processes

MDPI

Article

MOLA: Enhancing Industrial Process Monitoring Using a Multi-Block Orthogonal Long Short-Term Memory Autoencoder

Fangyuan Ma [1,2], Cheng Ji [2], Jingde Wang [2], Wei Sun [2], Xun Tang [3,*] and Zheyu Jiang [1,*]

[1] School of Chemical Engineering, Oklahoma State University, 420 Engineering North, Stillwater, OK 74078, USA; fangyuan.ma@okstate.edu
[2] College of Chemical Engineering, Beijing University of Chemical Technology, Beijing 100029, China; 2024700006@mail.buct.edu.cn (C.J.); sunwei@mail.buct.edu.cn (W.S.)
[3] Cain Department of Chemical Engineering, Louisisna State University, Baton Rouge, LA 70803, USA
* Correspondence: xuntang@lsu.edu (X.T.); zheyu.jiang@okstate.edu (Z.J.)

Abstract: In this work, we introduce MOLA, a multi-block orthogonal long short-term memory autoencoder paradigm, to conduct accurate, reliable fault detection of industrial processes. To achieve this, MOLA effectively extracts dynamic orthogonal features by introducing an orthogonality-based loss function to constrain the latent space output. This helps eliminate the redundancy in the features identified, thereby improving the overall monitoring performance. On top of this, a multi-block monitoring structure is proposed, which categorizes the process variables into multiple blocks by leveraging expert process knowledge about their associations with the overall process. Each block is associated with its specific orthogonal long short-term memory autoencoder model, whose extracted dynamic orthogonal features are monitored by distance-based Hotelling's T^2 statistics and quantile-based cumulative sum (CUSUM) designed for multivariate data streams that are nonparametric and heterogeneous. Compared to having a single model accounting for all process variables, such a multi-block structure significantly improves overall process monitoring performance, especially for large-scale industrial processes. Finally, we propose an adaptive weight-based Bayesian fusion (W-BF) framework to aggregate all block-wise monitoring statistics into a global statistic that we monitor for faults. Fault detection speed and accuracy are improved by assigning and adjusting weights to blocks based on the sequential order in which alarms are raised. We demonstrate the efficiency and effectiveness of our MOLA framework by applying it to the Tennessee Eastman process and comparing the performance with various benchmark methods.

Keywords: process monitoring; fault detection; long short-term memory autoencoder; Bayesian fusion; CUSUM

Citation: Ma, F.; Ji, C.; Wang, J.; Sun, W.; Tang, X.; Jiang, Z. MOLA: Enhancing Industrial Process Monitoring Using a Multi-Block Orthogonal Long Short-Term Memory Autoencoder. *Processes* **2024**, *12*, 2824. https://doi.org/10.3390/pr12122824

Academic Editor: Hector Budman

Received: 9 October 2024
Revised: 28 November 2024
Accepted: 6 December 2024
Published: 9 December 2024

1. Introduction

Effective, reliable process monitoring is essential to ensuring process safety, improving product quality, and reducing operating costs of industrial systems as they continue to expand in scale and complexity [1]. Nowadays, modern chemical plants are equipped with numerous sensors connected to distributed control systems (DCSs), continuously generating massive process data that can be leveraged for data-driven process monitoring in real time [2]. Traditional methods for process monitoring encompass principal component analysis (PCA), partial least squares (PLS), and independent component analysis (ICA), among many others [3]. The idea behind these fault detection methods largely falls in extracting underlying features that characterize process states (e.g., faulty vs. non-faulty) from historical process data and monitoring changes in these extracted features [4]. For instance, PCA leverages linear orthogonal transformations to extract key process features, projecting them in a principal component subspace and a residual subspace, followed by developing a monitoring statistic for each subspace for fault detection [5]. Nevertheless, the relationships among process variables being monitored in modern industrial systems

are often highly nonlinear, and conventional linear methods such as PCA face challenges in effectively capturing these nonlinear relationships. To overcome the intrinsic limitations of conventional linear methods, kernel-based methods, such as kernel PCA, have been proposed [6,7]. While these kernel-based methods can extract nonlinear relationships, identifying and computing the kernel functions can be time-consuming, limiting their capabilities in real-world applications that demand fast real-time fault detection [4]. Furthermore, compared to standard PCA, kernel-based methods exhibit greater sensitivity to noise and outliers [8], potentially deteriorating monitoring accuracy.

Leveraging the recent breakthroughs in deep learning, artificial neural networks (ANNs), which consist of multiple fully connected layers and nonlinear activation functions, have achieved remarkable successes in extracting complex nonlinear features among process variables [9] for process monitoring. Among prevailing ANN-based methods, multi-layer perceptron (MLP), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are some of the most effective and widely-used deep learning architectures for process monitoring [10–12]. Since traditional ANN-based process monitoring methods are essentially supervised classification methods [13], their monitoring performance relies on the availability of a large amount of labeled data, especially faulty data, which are typically limited and hard to acquire in practice [14].

Meanwhile, unsupervised methods, such as the autoencoder (AE), have gathered increasing attention due to their ability to extract features from unlabeled data, thereby presenting a more viable alternative for process monitoring [14]. As illustrated in Figure 1, the core structure of an AE comprises an input layer, an encoder layer, a latent space, a decoder layer, and an output layer. Specifically, an encoder maps the original input data into its latent space consisting of codes (or latent variables) that effectively capture and retain the key data representations. A decoder is then employed to accurately reconstruct the original information from these lower-dimensional embeddings, aiming to reproduce data that are indistinguishable from the original input data [15].

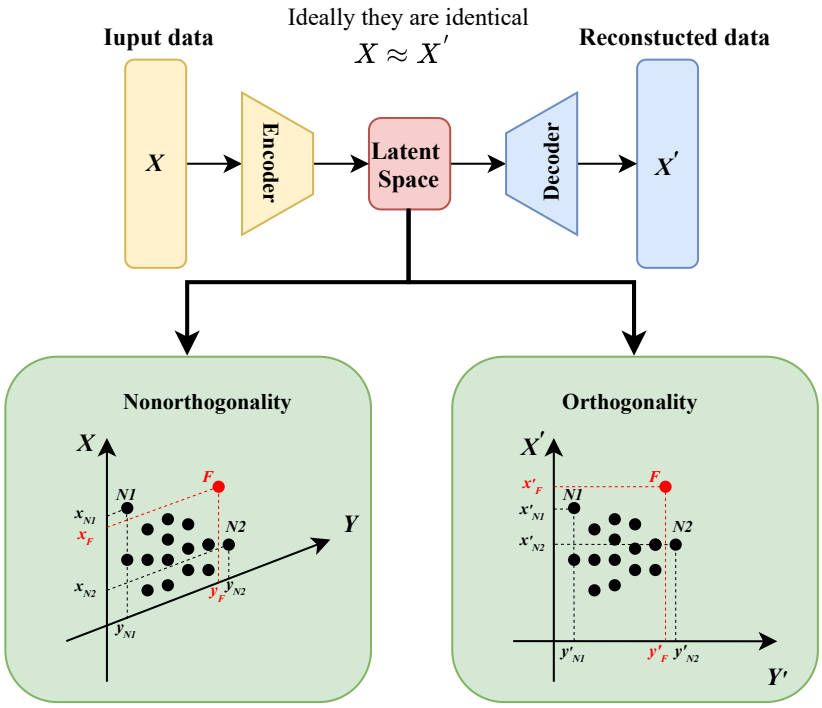

Figure 1. Illustration of the autoencoder structure and feature extraction.

In traditional AE-based process monitoring methods, the reconstruction error is used as the primary objective during training and is monitored for fault detection during deploy-

ment [16]. However, this approach does not explicitly make use of the lower-dimensional embeddings, which inherently represent the underlying process dynamics. Furthermore, solely minimizing the reconstruction error may introduce redundancy among extracted features (see Figure 1). This is illustrated in Figure 1, where the points shown represent the projections of the original input data onto a two-dimensional latent space after passing through the encoder layer. Here, the red point F denotes the projection of a faulty data sample, whereas the rest represent non-faulty data samples. When the extracted features contain redundancy, the two dimensions of the latent space are not orthogonal to each other. In this case, the projections of F onto X- and Y-axes will fall in the range of non-faulty data sample projections, thereby making fault detection more challenging. On the other hand, when the extracted features contain no redundancy, the latent variables are orthogonal to one another. In the same example, the projection of F onto the X'-axis now falls outside of the range of non-faulty data sample projections.

Based on this observation, the orthogonal autoencoder (OAE), which introduces a term characterizing orthogonality of latent space outputs in the loss function, was proposed to extract features that are inherently independent or nonredundant [17]. Cacciarelli et al. then applied the OAE to process monitoring applications by monitoring Hotelling's T^2 statistics of orthogonal latent features [18]. While OAE effectively improves the fault detection performance compared to traditional AE, it still has several limitations. First, actual industrial systems typically involve dynamic behaviors [19], which cannot be captured by existing OAE-based frameworks. Second, Hotelling's T^2 statistic may not be suited for capturing changes and shifts in the distribution of latent features [20]. It also does not explicitly account for the cumulative effects of process anomalies, which can play a crucial role in detecting certain types of faults promptly [21].

Furthermore, one of the unique characteristics of modern chemical process systems is that they typically comprise multiple heavily integrated (via mass, energy, and information flows) yet relatively autonomous subsystems, each with specific process functions such as raw material processing, reaction, and product separations. These subsystems are further disaggregated into smaller unit operations, such as reactors, heat exchangers, and distillation columns. Such structural complexity and hierarchy can pose significant challenges to conventional process monitoring paradigms that rely on a single model to monitor the entire process [22]. As the number of process variables being monitored increases, the number of hyperparameters in the deep learning model increases exponentially, significantly amplifying the training complexity. To address this challenge, a multi-block process monitoring methodology has been proposed for large-scale industrial process systems [23]. The idea is to categorize process variables into various blocks based on the variables' associations with the overall process and their relevance with other process variables, followed by building a process monitoring model for each block. The entire process will be monitored by integrating these block-wise process monitoring models via a data fusion mechanism [24]. Conventional data fusion techniques, such as Bayesian fusion, are static in nature and treat the monitoring statistics from different blocks equally [25]. However, in reality, process anomalies often stem from one block and propagate/spread to others as time progresses, making conventional data fusion techniques inadequate and less effective.

