

Article

# Fused Data-Driven Approach for Early Warning Method of Abnormal Conditions in Chemical Process

Xiaomiao Song<sup>1</sup> , Fabo Yin<sup>2</sup> and Dongfeng Zhao<sup>3,\*</sup> 

<sup>1</sup> College of Mechanical and Electrical Engineering, China University of Petroleum (East China), Qingdao 266580, China

<sup>2</sup> Center for Safety, Environmental & Engineering, China University of Petroleum (East China), Qingdao 266580, China

<sup>3</sup> College of Chemical Engineering, China University of Petroleum (East China), Qingdao 266580, China

\* Correspondence: b19040002@s.upc.edu.cn

**Abstract:** The utilization of data-driven methods in chemical process modeling has been extensively acknowledged due to their effectiveness. However, with the increasing complexity and variability of chemical processes, predicting and warning of anomalous conditions have become challenging. Extracting valuable features and constructing relevant warning models are critical problems that require resolution. This research proposed a novel fused method that integrates K-means density-based spatial clustering of applications with noise (DBSCAN) clustering and bi-directional long short-term memory multilayer perceptron (Bi-LSTM-MLP) to enable early warning of abnormal conditions in chemical processes. The paper applied the proposed method to analyze the early warning using actual process data from Eastman Tennessee and the atmospheric pressure reduction unit as an example. In the TE model and example, the root mean square error (RMSE) of this method is 0.006855 and 0.052546, respectively, which is quite low when compared to other methods. The experimental results confirmed the effectiveness of our approach.

**Keywords:** data-driven; chemical process; abnormal conditions



**Citation:** Song, X.; Yin, F.; Zhao, D. Fused Data-Driven Approach for Early Warning Method of Abnormal Conditions in Chemical Process. *Processes* **2023**, *11*, 2435. <https://doi.org/10.3390/pr11082435>

Academic Editor: Gade Pandu Rangaiah

Received: 29 June 2023

Revised: 31 July 2023

Accepted: 8 August 2023

Published: 12 August 2023



**Copyright:** © 2023 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

Chemical processes can pose a significant risk to safety and efficiency due to their potential for flammability, explosiveness, toxicity, and hazardous reactions. Abnormal conditions generated during chemical processes can cause disturbances in multiple variables that transmit across the entire process through logistics and equipment connections [1], leading to numerous alarms and abnormal conditions. Therefore, monitoring and warning of abnormal conditions are essential in the chemical industry to ensure plant safety and efficiency [2]. Traditional monitoring techniques rely on pre-set alarm thresholds, but early warning by simply lowering these thresholds can lead to an “alarm flood” [3]. The three main types of abnormal working condition warning techniques currently in use are analytical model-based techniques, knowledge-based techniques, and data-driven techniques [4]. Knowledge-based methods analyze and reason using knowledge and experience, but obtaining knowledge and experience is more challenging [5]. Data-driven methods predict changes in the data via mining. Analytical model-based methods require the establishment of a rigorous scientific data model, and the process is more complex.

With the advent of plant distributed control system (DCS) systems, large amounts of data require mining and analysis, making data-driven methods a hot topic for research in abnormality monitoring and fault diagnosis. Data-driven methods are categorized into statistical-based methods, artificial intelligence-based methods, and comprehensive methods [6]. Statistical-based methods involve projecting high-dimensional data into low-dimensional space, applying statistical principles to calculate information and statistics,

and comparing these results to threshold values for analysis. Principal component analysis (PCA) [7], partial least squares (PLS) [8], independent component analysis (ICA) [9], and Gaussian mixture model (GMM) [10] are examples of statistical-based methods that have strong data processing capabilities but weak diagnostic and analytical capabilities. Artificial-intelligence-based methods rely on machine learning and are categorized as supervised or unsupervised [11]. Supervised learning involves error minimization via training input-output relationships, while unsupervised learning can mine data relationships without example outputs. Decision trees [12], artificial neural networks [13], deep learning [14], support vector machines [15], integrated learning [16], k-means [17], and autoencoders [18] are all examples of AI-based methods. However, the stochastic nature of these models can lead to over-reliance on data quality. Integrated approaches are more scalable by applying statistical principles to AI methods, with Bayesian networks [19] and hidden Markov models [20] being examples of integrated approaches. Existing warning methods do not consider the characteristics of unlabeled dynamic data, resulting in weak generalization ability, poor stability, and limited effectiveness, accuracy, and early warning capabilities.

An investigation into the alarm systems of DCS systems in various chemical plants reveals that the chemical process data captured typically lack classification labels. Manually adding such labels would entail significant labor costs and be susceptible to errors introduced by human factors. As a result, chemical process early warning researchers have explored unsupervised learning methods. Among these methods, k-means, which is a widely employed partitioned clustering algorithm, has demonstrated notable outcomes in numerous applications owing to its fast convergence and simplicity [21–26]. For instance, Cai et al. [27] put forth a new clustering method that combined k-means with the MDS method to visualize the alarm vector and display results in a two-dimensional scatter plot, resulting in the removal of redundant alarms and making the alarm system more effective. Additionally, Zhang [28] and colleagues implemented an automatic hierarchical warning function via the K-means algorithm and utilized an advanced diagnostic model to achieve an intelligent diagnosis, successfully applying the model to axial flow fans. Validation was performed, underscoring the importance of feature selection and clustering in modeling and the potential of combining k-means with deep learning methods [29]. Nevertheless, a large number of studies have demonstrated that k-means and its related improved algorithms are fundamentally reliant on evaluating the similarity of sample points based on their distances from one another, resulting in an inability to differentiate anomalies in local density differences and leading to significant misclassification of normal data fluctuations. Moreover, for anomaly monitoring, the classical density-based algorithm, density-based spatial clustering of applications with noise (DBSCAN), is regarded as capable of identifying anomalous samples in the low-density regions of the sample space [30–32]. Garg et al. [33] proposed a novel integration-based anomaly detection technique, Boruta-Firefly-aided partitioning DBSCAN (BFA-PDBSCAN), which utilizes locality sensitive hashing (LSH) and k-distance graphs to address traditional problems of DBSCAN. Furthermore, Zhu et al. [34] used the DBSCAN approach to filter and improve the validity of the data acquired from monitoring and controlling wind turbine conditions through supervisory control and data acquisition (SCADA) data. Given the stochastic nature of the development of data features under abnormal states during the chemical process, further investigation is required to determine the adaptive nature of clustering parameter selection.

Moreover, given the high applicability, learning ability, processing power, and robustness of unlabeled data from deep learning methods, researchers have made several findings in this area [35–38]. Considering that most of the data collected during chemical processes are dynamic and that traditional applications are prone to classifying them as static data processing, researchers have turned their attention to the better dynamic data processing capabilities of long short-term Memory (LSTM) networks for abnormal data monitoring [39–41]. Ren et al. [42] improved the monitoring performance in the nonlinear batch process by processing the LSTM encoder–decoder network. Zheng et al. [43] combined system architecture evolution (SAE) with LSTM to extract low-dimensional

features and construct a pseudo-supervised model for online fault diagnosis. Bai et al. [44] proposed a dynamic intra-principal component analysis (DiPCA) and LSTM-based chemical process key alarm variable prediction model, reducing the complexity of the original prediction problem. Zhang et al. [45] combined LSTM with a ladder autoencoder (LAE), using semi-supervised learning to utilize unlabeled data and facilitate fault localization, resulting in a significant improvement in fault diagnosis performance. The research shows the effectiveness of LSTM in the direction of chemical process early warning.

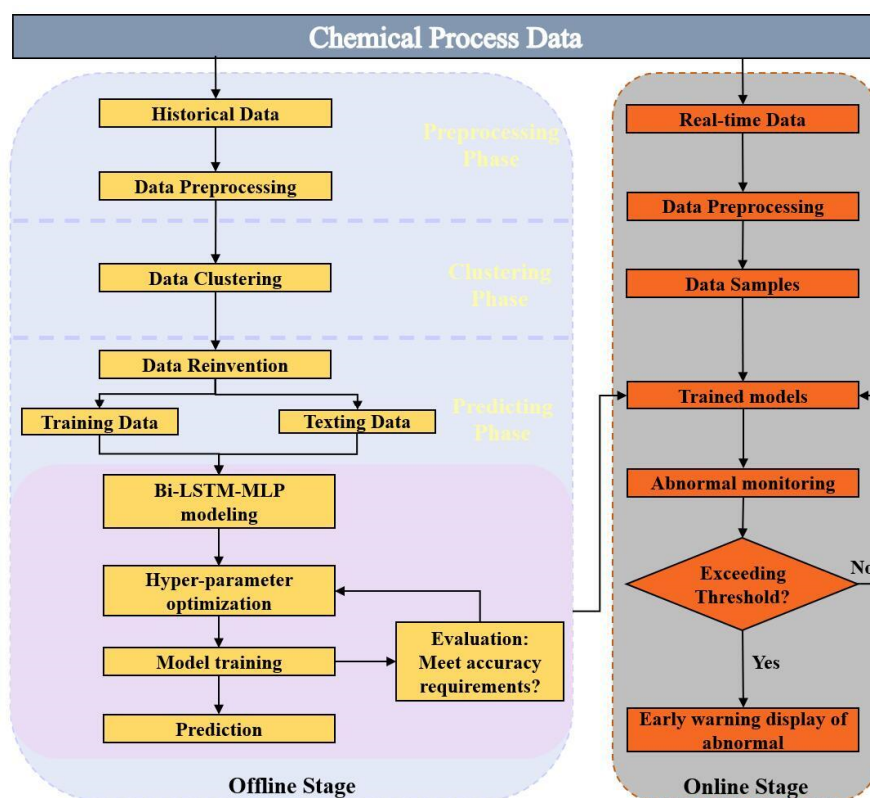
The unique characteristics of data in chemical processes, such as dynamic and unlabeled features, pose challenges for early warning and trend prediction of abnormal data development. This paper proposes a novel data-driven ultra-early warning method for chemical processes to improve the sensitivity and accuracy of abnormal data detection. The proposed method integrates two clustering methods, k-means and DBSCAN, to cluster the alarm data and capture the temporal dependence in the data using a better-integrated LSTM prediction method. Additionally, the multilayer perceptron (MLP) layer is fused to learn the nonlinear relationship between the features and target variables. The main contributions of this paper are as follows:

- It is suggested to develop a better pre-warning technique for abnormal working conditions in chemical processes that can anticipate the direction of data in advance to provide operators with more flexible processing time;
- The use of two data-driven fusion algorithms, the fusing of K-means and DBSCAN into two clustering techniques, as well as Bi-LSTM and MLP, enhances the scientific validity of the warning techniques;
- The experiments show how the proposed fusion data-driven technique for early warning of aberrant working conditions in chemical processes is reliable and sophisticated.

Section 2 outlines the methodology for developing the fused data-driven early warning method for chemical processes, which includes pre-processing, clustering, and prediction parts, and introduces the related models, including K-MEANS, DBSCAN, LSTM, and MLP. In Section 3, the application of the early warning method is demonstrated using specific examples based on a TE model and a normal pressure-reducing device in a petrochemical plant. The conclusions are presented in Section 4.

