRef](#)]
48. Chen, Z.; Liang, K.; Ding, S.X.; Yang, C.; Peng, T.; Yuan, X. A comparative study of deep neural network-aided canonical correlation analysis-based process monitoring and fault detection methods. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6158–6172. [[CrossRef](#)]
49. Wang, J. Fault Diagnosis Method of Marine Sewage Treatment Equipment Based on Cloud Model and Improved LSTM Algorithm. In Proceedings of the 2024 3rd International Conference on Energy and Electrical Power Systems (ICEEPS), Guangzhou, China, 14–16 July 2024; pp. 950–955.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.