To address the aforementioned challenges, we propose a novel process monitoring framework based on a multi-block orthogonal long short-term memory autoencoder (MOLA). As a significant variant of traditional autoencoders, long short-term memory autoencoder (LSTM AE) implements the LSTM architecture in both the encoder and decoder layers. This allows the extraction of dynamic process features from the time series data. On top of this, we propose the orthogonal LSTM autoencoder (OLAE), which incorporates an orthogonality-based loss function to constrain the latent space output. We also adopt the multi-block monitoring methodology to assign all process variables into several blocks based on process knowledge. A local OLAE model is developed for each block. To effectively detect anomalies of the orthogonal latent features in each block, in

addition to Hotelling's T^2 approach, we also incorporate a quantile-based multivariate cumulative sum (CUSUM) process monitoring method [26], a state-of-the-art approach to monitoring high-dimensional data streams that are nonparametric (i.e., data streams do not necessarily follow any specific distribution) and heterogeneous (i.e., data streams do not necessarily follow the same distribution) [27]. The use of quantile-based multivariate CUSUM successfully overcomes the barrier of Hotelling's T^2 method that overlooks the changes and shifts in the distribution of latent features. Finally, we propose an adaptive weight-based Bayesian fusion (W-BF) framework to effectively aggregate the monitoring results from individual blocks. Our proposed W-BF framework automatically assigns higher weights to blocks based on how early anomalies occur in each block, thereby improving overall fault detection speed and accuracy. Among all these technical advancements, the key contributions of this work are as follows:

1. We introduce a novel autoencoder architecture, OLAE, to extract non-redundant and mutually independent dynamic features. Compared to existing autoencoder designs, OLAE demonstrates superior fault detection performance.
2. We incorporate a state-of-the-art quantile-based multivariate CUSUM to our combined statistics to enable fast, accurate, and robust detection of process anomalies based on the mean shift in the distribution of extracted features.
3. We propose a novel W-BF approach to dynamically adjust the weights assigned to monitoring results from different blocks, which significantly enhances fault detection speed and accuracy.

Datasets from the benchmark Tennessee Eastman process (TEP) problem are used to evaluate our proposed MOLA framework. Some of the key capabilities of MOLA with respect to other related process monitoring methods are summarized in Table 1.

Table 1. A high-level comparison of key capabilities of different process monitoring methods.

Capability	PCA	KPCA	DPCA	AE	LSTM AE	Block PCA	Block LSTM AE	MOLA
Cross-correlation	✓	✓	✓	✓	✓	✓	✓	✓
Dynamic	×	×	✓	×	✓	×	✓	✓
Nonlinearity	×	✓	×	✓	✓	×	✓	✓
Orthogonality	✓	✓	✓	×	×	✓	×	✓
Large-scale monitoring	×	×	×	×	×	✓	✓	✓
Adaptive Weight	×	×	×	×	×	×	×	✓
Distribution monitoring	×	×	×	×	×	×	×	✓

The rest of this paper is organized as follows. Section 2 provides a brief review of LSTM and the quantile-based multivariate CUSUM method, followed by a detailed introduction to the proposed MOLA and the adaptive W-BF frameworks. Section 3 discusses the detailed steps involved in offline training and online monitoring of the proposed process monitoring framework. Next, in Section 4, we showcase the performance of our proposed methodology in the benchmark problem of the TEP, demonstrating its outstanding fault detection speed and accuracy compared to benchmark methods. To conclude, we summarize all the results and learnings and discuss potential improvements for future research in Section 5.

2. Theory and Methods

In this section, we provide the theoretical background of the backbone methods (e.g., LSTM and quantile-based multivariate CUSUM) upon which our proposed framework is based. We then formally introduce our proposed approaches, including OLAE and adaptive weight-based Bayesian fusion (W-BF).

2.1. LSTM

The recurrent neural network (RNN) is a class of neural network architectures specifically designed for time series modeling and prediction. This makes RNN-based methods

particularly attractive in capturing the dynamic features of industrial process data [28]. However, traditional RNNs are prone to the issues of gradient explosion and gradient vanishing when working with long-term time series [29]. To overcome these issues, LSTM, an advanced RNN that uses "gates" to capture both long-term and short-term memory, is introduced, featuring a unique structural unit at its core [30]. As illustrated in Figure 2, an LSTM unit contains three crucial control gates: the forget gate $f(t)$, the input gate $i(t)$, and the output gate $o(t)$. These gates work collectively to ensure that essential information is consistently retained while the less important information is discarded. Therefore, LSTM achieves superior performance in capturing complex temporal dynamics embedded in chemical process systems. For more detailed information and mathematical descriptions about LSTM, readers are encouraged to refer to [31].

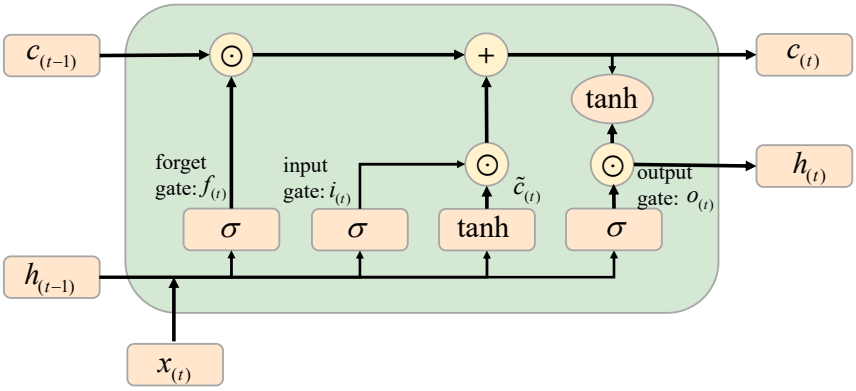

Figure 2. Illustration of LSTM unit architecture featuring forget, input, and output gates.

2.2. OLAE

The LSTM AE is a specialized type of AE that seamlessly integrates LSTM with AE. This hybrid architecture enables efficient encoding and decoding of temporal sequences while capturing long-term dynamic features and dependencies within individual data streams at the same time. Similar to AE, the LSTM AE typically uses the reconstruction error as the loss function for model training. The mean squared error (MSE) is one of the most widely adopted reconstruction error formulations:

$$\text{Loss}_{\text{MSE}} = \frac{1}{K}\sum_{i=1}^{K}(x_i - y_i)^2, \tag{1}$$

where K is the total number of samples, x_i represents the i-th original input vector, and y_i is the corresponding reconstructed output vector by the LSTM AE. Nevertheless, as discussed earlier, the use of only Equation (1) in the loss function can cause redundancies among the latent features, which will adversely affect the performance of LSTM AE in process monitoring tasks.

Therefore, in the OLAE architecture, as shown in Figure 3, we define an orthogonality-based loss function in Equation (2) to constrain the latent space output to generate non-redundant orthogonal latent features:

$$\text{Loss}_{\perp} = \|L_0(W)\|_F^2 + ||C^TC - I||_F^2,\ L_0(W) = \begin{bmatrix} w_1w_1^T & w_1w_2^T & \dots & w_1w_m^T \\ w_2w_1^T & w_2w_2^T & \dots & w_2w_m^T \\ \dots & \dots & \dots & \dots \\ w_mw_1^T & w_mw_2^T & \dots & w_mw_m^T \end{bmatrix}, \tag{2}$$

where $W = [w_1, w_2, \dots, w_m]^T$ is the weight matrix of the FC layer. The loss function $\text{Loss}_{\perp}$ consists of two components. The primary purpose of having the first component is to drive the inner products between the weight vectors of neurons in the FC layer towards zero, which indicates that the linear projection features from the encoder layer to the FC

layer are mutually orthogonal. This component can also be viewed as a regularization term, which drives the weights of the FC layer to smaller values during model training and thus effectively mitigates overfitting issues. Meanwhile, the second term ensures that the extracted latent features remain as mutually independent as possible even after undergoing nonlinear transformation (σ).

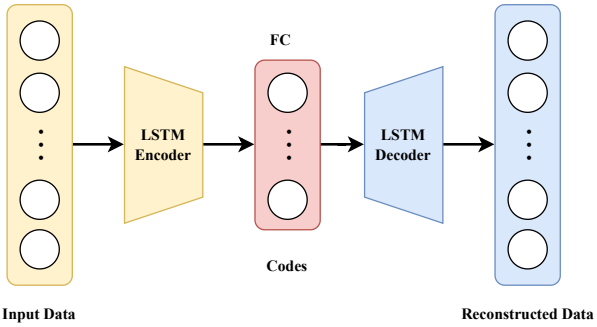

Figure 3. Our proposed OLAE architecture consists of an LSTM encoder, an LSTM decoder, and a fully connected (FC) layer that leverages orthogonality. The FC layer is denoted as $C = \sigma(Wh + b)$, where h and C represent the output of encoder and fully-connected (FC) layer, respectively. Here, b and σ are the bias term and nonlinear activation function of the FC layer, respectively.

The overall loss function of OLAE is defined as follows:

$$\text{Loss} = \text{Loss}_{\text{MSE}} + \text{Loss}_{\perp}. \tag{3}$$

By minimizing Loss, we ensure that the latent features are non-redundant and as mutually independent as possible.