## 2. Methodology

The present study introduces a novel fused data-driven early warning method to identify and predict abnormal working conditions in chemical processes, as depicted in Figure 1. To begin with, the original data undergo a pre-processing stage that includes cleaning, missing value imputation, and normalization. Next, the pre-processed data undergo a clustering stage using K-means and DBSCAN algorithms. Finally, the clustered data are fed as input to the Bi-LSTM-MLP model for prediction and warning output. The outlined approach is visualized in Figure 1, providing a clear and concise representation of the proposed method.



**Figure 1.** Fused Data-Driven Approach for Early Warning Method of Abnormal Working Conditions in Chemical Processes.

### 2.1. Data Pre-Processing Layer

The data pre-processing layer involves the critical step of data cleaning, which entails identifying and addressing issues such as missing values, outliers, and errors in the dataset. Following this, the data undergo feature scaling, where the normalization method is adopted, utilizing the equation as follows:

$$x^i_{(\text{normal},j)} = \frac{x^i(t,j) - x^i_{j\text{min}}}{x^i_{j\text{max}} - x^i_{j\text{min}}}, (j = 1, 2, \dots, n) \quad (1)$$

$$x^i_{\text{average}} = \frac{1}{n} \sum_{j=1}^n x^i_{(\text{normal},j)} \quad (2)$$

where  $x^i(t,j)$  is the original data and  $x^i_{(\text{normal},j)}$  is the normalized data.

The subsequent step in the data preprocessing layer is feature selection. The normalized data are used as input, and a sliding time window is employed to map the series from a one-dimensional space to a higher-dimensional space. The historical neighborhood of a specific moment is divided into multiple time windows, and 15-dimensional time domain features are extracted from each window, which include maximum, minimum, peak, peak-peak, mean, variance, mean square, mean square amplitude, square root amplitude, mean amplitude, peak indicator, waveform indicator, pulse indicator, margin indicator, and cliff indicator. A sliding window of width  $\omega$  is set in the historical time neighborhood  $[T + 1, t]$  for time point  $t$  with a sliding factor  $\delta = 1$ . The method fragments the  $s$  monitoring data using a sliding window to extract the 15-dimensional time-domain features within a total of  $(T - \omega + 1)$  time windows. This results in the creation of an associated measurement point feature space  $X = [X_1, X_2, \dots, X_{15}]$  which describes the changes in data within the window. The number of individual measurement point samples is  $N = T - \omega + 1$ . This process establishes the correlation between a moment of the process and its value in

the time neighborhood and satisfies the sample number requirement in the subsequent clustering detection.

The proposed method utilizes principal component analysis (PCA) to reconstruct the feature space and eliminate noise and redundant information. By mapping high-dimensional data to low-dimensional space, this method is capable of retaining the most important features while discarding irrelevant data. Specifically, PCA is applied to the 15-dimensional time-domain feature space, and its feature principal components are extracted based on a 95% feature contribution rate.

First, the feature samples are standardized to eliminate feature magnitude differences, and the covariance matrix is calculated to obtain the eigenmatrix and the eigenvalues and eigenvectors.

$$\sum_{ij} cov(X_i X_j) = E((X_i - \mu_i)(X_j - \mu_j)) \quad (3)$$

$$\lambda v = \sum v \quad (4)$$

The number of principal components is determined based on the cumulative contribution of the features, where the contribution represents the proportion of information contained in the first  $n$  dimensions of the principal component expression to that in the original feature set  $X$ . The dimensionality of the principal component  $n$  is determined when the cumulative contribution exceeds 95% (This value can be changed to reflect the current circumstances.)

$$P_n = \frac{\sum_{k=1}^n \lambda_k}{\sum_{k=1}^m \lambda_k} > 0.95 \quad (5)$$

Next, the projection direction of the principal component analysis is determined for the first  $n$  eigenvalue vectors, resulting in  $V_{15 \times n} = [v_1, v_2, \dots, v_n]$ . Finally, the original feature set  $X$  is projected in this direction to obtain the reduced dimensional feature space  $Y$ .

The PCA approach can be enhanced and treated differently depending on the data being used; just the base method is listed in this study.

## 2.2. Data Clustering Layer

### 2.2.1. K-Means

K-means clustering is a widely used unsupervised learning algorithm that partitions a set of data points into  $k$  clusters based on their similarity [46]. This algorithm consists of two fundamental steps [47]. Initially, the  $k$ -centroids are selected randomly as the center of mass for each cluster, followed by the assignment of each data point to the closest centroid based on the Euclidean distance between the data point and the centroid. The algorithm continues to iterate until convergence, with each iteration updating the centroid positions based on the new assignments of data points. Ultimately, K-means produces a set of  $k$ -clusters, with each cluster comprising data points that are similar to each other and dissimilar to the data points in the other clusters.

### 2.2.2. DBSCAN

DBSCAN is a clustering algorithm that effectively groups points based on their proximity while also identifying and isolating outliers [48]. It outperforms K-means in its ability to handle noisy and complex data, making it particularly useful for clustering chemical process data. DBSCAN employs two crucial parameters, the distance threshold  $\epsilon$  and the minimum number of points  $minPts$ , to define clusters [49]. The algorithm classifies data points as core points, boundary points, or noise points based on their local density. A point  $x_i$  is a core point if  $|N_i| \geq minPts$ , where  $N_i$  is the set of points within a distance of  $\epsilon$  from  $x_i$ . For each core point, DBSCAN creates a new cluster and adds all points within  $\epsilon$  distance of the core point to the cluster. These points are then marked as visited. Starting from an initial point, the algorithm forms a cluster by including all points that satisfy the core point criteria and fall within the  $\epsilon$  distance. The process continues until all unvisited points have been processed.

### 2.2.3. K-means DBSCAN Fusion Clustering Method

To effectively discriminate between abnormal data and normal fluctuations, the K-means clustering algorithm and the DBSCAN clustering algorithm have their respective advantages and limitations. While the K-means clustering algorithm can identify the spatial variability of the features, it sometimes misjudges normal fluctuations. On the other hand, the DBSCAN clustering algorithm can effectively detect the density difference of the feature spatial distribution, but determining the neighborhood radius (Eps) is dependent on the a priori information of the feature spatial distribution. To address these limitations, a K-means DBSCAN fusion clustering method is proposed. To begin the K-means DBSCAN fusion clustering method, first perform K-means clustering with  $k = 2$  on the feature space to identify feature location differences. Then assign class labels to each measurement point based on the majority cluster to which its samples belong to avoid misjudgments due to individual samples. Next, examine whether any measurement points belong to isolated classes and obtain a priori information on the feature distribution. If an isolated class is identified, use the maximum radius of the majority clusters as the neighborhood radius Eps and the number of samples from a single measurement point as the minimum neighborhood density Minpts for DBSCAN clustering. Then determine whether there is a significant difference in the distribution density, and if the characteristics of the measurement points satisfy both the location and distribution density variability criteria, judge them as anomalous data. The detailed steps of the process are provided below.

Based on the principal component feature set  $Y$ , K-means clustering is applied to the association process data features with a value of  $k = 2$ . The sample class  $label$  ( $label = 1, 2$ ) is assigned to each association measurement point  $k$  ( $k = 1, 2, \dots, s$ ); the majority class to which a measurement point's sample belongs is determined based on the label count of each measurement point.

$$c(i) = \begin{cases} 1, & label(i) = 1 \\ 0, & otherwise \end{cases} \quad (6)$$

$$class(k) = \begin{cases} 1, & \sum_{i=1}^N c(i) > \frac{N}{2} \\ 2, & otherwise \end{cases} \quad (7)$$

The obtained measurement point class labels are then counted to identify the presence of any isolated class.

$$C(k) = \begin{cases} 1, & class(k) = 1 \\ 0, & otherwise \end{cases} \quad (8)$$

$$outlier = \begin{cases} 1, & \sum_{i=1}^N C(k) < r_0 \text{ or } s - \sum_{i=1}^N C(k) < r_0 \\ 2, & otherwise \end{cases} \quad (9)$$

The discriminant threshold for an isolated class test point  $r_0$  is used to classify a test point as an isolated class test point if the number of samples belonging to that class is less than  $r_0$ .

In case K-means clustering detects any isolated points, the maximum cluster radius of most clusters is calculated using the following equation:

$$\frac{d_{max}}{2} = \frac{\max(dist(x_i, x_j))}{x} \quad i, j = 1, 2, \dots, n \quad (10)$$

where  $n$  refers to the number of samples in most clusters.

Following the identification of the sparse distribution of abnormal data and the relatively concentrated distribution of most normal data, DBSCAN clustering is performed to further distinguish and isolate the anomalous samples.

The resulting cluster of anomalous data points is denoted as  $D = \{p_1, p_2, \dots, p_m\}$ , where each data point is  $y$   $p_i$  with density  $\rho_i = |N_\varepsilon(p_i)|$ ,  $N_\varepsilon(p_i) = \{p_j \in D \mid \text{dist}(p_i, p_j) \leq \varepsilon\}$ . Here  $\text{dist}(p, q) = \|p - q\|_2$ , and  $|N|$  indicates the number of data points in set  $N$ .

$$\varepsilon = \frac{d_{max}}{2} \quad (11)$$

$$Minpts = n \quad (12)$$

The density of each data point in  $D$  is then calculated according to Equation (10).

If a data point  $p_i \in D$  satisfies  $\rho_i \geq MinPts$ , it is deemed to be a core point; conversely, if  $\rho_i < MinPts$ , it is considered a boundary point. The set of core points in  $D$  is designated as  $C$ .

$$R \leftarrow C$$

$$c \leftarrow 0$$

while  $R \neq \emptyset$  do

$$\left| \begin{array}{l} c \leftarrow c + 1 \\ \forall p_i \in R; \\ \mathcal{C}_c \leftarrow \{p_i\} \\ \mathcal{C}_c \leftarrow \mathcal{C}_c \cup \{p_j \mid p_j \text{ is the } p_i \text{ density reachable from}\} \\ \mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{C}_c; \end{array} \right. \quad (13)$$

end

$$\mathcal{C}_0 \leftarrow D \setminus \bigcup_{i=1}^c \mathcal{C}_{i_0}$$

The algorithm proceeds by assessing the distribution density difference between the majority and minority classes and identifying the clustering results of the samples belonging to each class. Specifically, if the clustering results indicate that the majority class samples are classified as the same kind, while the minority class samples are all identified as noise, and the difference in distribution density between the two classes is significant, then the noise class is classified as anomalous. In cases where the distribution density of the majority and minority classes is similar, the DBSCAN clustering result yields samples belonging to the same kind, and anomalous data are absent.

The misjudgment rate and the silhouette coefficient are the evaluation indicators used to assess the accuracy and superiority of the clustering method. The probability of classifying data points that do not belong to the same class into the same class during the clustering process, or the ratio of the number of samples that were incorrectly classified to the total number of samples, is referred to as the misjudgment rate of a clustering algorithm. The clustering effect is assessed using the silhouette coefficient, which combines the two cohesion and separation indicators to simultaneously assess intra-cluster and inter-cluster differences of various clustering methods on the same sample. The equation reads as follows:

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad (14)$$

In the formula,  $a_i$  stands for the average distance between the sample and other samples in the cluster, or the similarity of samples in the same cluster, reflecting the degree of separation, while  $b_i$  stands for the average distance between the sample and all other samples in the nearest cluster, reflecting the degree of cohesion.  $S_i \in [-1, 1]$ , the optimal clustering is when there are few intra-cluster differences and many inter-cluster differences, or when  $b_i > a_i$ . The closer to 1 the clustering is, the more reasonable it is.