2.3. Monitoring Statistics

As previously discussed, AE-based process monitoring methods typically use the reconstruction error as the monitoring statistic for fault detection. This approach does not make full use of the lower-dimensional embeddings, which could represent the underlying process dynamics. Thus, in this work, we propose to directly monitor the extracted features using the T^2 statistic defined as follows:

$$T^2 = c\Lambda_c^{-1}c^T, \tag{4}$$

where c represents the extracted codes, and Λ_c is the covariance matrix of the codes. Based on the T^2 statistic during in-control operations, the control limit for the monitoring statistic can be determined using the kernel density estimation (KDE) method [32].

While many process faults can be recognized based on changes in the numerical values of process variables being monitored, some faults are more represented by changes or variations in the distributions of process variables. To better detect the latter faults, in addition to employing the T^2 statistic, we adopt the quantile-based multivariate CUSUM method recently developed by Ye and Liu [26]. The basic idea behind this new CUSUM method is to detect process anomalies by monitoring any mean shifts in data stream distributions. Compared to traditional multivariate CUSUM techniques, this novel framework handles nonparametric and heterogeneous data streams for the first time. Previously, Jiang successfully applied this method to chemical process monitoring and achieved promising results [27]. Here, we build upon this CUSUM framework to monitor any subtle deviations in the distribution of dynamic orthogonal latent features.

For each process variable $x = 1, \ldots, p$, the data collected under normal operating conditions can be divided into d quantiles: $I_{x,1} = (-\infty, q_{x,1}]$, $I_{x,2} = (q_{x,1}, q_{x,2}]$, $\ldots$, $I_{x,d} = (q_{x,d-1}, +\infty)$, such that each quantile contains exactly $\frac{1}{d}$ of the in-control data samples. Next, we define a vector $Y_x(t) = (Y_{x,1}(t), Y_{x,2}(t), \ldots, Y_{x,d}(t))^T$ for each data stream x at time t, where $Y_{x,l} = \mathbb{I}\{G_x(t) \in I_{x,l}\}$ with $l = 1, \ldots, d$. Here, $\mathbb{I}\{G_x(t) \in I_{x,l}\}$ is an

indicator function that equals 1 when the online measurement $G_x(t)$ lies in the interval $I_{x,l}$ and 0 otherwise. Then, we define two vectors $A_x^+(t) = [A_{x,1}^+(t), \ldots, A_{x,d-1}^+(t)]^T$ and $A_x^-(t) = [A_{x,1}^-(t), \ldots, A_{x,d-1}^-(t)]^T$, where $A_{x,l}^+(t) = 1 - \sum_{i=1}^{l} Y_{x,i}(t)$, $A_{x,l}^-(t) = \sum_{i=1}^{l} Y_{x,i}(t)$. Ye and Liu [26] showed that detecting the mean shifts in the distribution of $G_x(t)$ is equivalent to detecting shifts in the distribution of $A_{x,l}^+(t)$ and $A_{x,l}^-(t)$ with respect to their expected values, which are $1 - \frac{l}{d}$ and $\frac{l}{d}$, respectively. With this, the multivariate CUSUM procedure originally proposed by Qiu and Hawkins [33] can now be employed to detect the mean shifts of $A_{x,l}^{\pm}(t)$. This is done by defining variable $C_x^{\pm}(t)$ as follows:

$$\begin{aligned} C_x^{\pm}(t) = &\left[A_x^{\pm}(t) - \mathbb{E}(A_x^{\pm}(t)) + S_x^{\pm 0}(t-1) - S_x^{\pm 1}(t-1)\right]^T \\ &\cdot [\mathrm{diag}(\mathbb{E}(A_x^{\pm}(t)) + S_x^{\pm 1}(t-1))]^{-1} \\ &\cdot \left[S_x^{\pm 0}(t-1) + S_x^{\pm 1}(t-1) - \mathbb{E}(A_x^{\pm}(t)) + A_x^{\pm}(t)\right], \end{aligned} \tag{5}$$

where $S_x^{\pm 0}(t)$ and $S_x^{\pm 1}(t)$ are $(d-1)$-dimensional vectors defined as follows:

$$\begin{cases} S_x^{\pm 0}(t) = 0,\ S_x^{\pm 1}(t) = 0 & \text{if } C_x^{\pm}(t) \le k; \\ S_x^{\pm 0}(t) = \frac{(S_x^{\pm 0}(t-1) + A_x^{\pm}(t))(C_x^{\pm}(t) - k)}{C_x^{\pm}(t)}; & \\ S_x^{\pm 1}(t) = \frac{(S_x^{\pm 1}(t-1) + \mathbb{E}(A_x^{\pm}(t)))(C_x^{\pm}(t) - k)}{C_x^{\pm}(t)} & \text{if } C_x^{\pm}(t) > k. \end{cases} \tag{6}$$

In Equation (6), k is a pre-computed allowance parameter that restarts the CUSUM procedure by resetting the local statistic back to 0 if there is no evidence of upward or downward mean shift after a while [26]. The local statistic $W_x^{\pm}(t)$ for detecting any mean shift in the upward $(+)$ or downward $(-)$ direction is calculated for each time t as follows:

$$W_x^{\pm}(t) = \max(0, C_x^{\pm}(t) - k). \tag{7}$$

Overall, we monitor the two-sided statistic $W_x(t) = \max(W_x^-(t), W_x^+(t))$ for both upward and downward mean shifts. An alarm is raised (i.e., an anomaly is detected) when the monitoring statistic, $\sum_{(x)=1}^{r} W_{(x)}(t)$, defined as the sum of the largest r local statistics W_x at each time t, exceeds a threshold h that is related to the pre-specified false alarm rate (e.g., 0.0027 for the typical 3σ-limit) [34]. The corresponding stopping time T is as follows:

$$T = \inf\left\{ t > 0 : \sum_{(x)=1}^{r} W_{(x)}(t) \ge h \right\}. \tag{8}$$

More information about the theory and application of quantile-based multivariate CUSUM, including detailed derivations of the mathematical formulations above, can be found in Ye and Liu [26].

2.4. Adaptive Weight-Based Bayesian Fusion Strategy

Here, we describe our multi-block monitoring framework for large-scale, complex industrial systems. Each block is monitored by two metrics (T^2 and W, respectively) using two approaches (Hotelling's T^2 and quantile-based CUSUM, respectively). We adopt a Bayesian data fusion framework to aggregate the two monitoring results into a single monitoring metric in each block, as well as to aggregate all block-level metrics into a single plant-wide fault index (PFI). Bayesian fusion has shown remarkable robustness and capabilities in integrating information from diverse sources. It leverages prior knowledge and real-time measurements to compute posterior probabilities using Bayes' theorem. In this work, we extend the classic Bayesian fusion methodology and propose an adaptive weight-based Bayesian fusion (W-BF) method. The idea is to dynamically adjust fusion weights based on the relative ranking of the current monitoring statistics from each block.

Under this framework, the probability of sample $X_i^n(t)$ of block n and monitoring metric i in normal and fault conditions are given by the following:

$$\mathbb{P}_i^n(X_i^n(t)|N) = \exp\left(-\frac{S_i^n(t)}{S_{i,\lim}^n}\right);\ \mathbb{P}_i^n(X_i^n(t)|F) = \exp\left(-\frac{S_{i,\lim}^n}{S_i^n(t)}\right), \tag{9}$$

where $S_i^n(t)$ and $S_{I,\lim}^n$ denote the current value and control limit of the monitoring metric i, respectively. With this, the posterior probability can be calculated using Bayes' rule as follows:

$$\begin{aligned} &\mathbb{P}_i^n(F|X_i^n(t)) = \frac{\mathbb{P}_i^n(X_i^n(t)|F)\mathbb{P}_i^n(F)}{\mathbb{P}_i^n(X_i^n(t)|F)\mathbb{P}_i^n(F) + \mathbb{P}_i^n(X_i^n(t)|N)\mathbb{P}_i^n(N)};\\ &\mathbb{P}_i^n(N) = 1-\alpha;\\ &\mathbb{P}_i^n(F) = \alpha, \end{aligned} \tag{10}$$

where $\mathbb{P}_i^n(N)$ and $\mathbb{P}_i^n(F)$ denote prior probabilities under normal and abnormal conditions, respectively; α is the significance level, which is taken to be 0.01 [35,36].

Thus, the fused monitoring statistic of block n, $B^n(t)$ is determined as follows:

$$B^n(t) = \frac{\sum_{i=1}^2 w_i^n(t)\mathbb{P}_i^n(X_i^n(t)|F)\mathbb{P}_i^n(F|X_i^n(t))}{\sum_{i=1}^2 w_i^n(t)\mathbb{P}_i^n(X_i^n(t)|F)}, \tag{11}$$

where $w_i^n(t)$ represents the weight for monitoring metric i, which is dynamically updated at every time t as follows:

$$w_i^n(t) = \frac{\exp((S_i^n(t) - S_{i,\lim}^n)/S_{i,\lim}^n)}{\sum_1^2 \exp((S_i^n(t) - S_{i,\lim}^n)/S_{i,\lim}^n)}. \tag{12}$$

Following a similar procedure, we adopt adaptive W-BF once again to aggregate $B^n(t)$ of all blocks to obtain the PFI. First, we derive the following block-wise probabilities:

$$\begin{aligned} &\mathbb{P}^n(B^n(t)|N) = \exp\left(-\frac{B^n(t)}{B_{\lim}^n}\right);\ \mathbb{P}^n(B^n(t)|F) = \exp\left(-\frac{B_{\lim}^n}{B^n(t)}\right);\\ &\mathbb{P}^n(F|B^n(t)) = \frac{\mathbb{P}^n(B^n(t)|F)\mathbb{P}^n(F)}{\mathbb{P}^n(B^n(t)|F)\mathbb{P}^n(F) + \mathbb{P}^n(B^n(t)|N)\mathbb{P}^n(N)};\\ &\mathbb{P}^n(N) = 1-\alpha;\\ &\mathbb{P}^n(F) = \alpha, \end{aligned} \tag{13}$$

where $\mathbb{P}^n(B^n(t)|N)$ and $\mathbb{P}^n(B^n(t)|F)$ represent the probability of block n in normal and fault conditions at time t, respectively; $\mathbb{P}^n(F|B^n(t))$, $\mathbb{P}^n(N)$, and $\mathbb{P}^n(F)$ are the posterior probability, the prior integration probability, and prior failure probability for block n, respectively. Finally, the PFI is calculated as follows:

$$\begin{aligned} \mathrm{PFI}(t) &= \frac{\sum_{n=1}^N w_n(t)\mathbb{P}^n(B^n(t)|F)\mathbb{P}^n(F|B^n(t))}{\sum_{n=1}^N w_n(t)\mathbb{P}^n(B^n(t)|F)};\\ w_n(t) &= \frac{\exp((B^n(t) - B_{\lim}^n)/B_{\lim}^n)}{\sum_{n=1}^N \exp((B^n(t) - B_{\lim}^n)/B_{\lim}^n)}, \end{aligned} \tag{14}$$

where N is the number of blocks, and $B_{\lim}^n$ is the control limit of the fused monitoring statistic of block n.

3. The MOLA Fault Detection Framework

Now that all components of our proposed process monitoring framework have been introduced, we will move on to discuss how these components are integrated with our

MOLA framework during offline learning and online monitoring stages. The overall flowchart of MOLA is shown in Figure 4.

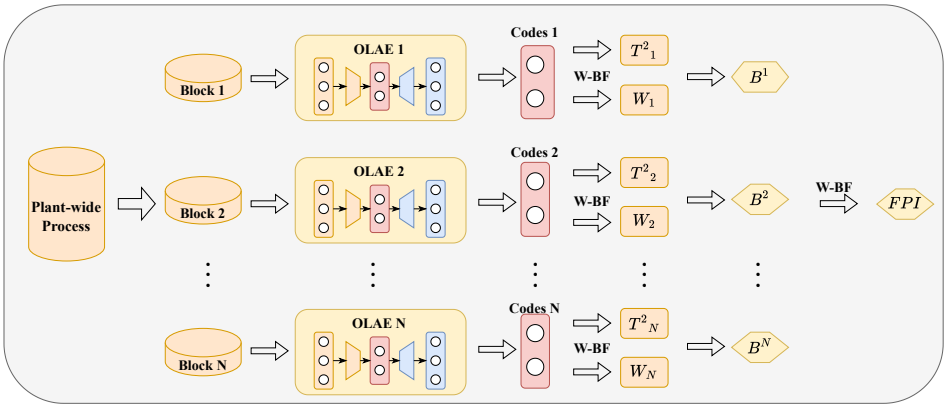

Figure 4. The MOLA process monitoring framework features an OLAE model for each block and adaptive W-BF for data fusion.

The steps involved in the offline learning stage are outlined below:

Step 1: Historical in-control data are collected, normalized, and standardized.

Step 2: Based on process knowledge, process variables are divided into blocks. For each block, we establish a local OLAE model. We use 70% of the historical in-control data to build the OLAE model, whereas the remaining data serve as the validation set to determine the optimal hyperparameters for the model.

Step 3: The validation data are sent to the OLAE model of each block to obtain the corresponding latent features or codes. Then, we calculate T^2 and W for each block and determine their control limits.

Step 4: For each block, we implement the adaptive W-BF strategy to obtain a fused monitoring statistic.

Step 5: Based on the fused monitoring statistics of each block, we apply the adaptive W-BF technique again to determine the PFI for the overall process.

The steps involved in the online monitoring stage are as follows:

Step 1: As online data are collected, they are standardized based on the mean and variance of the in-control data collected during the offline learning phase.

Step 2: Following the block assignments, standardized online data are sent to their corresponding block's OLAE model to obtain the codes and monitoring statistics T^2 and W.

Step 3: The fused monitoring statistic is calculated following the adaptive W-BF strategy for each block.

Step 4: Using the adaptive W-BF method again, we obtain the process-level PFI from the local statistics of all blocks. If the PFI is greater than our pre-defined significance level α (0.01), a fault is declared; otherwise, the process is under normal operation, and the monitoring continues.

4. Case Study

The Tennessee Eastman process (TEP) is a simulation process developed by the Eastman Chemical Company based on an actual chemical process [37]. The TEP has been widely adopted as a benchmark for chemical process control, optimization, and monitoring. As illustrated in the process flow diagram of Figure 5, the TEP contains five major unit operations, which are associated with 12 manipulated variables and 41 measured variables in total. Among them, 31 variables (listed in Table 2) are typically selected to conduct process monitoring, as the remaining variables have relatively low sampling frequencies. In multi-block process monitoring methods, the choice of block division and assignment approach directly influences the process monitoring performance. Existing block division

methods include process knowledge-based methods, variable relation-based methods, and fault information-aided methods [38]. In this work, we adopt the process knowledge-based approach developed by Zhu et al. [39] and divide the entire process into four distinct blocks ($N = 4$), as outlined in Table 3. The basic idea is to assign process variables associated with the same equipment into a single block. Due to the relatively small number of process variables being measured for the condenser and compressor, we assign these variables to the nearest separator block.

In this work, the simulation data are generated from a Simulink implementation with a sampling frequency of one minute [40]. Initially, a dataset comprising 60,000 samples under normal operating conditions is generated and used for offline learning. Subsequently, an additional 500 datasets, each containing 2,400 samples, are generated to determine the control limits in process monitoring models. Furthermore, this Simulink implementation can simulate 20 types of process faults (see Table 4) as test datasets for process online monitoring. It is worth noting that any process fault is introduced at the 600th sample in each test dataset.

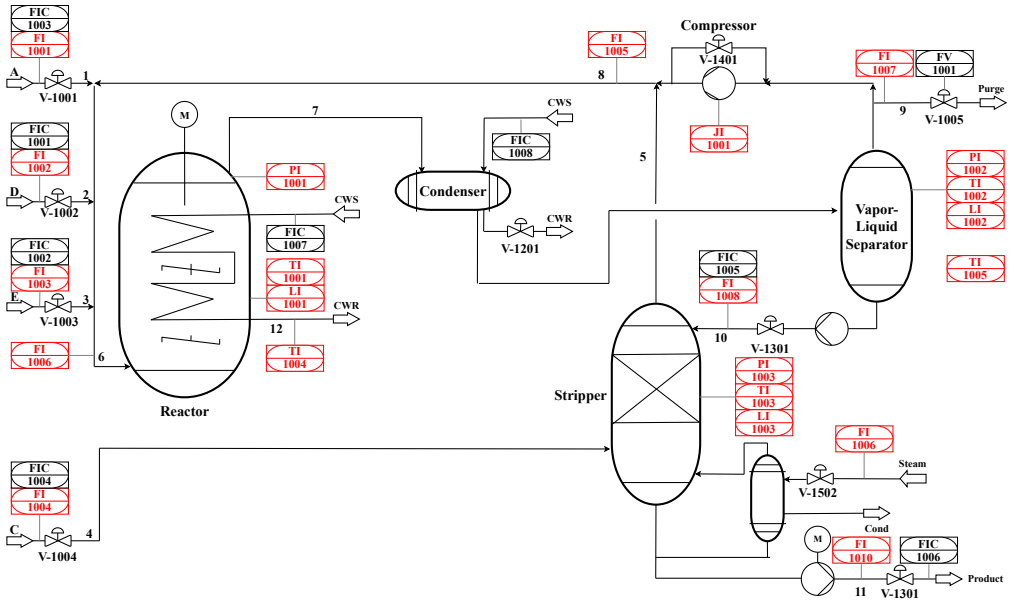

Figure 5. Process flow diagram of the TEP.

Table 2. Detailed description of the monitoring variables.

No.	Variable	Description	No.	Variable	Description
0	FI-1001	A feed (stream 1)	**16**	FI-1009	Stripper underflow (stream 11)
1	FI-1002	D feed (stream 2)	**17**	TI-1003	Stripper temperature
2	FI-1003	E feed (stream 3)	**18**	FI-1010	Stripper steam flow
3	FI-1004	A and C feed (stream 4)	**19**	JI-1001	Compressor work
4	FI-1005	Recycle flow (stream 8)	**20**	TI-1004	Reactor cooling water outlet temperature
5	FI-1006	Reactor feed rate (stream 6)	**21**	TI-1005	Separator cooling water outlet temperature
6	PI-1001	Reactor pressure	**22**	FIC-1001	D feed flow (stream 2)
7	LI-1001	Reactor level	**23**	FIC-1002	E feed flow (stream 3)
8	TI-1001	Reactor temperature	**24**	FIC-1003	A feed flow (stream 1)
9	FI-1007	Purge rate (stream 9)	**25**	FIC-1004	A and C feed flow (stream 4)
10	TI-1002	Product separator temperature	**26**	FV-1001	Purge valve (stream 9)
11	LI-1002	Product separator level	**27**	FIC-1005	Separator pot liquid flow (stream 10)
12	PI-1002	Product separator pressure	**28**	FIC-1006	Stripper liquid prod flow (stream 11)
13	FI-1008	Product separator underflow (stream 10)	**29**	FIC-1007	Reactor cooling water flow
14	LI-1003	Stripper level	**30**	FIC-1008	Condenser cooling water flow
15	PI-1003	Stripper pressure			

Table 3. Divided blocks of the monitoring variables.

Block	Variables	Description
1	0, 1, 2, 4, 5, 22, 23, 24	Input
2	6, 7, 8, 20, 29	Reactor
3	9, 10, 11, 12, 13, 19, 21, 26, 27, 30	Separator, compressor and condenser
4	3, 14, 15, 16, 17, 18, 25, 28	Stripper

Table 4. Detailed description of faults in TEP problem [39].