### 2.3. Data Prediction Layer

Upon completion of data clustering, the labeled clusters are utilized as input to the LSTM-MLP network to predict abnormal chemical working conditions. The LSTM

component captures time dependencies in the data, while the MLP part performs a non-linear mapping of input to output.

### 2.3.1. LSTM

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that has gained wide popularity in time series prediction due to its ability to remember patterns of a large number of sequences [50]. By using gates to regulate the flow of information, LSTM models are designed to capture temporal dependencies in data [51]. A typical LSTM network consists of multiple memory blocks, or cells, that enable the memorization of information. The input, forget, and output gates play crucial roles in regulating the flow of information in and out of the cells [52], like Figure 2.

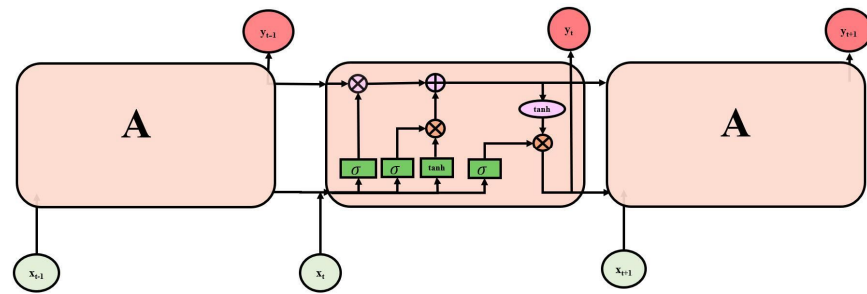


Figure 2. LSTM model.

The mathematical notation used in the LSTM model is described as follows: The input at time  $t$  is represented by  $x_t$ , and  $h_{t-1}$  represents the previous hidden state. The sigmoid activation function  $\sigma$  is used to control the flow of information through the model.  $W$  and  $U$  are matrices representing the weights associated with the input and hidden states, respectively, while  $b$  is a vector representing the bias term. The  $*$  operator denotes element-wise multiplication, which is used in the calculation of the input gate, forget gate, and output gate in the LSTM model.

In the LSTM cell, the input gate assesses the importance of new information carried by input data through the equation

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i). \quad (15)$$

Meanwhile, the forget gate determines whether to retain or delete historical information through

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \quad (16)$$

Lastly, the output gate determines which information to output.

The output gate:

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \quad (17)$$

The memory cell:

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \quad (18)$$

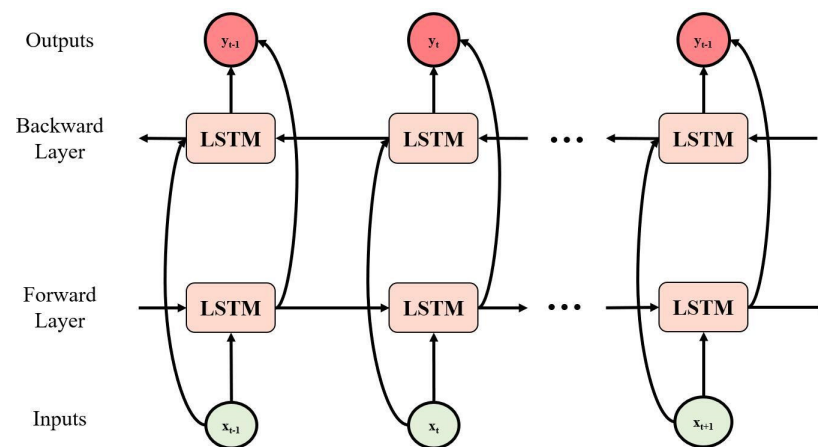
The hidden state:

$$h_t = o_t * \tanh(c_t) \quad (19)$$

The long short-term memory (LSTM) neural network is a type of unidirectional network used for long and short-term pattern recognition in time series data. In contrast, the bidirectional LSTM (Bi-LSTM) network is a type of bidirectional network that utilizes a positive and negative LSTM network in the training process, which are then combined in the output layer [53]. This approach allows the Bi-LSTM network to capture temporal dependencies from both past and future time steps, resulting in a more comprehensive and



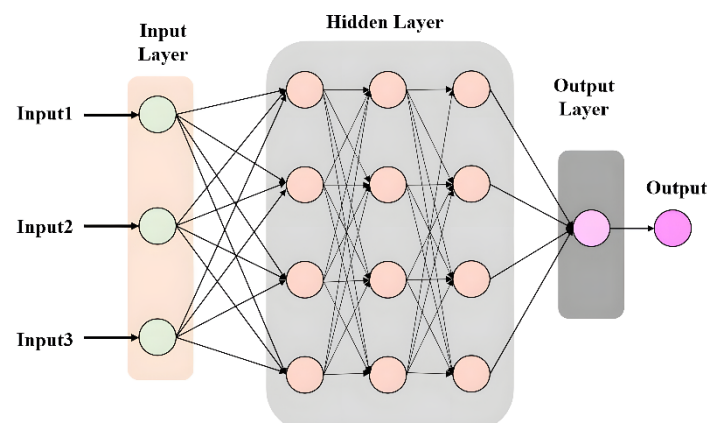
complete representation of time-series data compared to the unidirectional LSTM [54]. A schematic of the Bi-LSTM network structure is presented in Figure 3.



**Figure 3.** Bi-LSTM network structure.

### 2.3.2. MLP

While the Bi-LSTM network is known for its ability to capture both past and future information when extracting sequence data, it has been observed to exhibit slow convergence and dynamic prediction when mapping to target results [55]. To address this issue, this paper proposes deepening the fully connected layers by feeding the output of the Bi-LSTM into a multilayer perceptron. The MLP is a feed-forward neural network capable of handling high-dimensional data and nonlinear relationships and is commonly used in industry for nonlinear regression and classification tasks, as shown in Figure 4. The MLP achieves this by stacking multiple hidden layers composed of neurons that enable the network to learn complex relationships between input and output variables [56]. The training process of the MLP involves using a back-propagation method that iteratively minimizes the error between the actual and model outputs.



**Figure 4.** MLP model.

### 2.3.3. Bi-LSTM-MLP Fusion Method

Data reshaping is a crucial preprocessing step in building an effective Bi-LSTM model. This involves determining the number of layers of the LSTM, the number of neurons in each layer, and the input and output dimensions of the network. To incorporate anomalous observations after the clustering stage, the clustering assignment sequence must be reformatted into a structure that is compatible with the input Bi-LSTM-MLP model. A sliding window approach is employed to create fixed-length sequences, where the length

of the window corresponds to the number of time steps in the input sequence. For instance, if the length of the input sequence is set to 10, the first input sequence will contain cluster assignments of data points 1–10, the second input sequence will contain cluster assignments of data points 2–11, and so forth. The reshaped data are then represented as a three-dimensional tensor with dimensions  $[n_{samples} \ n_{timesteps} \ n_{clusters}]$ , where  $n_{samples}$  denotes the number of input sequences,  $n_{timesteps}$  is the length of each input sequence, and  $n_{clusters}$  is the number of clusters. A suitable partitioning ratio is employed to divide the input data into training, testing, and validation sets, depending on the size of the dataset and the complexity of the model. The reshaping process is carried out using three key parameters: sample size, time step, and number of features, with the reshaping function being varied based on the specific use case.

In this study, a Bi-LSTM-MLP model is proposed for processing the reshaped data. Specifically, the input sequence  $X_t$  is first passed through the Bi-LSTM layer in a bidirectional manner to capture the temporal dependence between cluster assignments. The output of this layer is a string of hidden states  $H$ , which has a dimension of  $[n_{samples} \ n_{timesteps} \ n_{hidden}]$ , where  $n_{hidden}$  is the number of hidden cells in the Bi-LSTM layer.

To calculate the hidden state for each time step, a specific approach is employed.

Forward hidden state:

$$h_{t_{fwd}} = LSTM_{Fwd}(x_t, h_{t-1_{fwd}}, c_{t_{fwd}}) \quad (20)$$

Backward hidden state:

$$h_{t_{bwd}} = LSTM_{Bwd}(x_t, h_{t+1_{bwd}}, c_{t_{bwd}}) \quad (21)$$

In the proposed Bi-LSTM-MLP model, the input sequence  $x_t$  is first processed in a bidirectional manner by the forward and backward LSTM cells denoted as  $LSTM_{Fwd}$  and  $LSTM_{Bwd}$ , respectively. At timestep  $t$ , the input  $x_t$  is used to compute the forward and backward hidden states  $h_{t_{fwd}}$  and  $h_{t_{bwd}}$ , as well as the forward and backward cell states  $c_{t_{fwd}}$  and  $c_{t_{bwd}}$ .

The output hidden state for timestep  $t$  is obtained by concatenating the forward and backward hidden states as  $h_t = [h_{t_{fwd}}; h_{t_{bwd}}]$ , where  $[:]$  denotes concatenation.

Following the Bi-LSTM layer processing, the output hidden states  $H$  are directed towards a multilayer perceptron (MLP) layer for mapping to the output space. The MLP layer, comprising one or more fully connected layers activated with functions such as sigmoid, generates a vector  $y$  of dimensions  $[n_{samples} \ n_{outputs}]$ , where  $n_{outputs}$  represents the number of output units within the MLP layer.

At each timestep  $t$ , the output is determined by first flattening the hidden state to obtain a 1-dimensional vector  $h_{t_{flat}}$  with dimensions  $[n_{hidden}^2]$ , where  $n_{hidden}^2$  corresponds to the concatenated forward and backward hidden state size.

The flattened hidden state is then passed through the MLP layer with appropriate weights and biases, yielding the output at timestep  $t$ , which can be represented as  $y_t = MLP(h_{t_{flat}})$ . The MLP layer is instantiated with the necessary weights and biases before being utilized in the model architecture.

The output of the previous layer will be utilized as the input of the current layer in the MLP input layer to the hidden layer process, and the calculation rule is frequently as follows: the product of the weights and the inputs plus the value of the bias. The formula is not provided here since experiments are used to determine the number of layers.

The loss function plays a crucial role in training the Bi-LSTM-MLP model. After obtaining the output vector  $y$  by passing the hidden state through the MLP layer, the model compares it to the true labels and computes the loss. The choice of loss function depends

on the task at hand and may include mean squared error for regression or cross-entropy loss for classification. The loss is denoted as

$$L = \text{loss}(y, y_{\text{true}}) \quad (22)$$

To train the model, an appropriate optimization algorithm, such as stochastic gradient descent or the Adam optimizer, is used to minimize the loss function. Instead of using parameter data from training, hyperparameters are parameters that are set to values before the model begins learning. Hyperparameters frequently need to be optimized, which involves experimenting with various hyperparameters before settling on a set that will increase learning performance. There are many hyperparameters. The model's hyperparameters in this study, including the learning rate, number of hidden units, and epochs, are fine-tuned via cross-validation.