Fault No.	Process Variable	Type
1	A/C feed ratio, B composition constant (stream 4)	Step
2	B composition, A/C ratio constant (stream 4)	Step
3	D feed temperature (stream 2)	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss (stream 1)	Step
7	C header pressure loss-reduced availability (stream 4)	Step
8	A, B, C feed composition (stream 4)	Random variation
9	D feed temperature (stream 2)	Random variation
10	C feed temperature (stream 4)	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16	Unknown	Unknown
17	Unknown	Unknown
18	Unknown	Unknown
19	Unknown	Unknown
20	Unknown	Unknown

We compare our proposed framework with other process monitoring models, including PCA, AE, LSTM AE, block PCA, and block LSTM AE. The complete structure of each neural network implemented in our framework is listed in Table 5. In this work, all neural networks use the rectified linear unit (ReLU) activation function. The number of iterations is determined using early stopping criteria. Specifically, we stop the training process when the loss function value on the validation dataset does not decrease for 30 consecutive iterations. The detailed structure of different neural networks is illustrated in Table 5. Tables 6 and 7 show the performance of these models in terms of fault detection delay (FDD) and fault detection rate (FDR), respectively. It can be seen that, in general, the incorporation of a multi-block monitoring strategy not only reduces the FDD but also increases the FDR of original process monitoring methods. This validates the effectiveness of our proposed multi-block method and data fusion technique. In particular, our MOLA framework exhibits superior performance over all other frameworks in terms of FDR and FDD. For the vast majority of fault cases, particularly faults 13, 15, 16, 24, and 18, MOLA can detect faults tens to hundreds of minutes ahead of other methods, providing invaluable time buffers for engineers and operators to take appropriate control actions. For a few faults, such as faults 3, 11, 14, and 20, MOLA falls behind the best-performing method by just a few minutes. However, considering that the differences are small and that MOLA achieves significant improvements in FDR in these faults, MOLA still outperforms other methods by a considerable margin. For example, for faults 3, 5, 9, 15, and 16, while other methods perform poorly on these faults, MOLA increases the FRD by 4.6 to 30 times. Note that faults 3, 9, and 15 are known to be notoriously difficult to detect due to their intricate process dynamics. Meanwhile, the fault detection rates for MOLA on these three faults

range between 88.3 and 98.5%, demonstrating MOLA's exceptional capabilities in tackling challenging fault detection scenarios.

Table 5. Detailed structure for different neural networks.

Model	Structure	Activation Function
AE	FC(64)-FC(16)-FC(64)	ReLU
LSTM AE	LSTM(64)-FC(16)-LSTM(64)	ReLU
Block LSTM AE	LSTM(15)-FC(5)-LSTM(15)	ReLU
OLAE	LSTM(15)-FC(5)-LSTM(15)	ReLU

We now take a close look at fault 3, whose process monitoring results for various methods are summarized in Figure 6. As we can see, while the LSTM AE-based approach successfully detects the fault at the 627th sample (i.e., 27 samples after the process fault is introduced into the simulation), its monitoring statistic only briefly exceeds the control limit and fails to raise a continuous alarm. Similarly, most other methods are unable to raise any valid alarm for this fault, as in practice alarms, would only be considered effective when the monitoring statistic exceeds the control limit multiple consecutive times (e.g., three) [41]. If a process monitoring model fails to continuously report faults, plant operators may erroneously assume that the alarm is false or the process has returned back to normal. On the other hand, our MOLA framework captures the fault starting at the 630th sample and subsequently raises alarms continuously, making it a useful process monitoring framework in practice.

Similarly, for fault 9 (see Figure 7), the MOLA framework successfully raises the alarm at the 625th sample and sustains the alarm, whereas all other methods remain ineffective in raising any meaningful alarm. Although MOLA occasionally produces false alarms during fault-free periods (see Table 8 for the false alarm rates), these false alarms are isolated incidents of brief statistical excursions at a single sample, which have minimal impact in practice. These false alarms are typically attributed to suboptimal tuning of hyperparameters and can be resolved when more comprehensive model training is conducted.

Table 6. Comparison of FDD results in TEP.

Fault No.	PCA (T^2)	PCA (Q)	AE	LSTM AE	Block PCA (T^2)	Block PCA (Q)	Block LSTM AE	MOLA
1	3	18	3	1	1	15	4	**1**
2	25	140	22	20	19	726	18	**16**
3	–	–	–	**27**	–	–	572	30
4	0	0	0	0	0	0	0	**0**
5	–	–	–	208	2	–	3	**2**
6	0	0	0	0	0	0	0	**0**
7	0	0	0	0	0	0	0	**0**
8	29	200	26	28	25	45	27	**25**
9	229	–	–	40	26	–	233	**25**
10	91	90	77	77	74	188	74	**70**
11	36	41	36	**22**	36	37	36	36
12	66	–	65	63	66	475	**48**	60
13	94	317	86	86	89	307	91	**9**
14	–	7	3	**0**	2	3	2	2
15	191	–	–	–	–	–	191	**6**
16	–	–	–	–	–	–	–	**21**
17	57	62	55	55	56	58	57	**24**
18	267	280	242	241	259	259	268	**147**
19	30	32	15	16	21	32	12	**11**
20	141	178	129	128	138	164	**123**	124

Note: "0" means that the fault is detected at the time of introduction, while "–" means that the fault is not effectively detected.

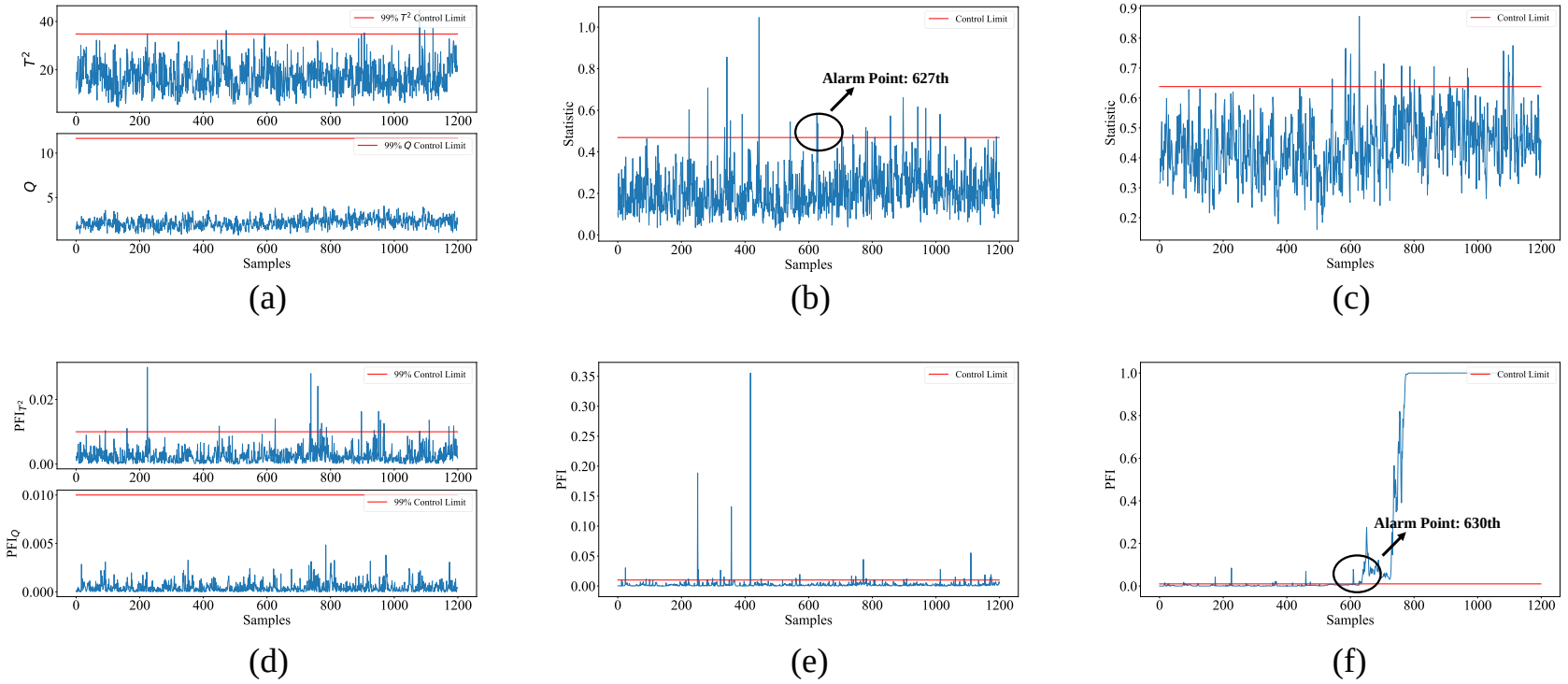

Figure 6. The process monitoring results of Fault 3 based on (**a**) PCA, (**b**) AE, (**c**) LSTM AE, (**d**) block PCA, (**e**) block LSTM AE, and (**f**) MOLA.