Once trained, the model can be used to make predictions on new data by feeding the encoded cluster assignments through the Bi-LSTM and MLP layers, where the MLP layer produces the predicted labels for the input data. To obtain accurate predictions, it is essential to normalize the predicted data back to the original scale using the  $\text{inverse}_{\text{transform}}(X)$ . This function reverses the scaling applied to the data during preprocessing. By doing so, the predictions can be compared to the original data to obtain an accuracy metric.

$$y_{\text{pred}} = \text{MLP}(\text{BiLSTM}(X_t)) \quad (23)$$

To assess the performance of the proposed model, a held-out test set is utilized to estimate its performance. The effectiveness of the model is then validated through the computation of various metrics such as the root mean square error (RMSE).

The RMSE serves as a measure of the average difference between the predicted and actual values, with the square root taken to ensure that the units of measurement are the same as the original values. The resulting RMSE value ranges between 0 and 1, with 0 being the most favorable outcome.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (24)$$

Additionally, the trend of RMSE is used to monitor the working condition of the chemical process. By monitoring the fluctuation of the residual RMSE, the operation status of the process can be accurately identified. When the changing trend exceeds or maintains the threshold level, it can be inferred that the chemical process has changed and is in an abnormal working condition.

#### 2.3.4. Early Warning

The warning signs are inputted beforehand to guarantee the promptness and accuracy of early warning. The historical state of the unit operation, process indicators, and alarm setpoints are all employed as early inputs in this method to extract the normal operating state and values with the help of the alarm management operator, as shown in Figure 5. The yellow line in the figure is supposed to represent the actual data value, the blue line is assumed to represent the anticipated value, and the red dashed line in the figure is assumed to represent the defined warning indicator. An early warning signal will be sent out before the DCS alarm when the anticipated data value exceeds the predetermined alarm value. This signal alerts the operator to pay attention and take action by warning that an abnormal situation is likely to arise.

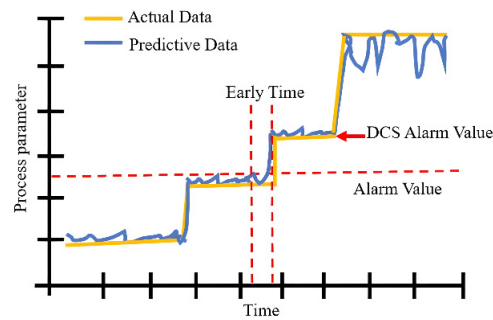


Figure 5. Indication of Early Warning Signals.

### 3. Analysis and Verification of Examples

The experimental setup for the retouching task involved conducting all experiments on an Ubuntu 16.04 operating system, utilizing an i8-CPU with 24 GB of RAM and a GPU Tesla3090. The implementation was performed using Python 3.7 and PyTorch 1.8. The proposed method was validated via the application of the TE model and real-world cases.

#### 3.1. TE Model

This study proposes an approach for monitoring the Tennessee Eastman (TE) process using a benchmark approach. The TE process comprises five reaction units, namely the reactor, condenser, separator, stripper, and compressor, and contains material transfer and control loops, as shown in Figure 6. The process involves 41 measured variables, including 19 component-measured variables and 22 continuous process-measured variables, and 12 control variables (Table 1), with 21 pre-defined failure modes, 16 of which are known and 5 are unknown. In this study, the training set is a set of data from normal operating circumstances, the simulation period is 48 h, and the sampling frequency is 3 min/time. The test set is chosen to include both a random change fault and a richer typical step change fault to better demonstrate the sensitivity of the suggested technique for early warning monitoring. A common step change error is fault 1, which is a step change in the A/C feed flow ratio. The reactor cooling water intake temperature fluctuates randomly under fault 11, which is more challenging to identify than a step change fault. At the start of the simulation, or eight hours in, both faults 1 and 11 were added.

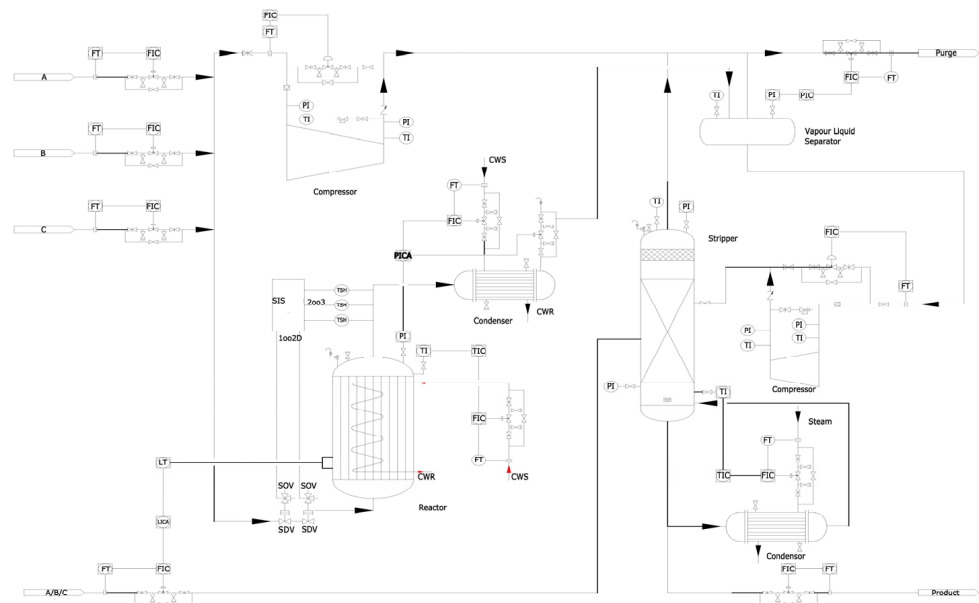
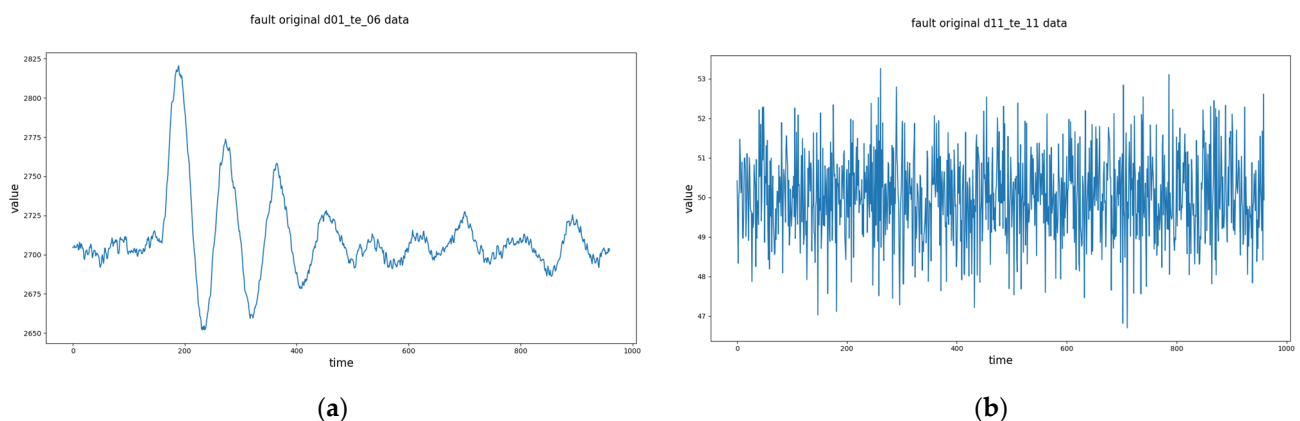


Figure 6. TE Model.

**Table 1.** TE Process Variables.

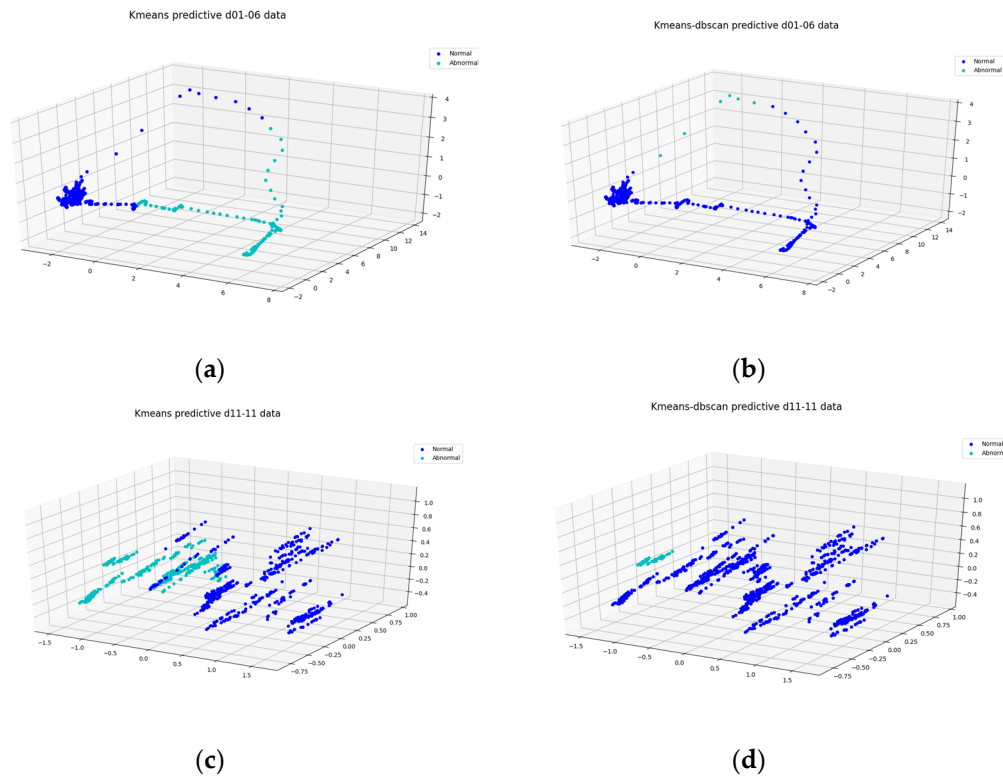
No.	Process Variable	No.	Control Variable	No.	Component Variable
1	A Material flow rate	1	D feed volume (stream 2)	1	A6
2	D Material flow	2	E feed volume (stream 3)	2	B6
3	E Material flow rate	3	A feed volume (stream 1)	3	C6
4	A and C mixture flow rate	4	Total feed volume (stream 4)	4	D6
5	Recovery flow rate	5	Compressor recirculation valve	5	E6
6	Reactor feed rate	6	Discharge valve (stream 9)	6	F6
7	Reactor pressure	7	Separator tank bottom flow (stream 10)	7	A9
8	Reactor level	8	Vapor extractor liquid product flow rate (Stream 11)	8	B9
9	Reactor temperature	9	Vapor stripper water flow valve	9	C9
10	Emptying rate	10	Reactor cooling water flow	10	D9
11	Product separator temperature	11	Condenser cooling water flow rate	11	E9
12	Product separator level	12	Stirring speed	12	F9
13	Product separator pressure	13		13	G9
14	Product separator outlet flow rate	14		14	H9
15	Separator level	15		15	D11
16	Separator pressure	16		16	E11
17	Extraction tower outlet flow rate	17		17	F11
18	Vapor extraction tower temperature	18		18	G11
19	Vapor Flow Rate	19		19	H11
20	Compressor operating power				
21	Reactor cooling water outlet temperature				
22	Condenser cooling water outlet temperature				

The product separator temperature of fault 11 and the reactor feed rate variable of fault 1 are chosen as the display because adding too many variables will cause results to become hazy. The raw data are first visualized in Figure 7, which allows for a series of subsequent preprocessing steps.



**Figure 7.** Raw data visualization. (a) The original data of fault 1 reactor feed rate. (b) The original data of fault 11 product separator temperature.

The trials on the pre-processed data used the clustering method described in Section 2.2. Figure 8 and Table 2 both feature comparisons to show how the proposed combined k-means and DBSCAN method outperforms a single clustering method. Figure 8 shows typical data in dark blue, while abnormal data are shown in cyan. It can be seen that the findings of the right b, d plot are more accurate when comparing the two variables' k-means clustering results (a, c) with the combined clustering results (b, d). The results of Table 2's comparison of the results using misclassification rates and contour coefficients show that the suggested fusion clustering approach performs better on this dataset than the single clustering method and can successfully distinguish between normal and abnormal data.



**Figure 8.** Comparison of clustering method predictions. (a) K-means clustering result for fault 1 reactor feed rate. (b) K-means DBSCAN clustering result for fault 1 reactor feed rate. (c) K-means clustering result for fault 11 product separator temperature. (d) K-means DBSCAN clustering result for fault 11 product separator temperature.

**Table 2.** Comparison of evaluation results of clustering methods for TE model.

Clustering Method	Profile Coefficient	False Positive Rate
K-means	0.8665	6.658%
DBSCAN	0.9024	4.374%
K-means DBSCAN	0.9564	1.256%

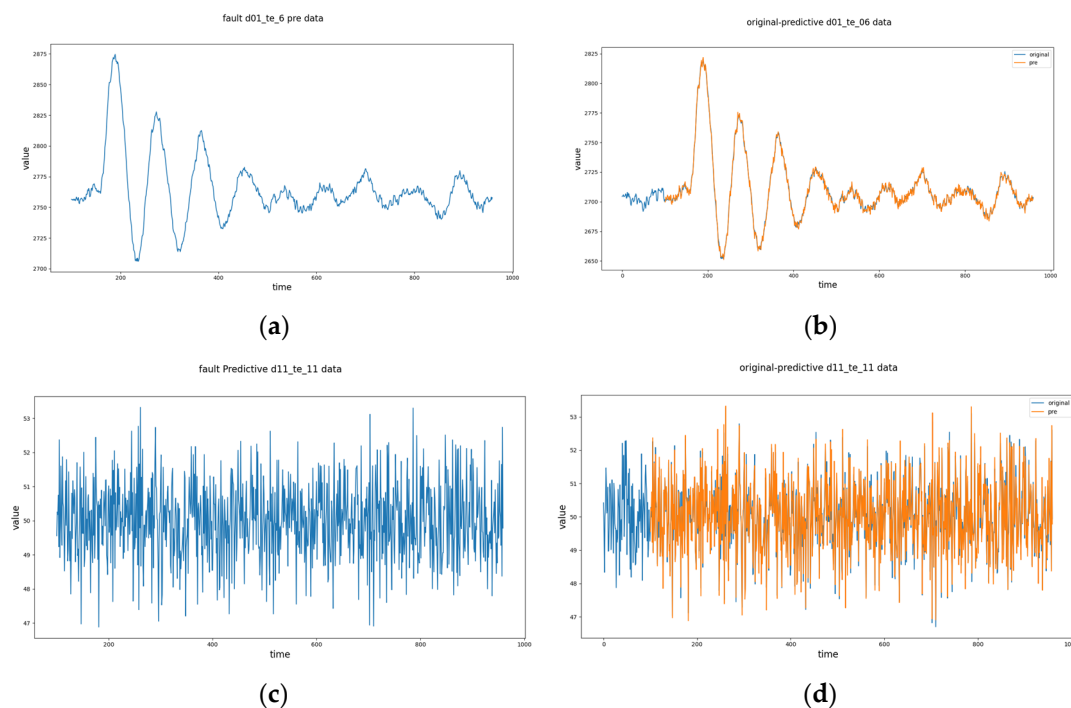
The model was evaluated with 11 distinct configurations applied to the activation functions of Bi-LSTM + MLP (ReLU, ELU, Tanh, SiLU, LeakyReLU), which differ in the number of various Bi-LSTM (1,2,3) and MLP layers (1,2,3). A feature introduced to artificial neural networks is the activation function, which aids in the network's ability to recognize intricate patterns in data. Layer count is a parametric evaluation metric that assesses the model's complexity and capacity for learning; using a simple model may result in underfitting issues, whilst using a complex model may result in overfitting issues. Table 3 provides an explanation of the model's performance on the test dataset using the RSME

as the results' evaluation metric. The Bi-LSTM + MLP model performs best with 1-layer Bi-LSTM and 1-layer MLP when employing the LeakyReLU activation function for this collection of data, as can be seen by the bolding of each best result.

**Table 3.** Comparison of the results of different settings of the Bi-LSTM-MLP model for TE model.

Method	Differences	RMSE
Different activation functions		
BiLSTM + MLP	ReLU	0.009854
BiLSTM + MLP	ELU	0.010254
BiLSTM + MLP	Tanh	0.012547
BiLSTM + MLP	SiLU	0.032874
BiLSTM + MLP	LeakyReLU	0.006854
Bi-LSTM different layers		
BiLSTM + MLP	1	0.006854
BiLSTM + MLP	2	0.015684
BiLSTM + MLP	3	0.042541
MLP different layers		
BiLSTM + MLP	1	0.006854
BiLSTM + MLP	2	0.025641
BiLSTM + MLP	3	0.051254

Figure 9 compares the actual data with the expected data to display the prediction results for the two variables in the proposed model. The results show that the fit is good and can reflect the validity and reliability of the model prediction.



**Figure 9.** Predicted results and comparison of predicted data and original-predictive results. (a) Prediction results of fault 1 reactor feed rate. (b) Results of comparing the original and predicted data for fault 1 reactor feed rate. (c) Prediction results of fault 11 product separator temperature. (d) Results of comparing the original and predicted data for fault 11 product separator temperature.

The results of this comparison, which are displayed in Table 4, further indicate the superiority of the fused Bi-LSTM + MLP model over competing models. The suggested fused Bi-LSTM + MLP model has the best prediction result among the six models, as shown

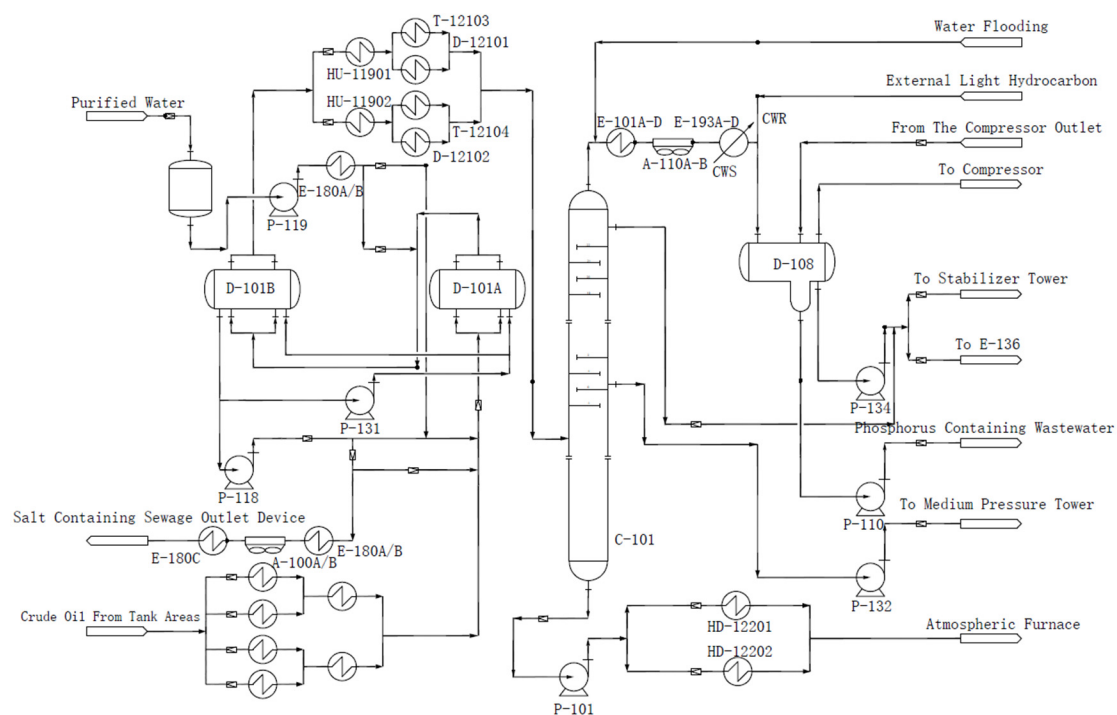
by the RMSE, which also shows that the model has a better RMSE value of 0. This shows that the fused Bi-LSTM + MLP model is more capable of forecasting abnormal working circumstances for chemical processes. Data are more useful.

**Table 4.** Comparison of the results of various prediction models for TE model.

Method	RMSE
KNN regression model	0.086214
SVM regression model	0.058621
LSTM	0.023242
LSTM + MLP	0.010145
BiLSTM	0.015254
BiLSTM + MLP	0.006854

### 3.2. Real Atmospheric Depressurization Unit

The present study applies the proposed method to the electric desalination process of a 10-million-ton/year normally reduced pressure distillation unit in a refinery. Specifically, the electric desalination process involves injecting water and an emulsion breaker into crude oil to remove salt from the oil via the action of a high-voltage electric field. This process is a critical component of the normal reduced pressure unit, which is depicted in the accompanying Figure 10. To confirm the accuracy of the suggested model, use actual process data from this typically depressurized unit. The period covers the complete recording of abnormal working conditions associated with crude oil and water accidents.