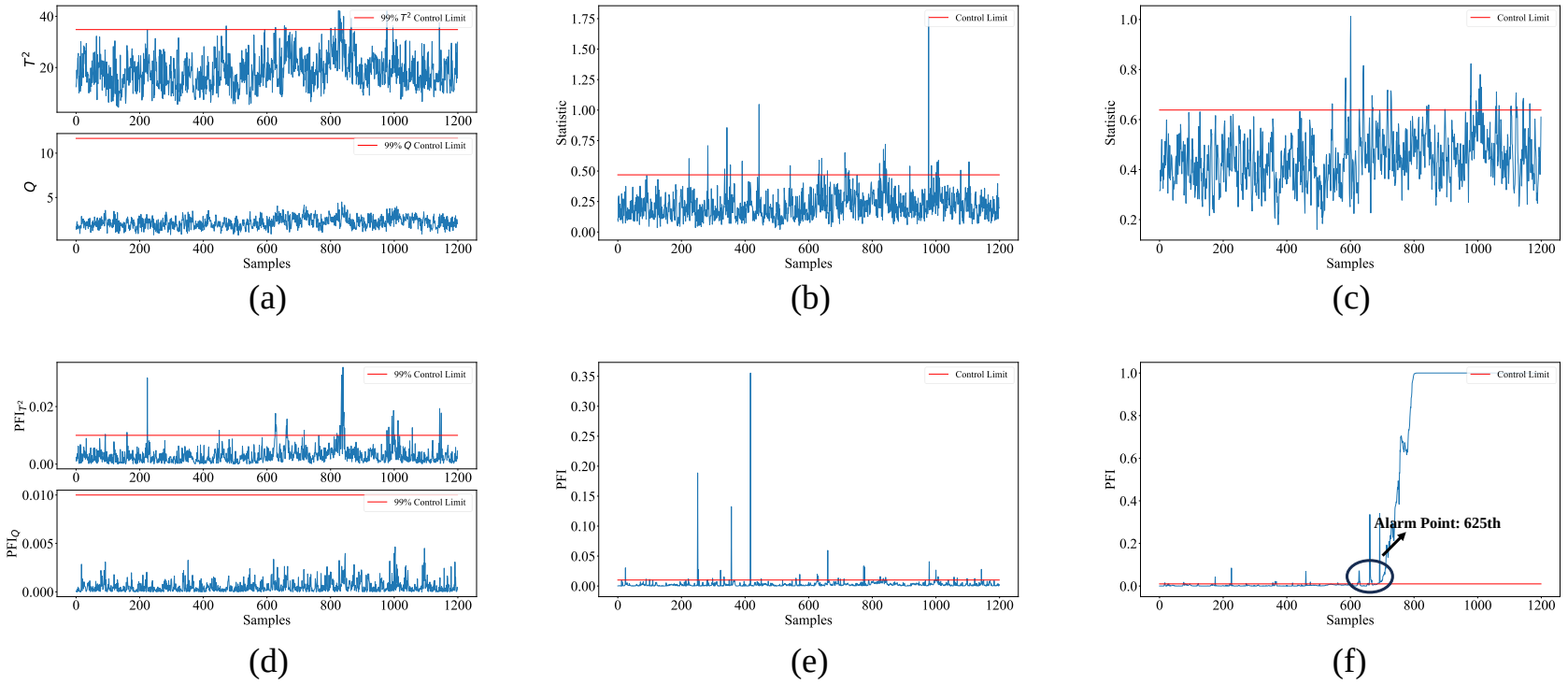

Figure 7. The process monitoring results of Fault 9 based on (**a**) PCA, (**b**) AE, (**c**) LSTM AE, (**d**) block PCA, (**e**) block LSTM AE, and (**f**) MOLA.

Table 7. Comparison of FDR results in TEP. Here, Q is the square prediction error, and % improvement measures the % difference in FDR between MOLA and the best-performing method.

No.	PCA (T^2)	PCA (Q)	AE	LSTM AE	Block PCA (T^2)	Block PCA (Q)	Block LSTM AE	MOLA	% Improvement
1	0.9967	0.9700	0.9950	0.9967	0.9983	0.9750	0.9950	**0.9983**	0.00
2	0.9617	0.6850	0.9667	0.9650	0.9683	0.7200	0.9700	**0.9767**	0.68
3	0.0067	0.0000	0.0183	0.0467	0.0267	0.0000	0.0350	**0.9567**	1950.00
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	**1.0000**	0.00
5	0.0183	0.0000	0.0033	0.0300	0.0350	0.0000	0.0467	**0.2633**	464.28
6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	**1.0000**	0.00
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	**1.0000**	0.00
8	0.8500	0.6683	0.8883	0.8500	0.8883	0.7000	0.9000	**0.9433**	4.81
9	0.0300	0.0000	0.0567	0.0633	0.0517	0.0000	0.0400	**0.8850**	1297.37
10	0.6700	0.5367	0.8100	0.8033	0.8233	0.0400	0.8350	**0.8833**	5.79
11	0.9400	0.8417	0.9433	0.9500	0.9483	0.8600	0.9433	**0.9533**	0.35
12	0.3467	0.0000	0.3083	0.3267	0.4383	0.0100	0.4550	**0.6467**	42.12
13	0.8467	0.4167	0.8483	0.8417	0.8583	0.4133	0.8583	**0.8733**	1.75
14	0.9883	0.6567	0.9950	0.9967	0.9967	0.7833	0.9767	**0.9967**	0.00
15	0.0117	0.0000	0.0150	0.0100	0.0133	0.0000	0.0317	**0.9850**	3010.52
16	0.0167	0.0000	0.0117	0.0083	0.0083	0.0000	0.0250	**0.6450**	2480.00
17	0.9017	0.7683	0.9083	0.9100	0.9067	0.8283	0.9050	**0.9233**	2.03
18	0.3500	0.2567	0.4850	0.4817	0.4200	0.3867	0.3950	**0.7633**	57.39
19	0.9317	0.8000	0.9783	0.9683	0.9683	0.6550	0.9750	**0.9817**	0.68
20	0.7650	0.5400	0.7933	0.7900	0.7750	0.5583	0.7917	**0.7933**	0.00

Table 8. Comparison of false-alarm rates in TEP.

	PCA (T^2)	PCA (Q)	AE	LSTM AE	Block PCA (T^2)	Block PCA (Q)	Block LSTM AE	MOLA
FDR	0.0017	0.0000	0.0133	0.0067	0.0067	0.0000	0.0283	0.0350

Ablation Studies Evaluating the Contribution of MOLA Components to Its Process Monitoring Performance

Our MOLA framework consists of several innovative components. To examine how these components contribute to the overall success of MOLA, we present results from ablation studies designed to individually evaluate the effectiveness of each component or improvement, offering quantitative understanding and deep insights into MOLA.

First, we validate the effectiveness of MOLA in extracting non-redundant features. The maximal information coefficient (MIC) [42], ranging from 0 to 1, serves as a quantitative measure of the correlation between two variables. An MIC value of 0 between two variables indicates their mutual independence, whereas as MIC value that approaches 1 suggests the presence of a strong correlation between the two variables. Here, we use MIC as an indicator to assess the correlation among the extracted features. Specifically, we calculate the MIC values between any two features extracted by both the LSTM AE and the MOLA and summarize the results in Table 9. Clearly, the MIC values for features extracted by MOLA are close to 0, whereas the MIC values for features extracted by LSTM AE are significantly larger. This indicates that MOLA demonstrates a significant advantage in extracting non-redundant features. As illustrated in Table 10, this leads to faster fault detection speed at the block level for MOLA compared to LSTM AE, when both methods use T^2 as the monitoring statistic. Meanwhile, MOLA without W-BF and CUSUM can detect process faults earlier than LSTM AE. This result indicates that introducing orthogonality constraints can facilitate faster detection of process faults. This is primarily because the redundancies among features extracted by traditional autoencoders may obscure the underlying data

structure, making faults harder to detect. Instead, our proposed OLAE only extracts orthogonal features, making fault information more prominent and thus easier to detect.

Table 9. Orthogonality and mutual independence of extracted latent variables.

No.	Block 1		Block 2		Block 3		Block 4	
	LSTM AE	MOLA	LSTM AE	MOLA	LSTM AE	MOLA	LSTM AE	MOLA
MIC(F1,F2)	0.2637	0.0337	0.0555	0.0541	0.1278	0.0706	0.9752	0.0405
MIC(F1,F3)	0.7957	0.1558	0.0708	0.0754	0.0427	0.0590	0.6531	0.0586
MIC(F1,F4)	0.3043	0.0628	0.1924	0.0908	0.0901	0.0867	0.7901	0.0547
MIC(F1,F5)	0.7275	0.0256	0.0774	0.1182	0.1287	0.0978	0.9610	0.0438
MIC(F2,F3)	0.7238	0.0723	0.4757	0.0879	0.0712	0.0938	0.9175	0.0789
MIC(F2,F4)	0.6029	0.1292	0.2940	0.1075	0.0663	0.0916	0.8663	0.0664
MIC(F2,F5)	0.5836	0.0946	0.5009	0.0918	0.1219	0.0559	0.9773	0.0550
MIC(F3,F4)	0.5106	0.1165	0.3697	0.0595	0.0352	0.0347	0.8658	0.0480
MIC(F3,F5)	0.9525	0.0948	0.9523	0.0742	0.0523	0.0387	0.8695	0.0417
MIC(F4,F5)	0.4059	0.0397	0.3442	0.0442	0.0944	0.1517	0.5427	0.0598

Note: F1, F2, F3, F4, and F5 represent the features extracted from the latent space.

Table 10. Comparison of fault detection speed using LSTM AE and MOLA.