**Figure 10.** Flow chart of atmospheric pressure reducing device.

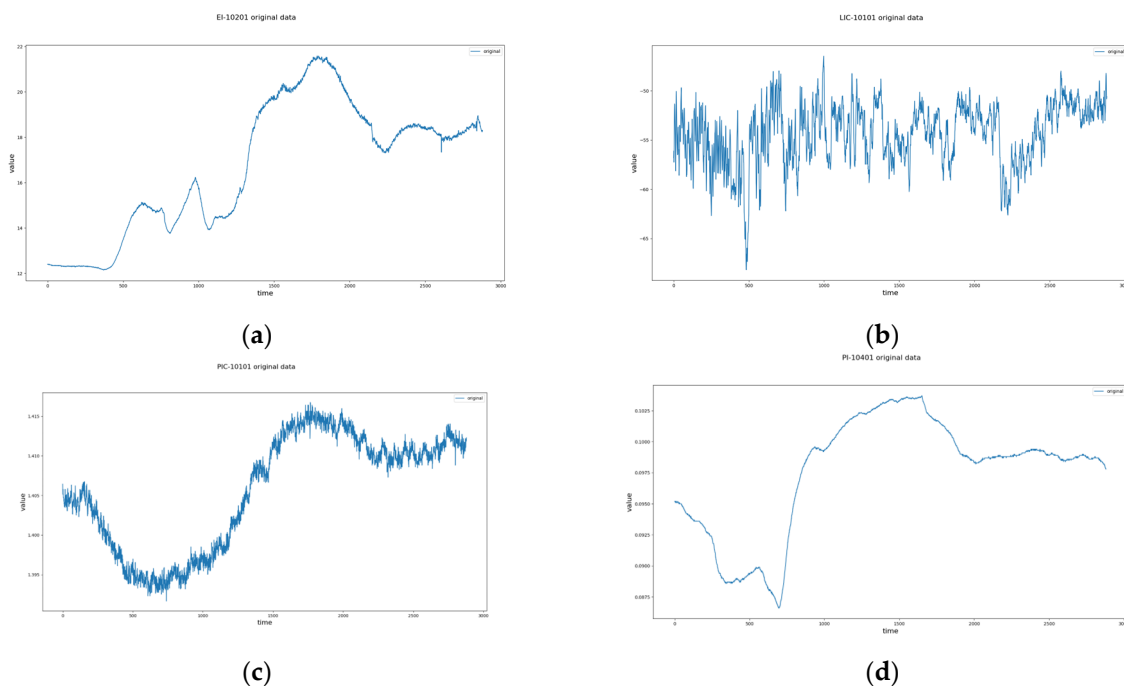
There are numerous variables in Figure 10, and we tested the model for 29 of them before identifying the nine most important factors that are directly associated with incidents involving crude oil and water in Table 5. Four of them—EI10201 for the electric desalination tank current, LIC10101 for the electric desalination tank boundary level, PIC10101 for the electric desalination tank pressure, and PI10401 for the top pressure of the primary distillation tower—are shown due to the condensed nature of the paper. These four contain more information about the accident condition than the others. All of the example's data



were obtained from the chemical plant's distributed control system (DCS) system, and they were all collected over 380 h using a total of 273,600 samples at intervals of 5 s. The raw data for the four chosen variables are shown in Figure 11 and will be further processed in the data pre-processing stage. It is important to note that in instance 4.2, we only chose a data set with a length of 3000 as validation data to verify the correctness of the suggested model due to the limitations of the graphical display.

**Table 5.** Instances of partial variables.

Bit No.	Description
EI10201	Electric desalination tank current
FIC10101	Electric desalination tank
LIC10101	Electric desalination tank boundary level
PIC10101	Electric desalination tank pressure
TI10102	Electric desalination tank inlet temperature
PDIC10101	Primary distillation column top fractionation tank pressure
TI10801	Primary distillation tower top cold after temperature
PI10401	Primary distillation tower top pressure
FIC11703	Crude oil feed volume



**Figure 11.** Raw data visualization. (a) The original data of EI10201. (b) The original data of LIC10101. (c) The original data of PIC10101. (d) The original data of PI10401.

Figure 12a–h demonstrates how each of the four variables is projected in three dimensions using K-means and the two alternative clustering techniques suggested in Section 2.2, K-means and DBSCAN, combined. It can be seen that in Figure 12, the cyan hue represents abnormal data, which has a smaller range, and the dark blue color represents normal data, which has a vast and dense range. Figure 12's four comparisons (a–b, c–d, e–f, and g–h) show that clustering utilizing k-means in combination with DBSCAN is capable of having a more precise range of anomaly identification. Table 6 shows the suggested method's superiority in terms of contour coefficients and false positive rates when compared to k-means alone and DBSCAN alone clustering, respectively. The prediction model will be further trained using the normal and abnormal data split into distinct columns in Figure 12.

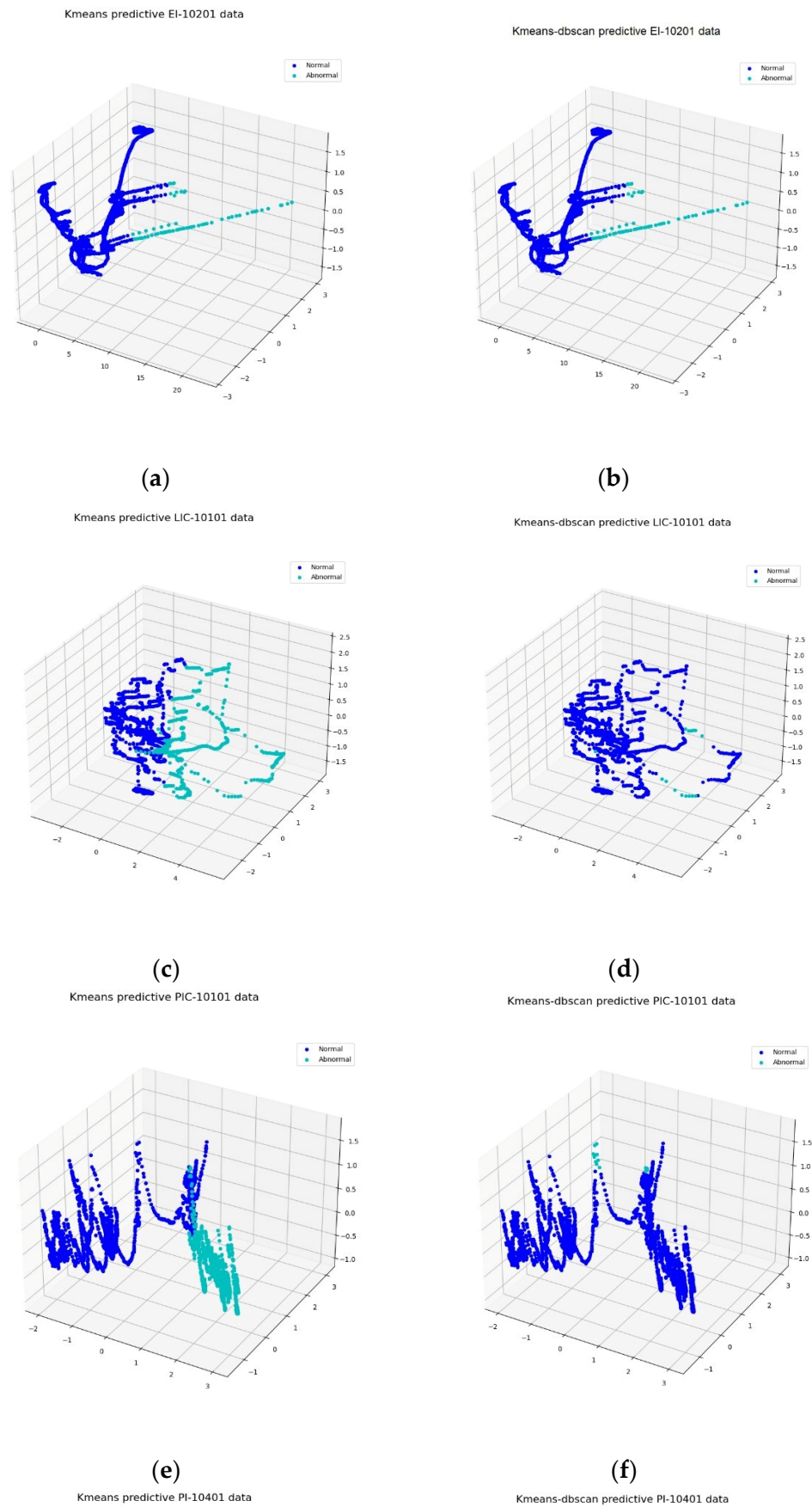
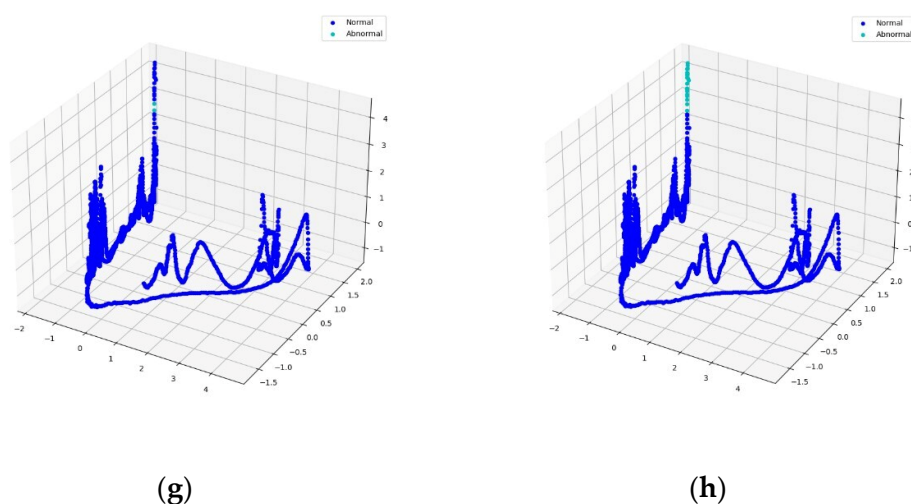


Figure 12. Cont.



**Figure 12.** Comparison results of clustering method predictions. (a) K-means clustering result for EI10201. (b) K-means DBSCAN clustering result for EI10201. (c) K-means clustering result for LIC10101. (d) K-means DBSCAN clustering result for LIC10101. (e) K-means clustering result for PIC10101. (f) K-means DBSCAN clustering result for PIC10101. (g) K-means clustering result for PI10401. (h) K-means DBSCAN clustering result for PI10401.

**Table 6.** Comparison of evaluation results of clustering methods for atmospheric depressurization unit.

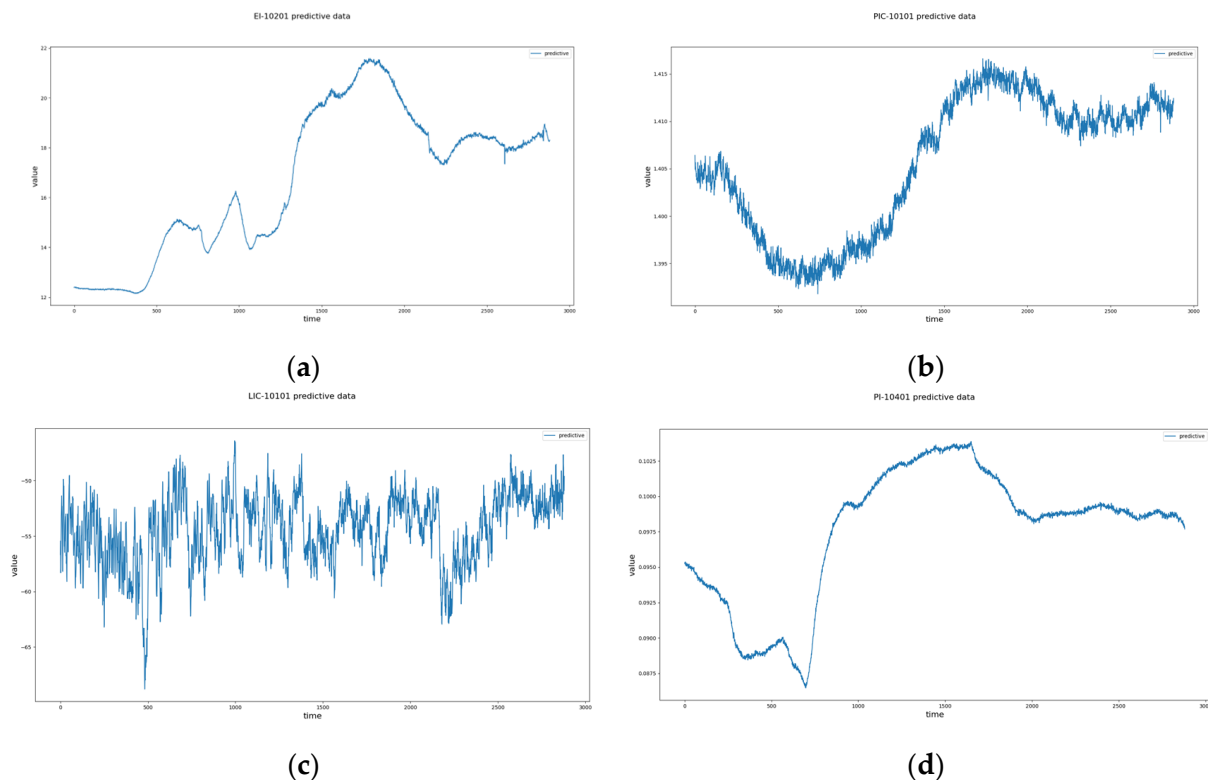
Clustering Method	Profile Coefficient	False Positive Rate
K-means	0.7541	9.985%
Dbscan	0.8214	7.565%
K-means Dbscan	0.9214	4.258%

To find the best configuration for a better deep learning network and avoid overfitting, both Bi-LSTM and MLP must be tested based on the data. Table 7 displays the outcomes of applying various configurations for testing. Once more, the RMSE values for various situations are selected to aid in selecting the optimum setup. LeakyReLU() is chosen as the activation function because, as can be seen, it has the lowest RMSE value among the various activation functions. This group of data performs best when 2 layers are chosen for the number of layers, and 2 layers also perform well when choosing the number of MLP layers. The second set of real data has a lower RMSE index than the first TE model, and the best number of layers for BiLSTM and MLP are also different. This is likely because the second set of real data is a real collection, which gets noisier over time and necessitates a deeper network for fitting.

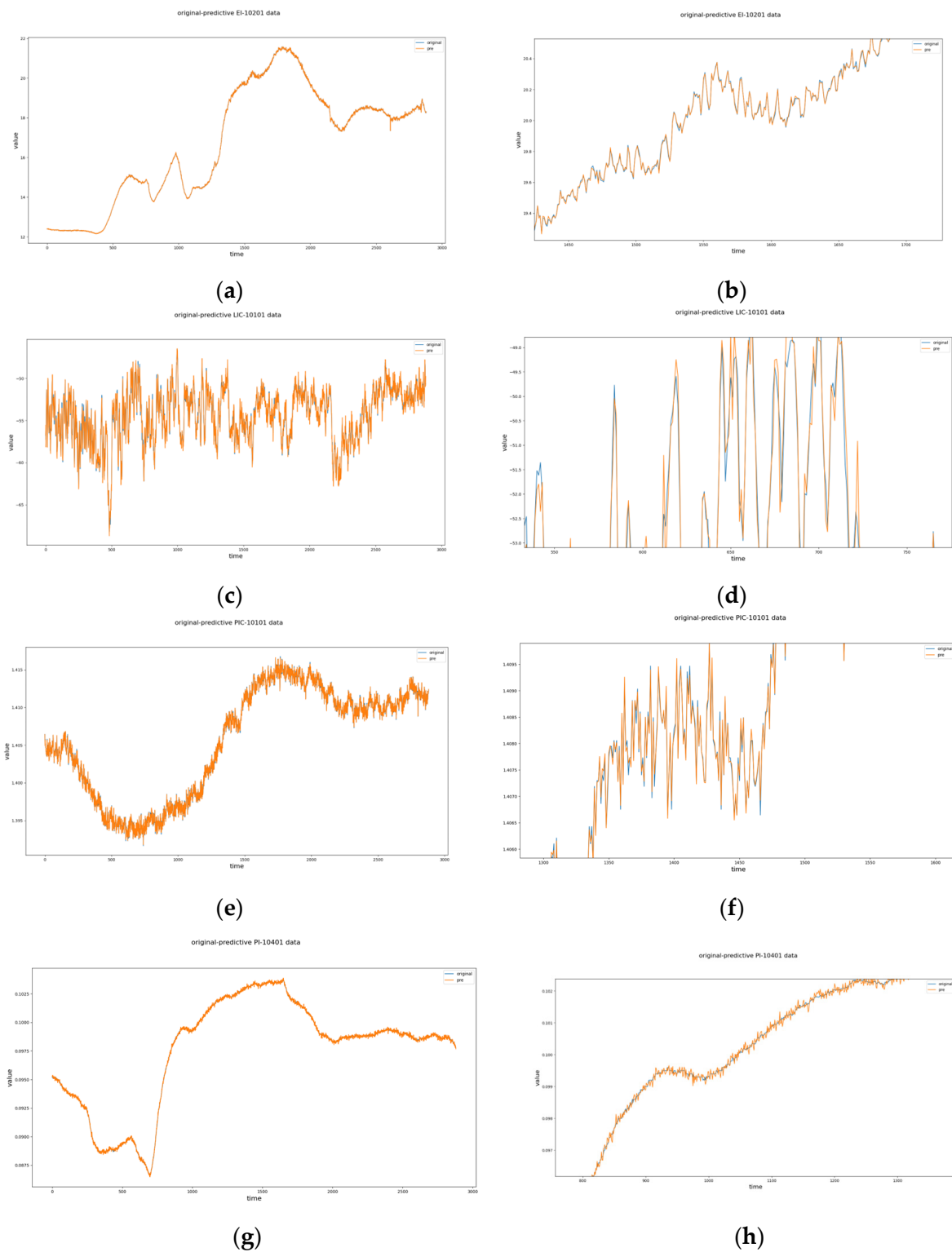
The prediction results of four variables are given in Figure 13 to further highlight the efficacy of the proposed fused data-driven methodology. In Figure 14, the real data and the predicted data are compared and partially enlarged to be more intuitive and verify the results. The projected data are displayed in orange, whereas the actual data are displayed in blue. The degree of curve fitting indicates how well the expected results match the actual data and how reliable the model is.

**Table 7.** Comparison of the results of different settings of the Bi-LSTM-MLP model for atmospheric depressurization unit.

Method	Differences	RMSE
	Different activation functions	
BiLSTM + MLP	ReLU()	0.062546
BiLSTM + MLP	ELU()	0.098545
BiLSTM + MLP	Tanh()	0.105412
BiLSTM + MLP	SiLU()	0.112874
BiLSTM + MLP	LeakyReLU()	0.052546
	Bi-LSTM different layers	
BiLSTM + MLP	1	0.098541
BiLSTM + MLP	2	0.052546
BiLSTM + MLP	3	0.066541
	MLP different layers	
BiLSTM + MLP	1	0.084124
BiLSTM + MLP	2	0.052546
BiLSTM + MLP	3	0.062541

**Figure 13.** Predicted results. (a) Prediction results of EI10201. (b) Prediction results of LIC10101. (c) Prediction results of PIC10101. (d) Prediction results of PI10401.

Further proving the superiority of the proposed model, Table 8 compares the performance indices of the fused Bi-LSTM + MLP model with those of the KNN, SVM, LSTM, LSTM + MLP, and Bi-LSTM models using RMSE representation. According to Table 8, the developed method can more precisely and dependably monitor and timely warn of the abnormal working conditions of the chemical process. The model incorporating Bi-LSTM + MLP can effectively predict the process data in the chemical process and reduce the gap between the predicted data and the real data.



**Figure 14.** Predicted data and original-predictive results and partially enlarged view. (a) Results of comparing the original and predicted data for EI10201. (b) Partially enlarged view of (a). (c) Results of comparing the original and predicted data for LIC10101. (d) Partially enlarged view of (c). (e) Results of comparing the original and predicted data for PIC10101. (f) Partially enlarged view of (e). (g) Results of comparing the original and predicted data for PI10401. (h) Partially enlarged view of (g).

**Table 8.** Comparison of the results of various prediction models for atmospheric depressurization unit.

Method	RMSE
KNN regression model	0.111532
SVM regression model	0.109854
LSTM	0.094562
LSTM + MLP	0.073568
Bi-LSTM	0.086985
Bi-LSTM + MLP	0.052546

#### 4. Conclusions

The present study introduces a novel data-driven early warning approach to effectively monitor abnormal working conditions in chemical processes. The proposed method, which combines the strengths of k-means, DBSCAN, Bi-LSTM, and MLP algorithms, offers improved accuracy in early warning. Specifically, the method leverages the k-means and DBSCAN algorithms to extract abnormal data for input into the Bi-LSTM-MLP model, which better integrates the temporal features of various states. The proposed method is evaluated using both the TE model and a real chemical plant example. Results indicate that the method yields high accuracy and presents a viable solution for detecting abnormal states in chemical processes, affording monitoring and maintenance personnel more flexible time to take appropriate actions and improve the safety and reliability of the process. However, different data preprocessing methods can be considered for linear and nonlinear processes, and further study could look into the creation of modules to help operators decide what to do in unusual circumstances.

**Author Contributions:** Conceptualization, writing—original draft preparation, methodology, X.S.; data curation, investigation, resources, F.Y.; supervision, D.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the the Major Scientific and Technological Innovation Project of Shandong Province (2019JZZY020502).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

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

#### References

1. Yuan, J.; Wang, S.; Wang, F.; Zhang, S. Abnormal condition identification via OVR-IRBF-NN for the process industry with imprecise data and semantic information. *Ind. Eng. Chem. Res.* **2020**, *59*, 5072–5086. [[CrossRef](#)]
2. Tian, W.; Liu, Z.; Li, L.; Zhang, S.; Li, C. Identification of abnormal conditions in high-dimensional chemical processes based on feature selection and deep learning. *Chin. J. Chem. Eng.* **2020**, *28*, 1875–1883. [[CrossRef](#)]
3. Goode, K.B.; Moore, J.; Roylance, B.J. Plant machinery working life prediction method utilizing reliability and condition-monitoring data. *Proc. Inst. Mech. Eng. Part E J. Process Mech. Eng.* **2000**, *214*, 109–122. [[CrossRef](#)]
4. Yang, F.; Dai, C.; Tang, J.; Xuan, J.; Cao, J. A hybrid deep learning and mechanistic kinetics model for the prediction of fluid catalytic cracking performance. *Chem. Eng. Res. Des.* **2020**, *155*, 202–210. [[CrossRef](#)]
5. Vitkus, D.; Steckevičius, Ž.; Goranin, N.; Kalibastienė, D.; Čenys, A. Automated expert system knowledge base development method for information security risk analysis. *Int. J. Comput. Commun. Control* **2019**, *14*, 743–758. [[CrossRef](#)]
6. Yin, S.; Ding, S.X.; Xie, X.; Luo, H. A review on basic data-driven approaches for industrial process monitoring. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6418–6428. [[CrossRef](#)]
7. Dai, Y.; Wang, H.; Khan, F.; Zhao, J. Abnormal situation management for smart chemical process operation. *Curr. Opin. Chem. Eng.* **2016**, *14*, 49–55. [[CrossRef](#)]
8. Kruger, U.; Chen, Q.; Sandoz, D.J.; McFarlane, R.C. Extended PLS approach for enhanced condition monitoring of industrial processes. *AIChE J.* **2001**, *47*, 2076–2091. [[CrossRef](#)]
9. Ge, Z.; Xie, L.; Kruger, U.; Song, Z. Local ICA for multivariate statistical fault diagnosis in systems with unknown signal and error distributions. *AIChE J.* **2012**, *58*, 2357–2372. [[CrossRef](#)]
10. Choi, S.W.; Park, J.H.; Lee, I.B. Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis. *Comput. Chem. Eng.* **2004**, *28*, 1377–1387. [[CrossRef](#)]