No.	Block 1		Block 2		Block 3		Block 4		Block	MOLA w/o
	LSTM AE	OLAE	LSTM AE	OLAE	LSTM AE	OLAE	LSTM AE	OLAE	LSTM AE	BF and CUSUM
1	21	**18**	4	**1**	4	**3**	3	**2**	4	**2**
2	**117**	118	21	**17**	22	**16**	18	**9**	18	**16**
3	-	-	-	**139**	573	**573**	-	-	572	**160**
4	-	-	0	**0**	10	**10**	-	**44**	0	**0**
5	-	-	-	**1**	3	**2**	-	**3**	3	**2**
6	0	**0**	1	**0**	3	**3**	2	**2**	0	**0**
7	-	-	0	**0**	1	**1**	0	**0**	0	**0**
8	56	**36**	182	**28**	32	**32**	27	**24**	27	**27**
9	-	**23**	233	**61**	-	**61**	-	**62**	233	**61**
10	-	-	-	**380**	170	**170**	74	**70**	74	**70**
11	-	-	36	**36**	47	**41**	63	**49**	36	**36**
12	-	**553**	85	**60**	**48**	60	65	**65**	**48**	60
13	249	**245**	97	**74**	89	**84**	106	**9**	91	**84**
14	-	**202**	2	**2**	-	**56**	-	**26**	2	**2**
15	-	-	-	**82**	192	**191**	-	**209**	**191**	192
16	-	-	-	-	54	**18**	-	**19**	-	**25**
17	-	-	56	**54**	118	**117**	140	**57**	57	**56**
18	-	-	345	**345**	263	**260**	346	**346**	268	**260**
19	-	-	409	**244**	25	**20**	12	**11**	12	**12**
20	164	**162**	212	**177**	129	**129**	**123**	124	**123**	127

Note: "0" means that the fault is detected at the time of its introduction, and "–" means that the fault is not effectively detected during the entire monitoring period.

Next, we investigate the benefits of introducing the quantile-based CUSUM method and adaptive W-BF strategy on process monitoring performance. By designing ablation experiments, we quantify the enhancements and present the results in Table 11. Specifically, "Full MOLA" represents the complete MOLA framework presented earlier, "MOLA no CUSUM" refers to the MOLA framework without implementing the CUSUM procedure, and "MOLA no BF" is the MOLA framework without implementing an adaptive W-BF strategy. Table 11 shows that introducing adaptive W-BF significantly improves the fault detection speed for faults 13 and 15 and enhances the fault detection rate for all faults other than 1, 4, 6, 7, 10, and 14. For instance, for fault 13, the weights assigned to blocks 1 through

4 by the adaptive W-BF strategy at the 609th time step (i.e., nine samples after the fault is introduced) are 0.2396, 0.1372, 0.1542, and 0.4689, respectively. Block 4, which has the highest weight to PFI among all blocks, also turns out to be the first block where an alarm is raised. Thus, the adaptive W-BF method automatically prompts the FPI to give greater attention to blocks that detect anomalies earlier, which enhances both fault detection rate and speed.

We remark that the W-BF method is particularly attractive in monitoring modern, large-scale industrial processes, whose scale and complexity require process monitoring to be performed in a distributed manner. Since process faults typically occur locally, by decomposing the overall process into multiple blocks, our W-BF method ensures that, as a fault occurs, process monitoring performance is sensitive to the faulty blocks and will not be negatively impacted by the surveillance of non-faulty blocks.

Table 11. Contribution of quantile-based CUSUM and adaptive W-BF to improvements of process monitoring performance.

Fault No.	Full MOLA		MOLA no BF		MOLA no CUSUM	
	FDD	FDR	FDD	FDR	FDD	FDR
1	**1**	**0.9983**	1	0.9983	1	0.9983
2	**16**	**0.9767**	16	0.9750	16	0.9750
3	**30**	**0.9567**	31	0.9550	139	0.1117
4	**0**	**1.0000**	0	1.0000	0	1.0000
5	**2**	**0.2633**	2	0.2283	2	0.0717
6	**0**	**1.0000**	0	1.0000	0	1.0000
7	**0**	**1.0000**	0	1.0000	0	1.0000
8	**25**	**0.9433**	25	0.9350	26	0.9167
9	25	**0.8850**	25	0.8800	**23**	0.1550
10	**70**	**0.8833**	70	0.8833	70	0.8567
11	**36**	**0.9533**	36	0.9500	36	0.9500
12	**60**	**0.6467**	60	0.6217	60	0.5517
13	**09**	**0.8733**	84	0.8667	84	0.8717
14	**2**	**0.9967**	2	0.9967	2	0.9967
15	**6**	**0.9850**	12	0.9550	192	0.0667
16	**21**	**0.6450**	25	0.6233	25	0.0450
17	**24**	**0.9233**	24	0.9183	54	0.9117
18	**147**	**0.7633**	148	0.7567	260	0.4567
19	**11**	**0.9817**	12	0.9800	11	0.9817
20	**124**	**0.7933**	124	0.7917	124	0.7933
Average	**30.45**	**0.8734**	30.85	0.8658	56.25	0.6855

Meanwhile, a comparison between "full MOLA" and "MOLA no CUSUM" shows that incorporating quantile-based CUSUM procedure greatly enhances the fault detection rate and speed for hard-to-detect faults of 3, 9, and 15. Again, take fault 3 as an example. As shown in Figure 8, under normal and faulty conditions, the numerical range of one of the specific features extracted from block 2 remains largely the same, making traditional distance-based monitoring statistics such as T^2 struggle to detect the fault. However, when plotting the distribution of the features under normal and faulty conditions, considerable changes are noticed. Therefore, the quantile-based multivariate CUSUM method, which directly targets the detection of distribution changes, significantly improves the accuracy and speed of fault detection under these scenarios. Furthermore, as shown in Figure 9, the quantile-based CUSUM statistic continuously accumulates as fault 3 occurs and will not go below the control limit again once an alarm is raised. This causes less confusion among plant operators in interpreting alarms and thus offers an additional advantage over other monitoring methods.

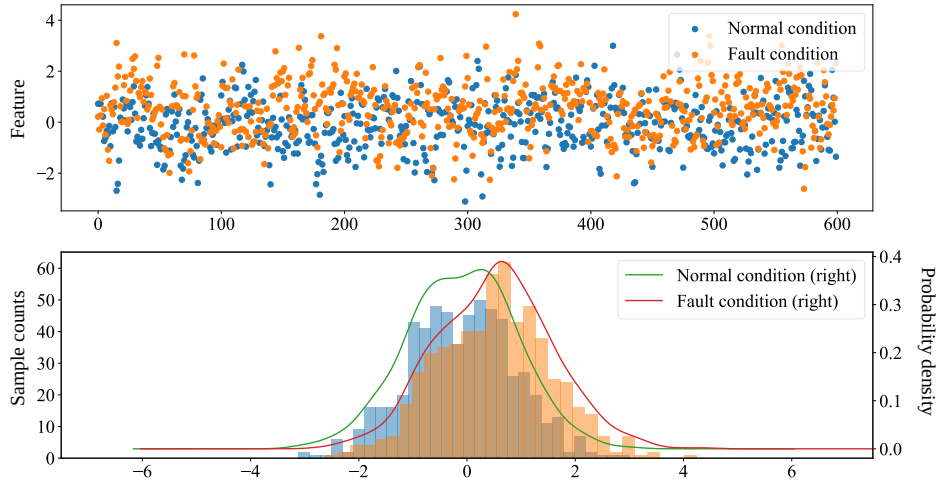

Figure 8. Feature values and their distribution.

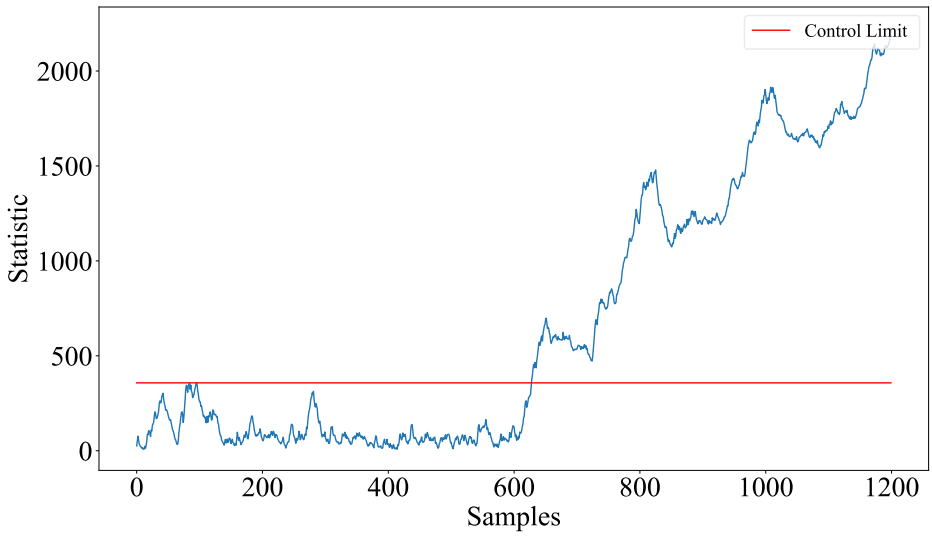

Figure 9. Variations of the quantile-based non-parametric CUSUM statistic for block 2.

5. Conclusions

In this work, we develop a novel process monitoring framework named MOLA for large-scale industrial processes. By adopting a multi-block monitoring strategy, MOLA successfully addresses the challenges posed by the scale and complexity of modern industrial processes. MOLA can extract dynamic orthogonal latent features, making sure that the most essential, non-redundant features are identified and extracted. In addition, MOLA incorporates a quantile-based multivariate CUSUM method, which enhances the ability to detect faults characterized by subtle changes in feature distributions. Furthermore, the adaptive weight-based Bayesian fusion strategy enhances fault detection rate and speed. We remark that these methods lead to synergistic improvement in process monitoring performance when they are integrated with the MOLA framework. Case study results on the TEP problem indicate that MOLA not only significantly improves fault detection rates and speeds but also successfully detects faults that are considered difficult to detect in prior research, thereby opening up many exciting opportunities for fast, accurate, and reliable industrial process monitoring applications.