11. Alloghani, M.; Al-Jumeily, D.; Mustafina, J.; Hussain, A.; Aljaaf, A.J. A systematic review on supervised and unsupervised machine learning algorithms for data science. In *Supervised and Unsupervised Learning for Data Science*; Springer: Cham, Switzerland, 2020; pp. 3–21.
12. Zhou, Y.; Hahn, J.; Mannan, M.S. Process monitoring based on classification tree and discriminant analysis. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 546–555. [[CrossRef](#)]
13. Ruiz, D.; Nougués, J.; Calderon, Z.; Espuna, A.; Puigjaner, L. Neural network based framework for fault diagnosis in batch chemical plants. *Comput. Chem. Eng.* **2000**, *24*, 777–784. [[CrossRef](#)]
14. Wang, Y.; Pan, Z.; Yuan, X.; Yang, C.; Gui, W. A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Trans.* **2020**, *96*, 457–467. [[CrossRef](#)]
15. Arunthavanathan, R.; Khan, F.; Ahmed, S.; Imtiaz, S. Autonomous fault diagnosis and root cause analysis for the processing system using one-class SVM and NN permutation algorithm. *Ind. Eng. Chem. Res.* **2022**, *61*, 1408–1422. [[CrossRef](#)]
16. Natarajan, S.; Srinivasan, R. Implementation of multi agents based system for process supervision in large-scale chemical plants. *Comput. Chem. Eng.* **2014**, *60*, 182–196. [[CrossRef](#)]
17. Abd Majid, N.A.; Young, B.R.; Taylor, M.P.; Chen, J.J. K-means clustering pre-analysis for fault diagnosis in an aluminium smelting process. In Proceedings of the 2012 4th Conference on Data Mining and Optimization (DMO), Langkawi, Malaysia, 2–4 September 2012; pp. 43–46.
18. Bi, X.; Zhao, J. A novel orthogonal self-attentive variational autoencoder method for interpretable chemical process fault detection and identification. *Process Saf. Environ. Prot.* **2021**, *156*, 581–597. [[CrossRef](#)]
19. Gharahbagheri, H.; Imtiaz, S.A.; Khan, F.I. Application of Bayesian network for root cause diagnosis of chemical process fault. In Proceedings of the 2017 Indian Control Conference (ICC), Guwahati, India, 4–6 January 2017; pp. 188–193.
20. Don, M.G.; Khan, F. Dynamic process fault detection and diagnosis based on a combined approach of hidden Markov and Bayesian network model. *Chem. Eng. Sci.* **2019**, *201*, 82–96.
21. Khediri, I.B.; Weihs, C.; Limam, M. Kernel k-means clustering based local support vector domain description fault detection of multimodal processes. *Expert Syst. Appl.* **2012**, *39*, 2166–2171. [[CrossRef](#)]
22. Tong, C.; Palazoglu, A.; Yan, X. An adaptive multimode process monitoring strategy based on mode clustering and mode unfolding. *J. Process Control* **2013**, *23*, 1497–1507. [[CrossRef](#)]
23. Shi, W.; Zeng, W. Application of k-means clustering to environmental risk zoning of the chemical industrial area. *Front. Environ. Sci. Eng.* **2014**, *8*, 117–127. [[CrossRef](#)]
24. Zhang, M.-M. The research and application of process industry alarm management system. *Autom. Petro-Chem. Ind.* **2015**, *51*.
25. Yu, J.; Jang, J.; Yoo, J.; Park, J.H.; Kim, S. A clustering-based fault detection method for steam boiler tube in thermal power plant. *J. Electr. Eng. Technol.* **2016**, *11*, 848–859. [[CrossRef](#)]
26. Jie, Y.-P.; Zhao, P.; Dang, W.M. Fault Diagnosis for Batch Fermentation Process Based on KMC-KECA. *Autom. Petro-Chem. Ind.* **2016**, *52*, 21–26.
27. Cai, S.; Zhang, L.; Palazoglu, A.; Hu, J. Clustering analysis of process alarms using word embedding. *J. Process Control* **2019**, *83*, 11–19. [[CrossRef](#)]
28. Zhang, G.; Yao, J.; Wu, P. Design and Application of Intelligent Diagnosis Model Based on Data Driven. In Proceedings of the 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS), Liuzhou, China, 20–22 November 2020; pp. 87–92.
29. Choi, Y.; Bhadriaju, B.; Cho, H.; Lim, J.; Han, I.S.; Moon, I.; Kwon, J.S.-I.; Kim, J. Data-driven modeling of multimode chemical process: Validation with a real-world distillation column. *Chem. Eng. J.* **2023**, *457*, 141025. [[CrossRef](#)]
30. Jiang, H.; Wang, R.; Gao, J.; Gao, Z.; Gao, X. Evidence fusion-based framework for condition evaluation of complex electromechanical system in process industry. *Knowl. Based Syst.* **2017**, *124*, 176–187. [[CrossRef](#)]
31. Khodabakhsh, A.; Ari, I.; Bakir, M.; Ercan, A.O. Multivariate sensor data analysis for oil refineries and multi-mode identification of system behavior in real-time. *IEEE Access* **2018**, *6*, 64389–64405. [[CrossRef](#)]
32. Garg, S.; Kaur, K.; Batra, S.; Kaddoum, G.; Kumar, N.; Boukerche, A. A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications. *Future Gener. Comput. Syst.* **2020**, *104*, 105–118. [[CrossRef](#)]
33. Sompura, J.; Joshi, A.; Srinivasan, B.; Srinivasan, R. A practical approach to improve alarm system performance: Application to power plant. *Chin. J. Chem. Eng.* **2019**, *27*, 1094–1102. [[CrossRef](#)]
34. Zhu, A.; Zhao, Q.; Yang, T.; Zhou, L.; Zeng, B. Condition monitoring of wind turbine based on deep learning networks and kernel principal component analysis. *Comput. Electr. Eng.* **2023**, *105*, 108538. [[CrossRef](#)]
35. Dai, Y.; Chen, N.; Zhao, J.; Chen, B. Application of AIS to batch chemical process fault diagnosis. *CIESC J.* **2009**, *60*, 172–176.
36. Zhao, J.; Shu, Y.; Zhu, J.; Dai, Y. An online fault diagnosis strategy for full operating cycles of chemical processes. *Ind. Eng. Chem. Res.* **2014**, *53*, 5015–5027.
37. Li, C.; Zhao, D.; Mu, S.; Zhang, W.; Shi, N.; Li, L. Fault diagnosis for distillation process based on CNN-DAE. *Chin. J. Chem. Eng.* **2019**, *27*, 598–604. [[CrossRef](#)]
38. Arunthavanathan, R.; Khan, F.; Ahmed, S.; Imtiaz, S.; Rusli, R. Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique. *Comput. Chem. Eng.* **2020**, *134*, 106–697. [[CrossRef](#)]
39. Zhang, X.; Zou, Y.; Li, S.; Xu, S. A weighted auto regressive LSTM based approach for chemical processes modeling. *Neurocomputing* **2019**, *367*, 64–74. [[CrossRef](#)]

40. Park, P.; Marco, P.D.; Shin, H.; Bang, J. Fault detection and diagnosis using combined autoencoder and long short-term memory network. *Sensors* **2019**, *19*, 4612.
41. Han, Y.; Ding, N.; Geng, Z.; Wang, Z.; Chu, C. An optimized long short-term memory network based fault diagnosis model for chemical processes. *J. Process Control* **2020**, *92*, 161–168. [[CrossRef](#)]
42. Ren, J.; Ni, D. A batch-wise LSTM-encoder decoder network for batch process monitoring. *Chem. Eng. Res. Des.* **2020**, *164*, 102–112. [[CrossRef](#)]
43. Zheng, S.; Zhao, J. A new unsupervised data mining method based on the stacked autoencoder for chemical process fault diagnosis. *Comput. Chem. Eng.* **2020**, *135*, 106755.
44. Bai, Y.; Xiang, S.; Cheng, F.; Zhao, J. A dynamic-inner LSTM prediction method for key alarm variables forecasting in chemical process. *Chin. J. Chem. Eng.* **2023**, *55*, 266–276. [[CrossRef](#)]
45. Zhang, S.; Qiu, T. Semi-supervised LSTM ladder autoencoder for chemical process fault diagnosis and localization. *Chem. Eng. Sci.* **2022**, *251*, 117467. [[CrossRef](#)]
46. Ghazal, T.M. Performances of K-means clustering algorithm with different distance metrics. *Intell. Autom. Soft Comput.* **2021**, *30*, 735–742. [[CrossRef](#)]
47. Froese, R.; Klassen, J.W.; Leung, C.K.; Loewen, T.S. The border k-means clustering algorithm for one dimensional data. In Proceedings of the 2022 IEEE International Conference on Big Data and Smart Computing (BigComp), Daegu, Republic of Korea, 17–20 January 2022; pp. 35–42.
48. Duggimpudi, M.B.; Abbady, S.; Chen, J.; Raghavan, V.V. Spatio-temporal outlier detection algorithms based on computing behavioral outlierness factor. *Data Knowl. Eng.* **2019**, *122*, 1–24. [[CrossRef](#)]
49. Güngör, E.; Özmen, A. Distance and density based clustering algorithm using Gaussian kernel. *Expert Syst. Appl.* **2017**, *69*, 10–20. [[CrossRef](#)]
50. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
51. Lazzara, M.; Chevalier, M.; Colombo, M.; Garcia, J.G.; Lapeyre, C.; Teste, O. Surrogate modelling for an aircraft dynamic landing loads simulation using an LSTM AutoEncoder-based dimensionality reduction approach. *Aerosp. Sci. Technol.* **2022**, *126*, 107629. [[CrossRef](#)]
52. Kaur, D.; Kumar, R.; Kumar, N.; Guizani, M. Smart grid energy management using rnn-lstm: A deep learning-based approach. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
53. Ma, C.; Dai, G.; Zhou, J. Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM\_BILSTM method. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 5615–5624. [[CrossRef](#)]
54. Weerakody, P.B.; Wong, K.W.; Wang, G.; Ela, W. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing* **2021**, *441*, 161–178. [[CrossRef](#)]
55. Sun, J.; Zhang, X.; Wang, J. Lightweight bidirectional long short-term memory based on automated model pruning with application to bearing remaining useful life prediction. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105662.
56. Bisong, E.; Bisong, E. *The Multilayer Perceptron (MLP). Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*; Apress: New York, NY, USA, 2019; pp. 401–405.

**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.