Despite these achievements, we remark that there is still room for improvement in our MOLA framework. For instance, the current block assignment technique is solely based on our expert knowledge without fully considering the correlations among blocks, which may

have an impact on both local and global monitoring performance. Therefore, future research will focus on exploring more scientific and reasonable methods for dividing process sub-blocks, taking into account the correlations between sub-blocks to further optimize the process monitoring model. To be specific, it is worth investigating the correlation analysis on process variables prior to assigning them to different blocks. In this way, the dynamic features extracted not only capture the characteristics of process data within individual blocks but also incorporate the interconnections across different blocks. This will further enhance its monitoring performance and robustness in complex industrial processes.

Author Contributions: Conceptualization, F.M. and Z.J.; methodology, F.M. and Z.J.; software, F.M. and C.J.; validation, F.M. and Z.J.; formal analysis, F.M. and Z.J.; investigation, F.M. and Z.J.; resources, Z.J.; data curation, F.M.; writing—original draft preparation, F.M.; writing—review and editing, Z.J., X.T., J.W., and W.S.; visualization, F.M. and Z.J.; supervision, Z.J., X.T., J.W., and W.S.; project administration, Z.J.; funding acquisition, Z.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the U.S. National Science Foundation (NSF) under award number 2331080 and Oklahoma Center for Advancement of Science and Technology (OCAST) Oklahoma Applied Research Support (OARS) program grant number AR24-069. Financial support from the startup fund of College of Engineering, Architecture, and Technology at Oklahoma State University is also greatly acknowledged.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Amin, M.T.; Imtiaz, S.; Khan, F. Process system fault detection and diagnosis using a hybrid technique. *Chem. Eng. Sci.* **2018**, *189*, 191–211. [CrossRef]
2. Nawaz, M.; Maulud, A.S.; Zabiri, H.; Suleman, H. Review of multiscale methods for process monitoring, with an emphasis on applications in chemical process systems. *IEEE Access* **2022**, *10*, 49708–49724. [CrossRef]
3. Qin, S.J. Survey on data-driven industrial process monitoring and diagnosis. *Annu. Rev. Control.* **2012**, *36*, 220–234. [CrossRef]
4. Li, S.; Luo, J.; Hu, Y. Nonlinear process modeling via unidimensional convolutional neural networks with self-attention on global and local inter-variable structures and its application to process monitoring. *ISA Trans.* **2022**, *121*, 105–118. [CrossRef]
5. Dong, Y.; Qin, S.J. A novel dynamic PCA algorithm for dynamic data modeling and process monitoring. *J. Process. Control* **2018**, *67*, 1–11. [CrossRef]
6. Bounoua, W.; Bakdi, A. Fault detection and diagnosis of nonlinear dynamical processes through correlation dimension and fractal analysis based dynamic kernel PCA. *Chem. Eng. Sci.* **2021**, *229*, 116099. [CrossRef]
7. Pilario, K.E.; Shafiee, M.; Cao, Y.; Lao, L.; Yang, S.H. A review of kernel methods for feature extraction in nonlinear process monitoring. *Processes* **2019**, *8*, 24. [CrossRef]
8. Tan, R.; Ottewill, J.R.; Thornhill, N.F. Monitoring statistics and tuning of kernel principal component analysis with radial basis function kernels. *IEEE Access* **2020**, *8*, 198328–198342. [CrossRef]
9. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef]
10. Wu, H.; Zhao, J. Deep convolutional neural network model based chemical process fault diagnosis. *Comput. Chem. Eng.* **2018**, *115*, 185–197. [CrossRef]
11. Arunthavanathan, R.; Khan, F.; Ahmed, S.; Imtiaz, S. A deep learning model for process fault prognosis. *Process. Saf. Environ. Prot.* **2021**, *154*, 467–479. [CrossRef]

12. Heo, S.; Lee, J.H. Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine* **2018**, *51*, 470–475. [CrossRef]
13. Yang, Z.; Xu, B.; Luo, W.; Chen, F. Autoencoder-based representation learning and its application in intelligent fault diagnosis: A review. *Measurement* **2022**, *189*, 110460. [CrossRef]
14. Ji, C.; Sun, W. A review on data-driven process monitoring methods: Characterization and mining of industrial data. *Processes* **2022**, *10*, 335. [CrossRef]
15. Fan, J.; Wang, W.; Zhang, H. AutoEncoder based high-dimensional data fault detection system. In Proceedings of the 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Emden, Germany, 24–26 July 2017; pp. 1001–1006.
16. Qian, J.; Song, Z.; Yao, Y.; Zhu, Z.; Zhang, X. A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemom. Intell. Lab. Syst.* **2022**, *231*, 104711. [CrossRef]
17. Wang, W.; Yang, D.; Chen, F.; Pang, Y.; Huang, S.; Ge, Y. Clustering with orthogonal autoencoder. *IEEE Access* **2019**, *7*, 62421–62432. [CrossRef]
18. Cacciarelli, D.; Kulahci, M. A novel fault detection and diagnosis approach based on orthogonal autoencoders. *Comput. Chem. Eng.* **2022**, *163*, 107853. [CrossRef]
19. Md Nor, N.; Che Hassan, C.R.; Hussain, M.A. A review of data-driven fault detection and diagnosis methods: Applications in chemical process systems. *Rev. Chem. Eng.* **2020**, *36*, 513–553. [CrossRef]
20. Ji, C.; Ma, F.; Wang, J.; Sun, W. Orthogonal projection based statistical feature extraction for continuous process monitoring. *Comput. Chem. Eng.* **2024**, *183*, 108600. [CrossRef]
21. Nawaz, M.; Maulud, A.S.; Zabiri, H.; Taqvi, S.A.A.; Idris, A. Improved process monitoring using the CUSUM and EWMA-based multiscale PCA fault detection framework. *Chin. J. Chem. Eng.* **2021**, *29*, 253–265. [CrossRef]
22. Ge, Z. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemom. Intell. Lab. Syst.* **2017**, *171*, 16–25. [CrossRef]
23. Huang, J.; Ersoy, O.K.; Yan, X. Fault detection in dynamic plant-wide process by multi-block slow feature analysis and support vector data description. *ISA Trans.* **2019**, *85*, 119–128. [CrossRef] [PubMed]
24. Zhai, C.; Sheng, X.; Xiong, W. Multi-block Fault Detection for Plant-wide Dynamic Processes Based on Fault Sensitive Slow Features and Support Vector Data Description. *IEEE Access* **2020**, *8*, 120737–120745. [CrossRef]
25. Li, Y.; Peng, X.; Tian, Y. Plant-wide process monitoring strategy based on complex network and Bayesian inference-based multi-block principal component analysis. *IEEE Access* **2020**, *8*, 199213–199226. [CrossRef]
26. Ye, H.; Liu, K. A generic online nonparametric monitoring and sampling strategy for high-dimensional heterogeneous processes. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1503–1516. [CrossRef]
27. Jiang, Z. Online Monitoring and Robust, Reliable Fault Detection of Chemical Process Systems. In *Computer Aided Chemical Engineering*; Elsevier: Amsterdam, The Netherlands, 2023; Volume 52, pp. 1623–1628.
28. Zhang, S.; Bi, K.; Qiu, T. Bidirectional recurrent neural network-based chemical process fault diagnosis. *Ind. Eng. Chem. Res.* **2019**, *59*, 824–834. [CrossRef]
29. Deng, W.; Li, Y.; Huang, K.; Wu, D.; Yang, C.; Gui, W. LSTMED: An uneven dynamic process monitoring method based on LSTM and Autoencoder neural network. *Neural Netw.* **2023**, *158*, 30–41. [CrossRef]
30. Ren, J.; Ni, D. A batch-wise LSTM-encoder decoder network for batch process monitoring. *Chem. Eng. Res. Des.* **2020**, *164*, 102–112. [CrossRef]
31. Ma, F.; Wang, J.; Sun, W. A Data-Driven Semi-Supervised Soft-Sensor Method: Application on an Industrial Cracking Furnace. *Front. Chem. Eng.* **2022**, *4*, 899941. [CrossRef]
32. Mao, T.; Zhang, Y.; Ruan, Y.; Gao, H.; Zhou, H.; Li, D. Feature learning and process monitoring of injection molding using convolution-deconvolution auto encoders. *Comput. Chem. Eng.* **2018**, *118*, 77–90. [CrossRef]
33. Qiu, P.; Hawkins, D. A nonparametric multivariate cumulative sum procedure for detecting shifts in all directions. *J. R. Stat. Soc. Ser. D Stat.* **2003**, *52*, 151–164. [CrossRef]
34. Mei, Y. Quickest detection in censoring sensor networks. In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 31 July–5 August 2011; pp. 2148–2152.
35. Rong, M.; Shi, H.; Song, B.; Tao, Y. Multi-block dynamic weighted principal component regression strategy for dynamic plant-wide process monitoring. *Measurement* **2021**, *183*, 109705. [CrossRef]
36. Ge, Z.; Chen, J. Plant-wide industrial process monitoring: A distributed modeling framework. *IEEE Trans. Ind. Inform.* **2015**, *12*, 310–321. [CrossRef]
37. Downs, J.J.; Vogel, E.F. A plant-wide industrial process control problem. *Comput. Chem. Eng.* **1993**, *17*, 245–255. [CrossRef]
38. Jiang, Q.; Yan, X.; Huang, B. Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes. *Ind. Eng. Chem. Res.* **2019**, *58*, 12899–12912. [CrossRef]
39. Zhu, J.; Ge, Z.; Song, Z. Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1877–1885. [CrossRef]
40. Bathelt, A.; Ricker, N.L.; Jelali, M. Revision of the Tennessee Eastman process model. *IFAC-PapersOnLine* **2015**, *48*, 309–314. [CrossRef]

41. Alakent, B. Early fault detection via combining multilinear PCA with retrospective monitoring using weighted features. *Braz. J. Chem. Eng.* **2024**, *41*, 1–23. [CrossRef]
42. Kinney, J.B.; Atwal, G.S. Equitability, mutual information, and the maximal information coefficient. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 3354–3359. [CrossRef]