Proceedings of the 6th European Conference on Industrial Engineering and Operations Management
Lisbon, Portugal, July 18-20, 2023

An Improved Autoencoder with Dynamic Hidden Layer for Anomaly Detection Monitoring

Haitian Zhang
PhD Candidate in the Department of Chemical Engineering
University of Waterloo
Waterloo, Canada

Qinqin Zhu
Assistant Professor in the Department of Chemical Engineering
University of Waterloo
Waterloo, Canada

Ali Ahmadian
Postdoctoral Research Fellow in the Department of Chemical Engineering
University of Waterloo
Waterloo, Canada

Ali Elkamel
Full Professor in the Department of Chemical Engineering
University of Waterloo
Waterloo, Canada
aelkamel@uwaterloo.ca

Abstract

Due to the rapid advancement of Industry 4.0, the increasing complexity of industrial applications leads to the expanding dimensionality of time series data. To maintain the performance, avoid economic losses, and ensure safety during the industrial processes, anomaly detection draws great attention. In view of advantages in dimensionality reduction and feature retention, autoencoder (AE) technology is widely applied for anomaly detection monitoring. In this work, considering both high dimensionality and dynamic relations between elements in the hidden layer, an improved autoencoder with dynamic hidden layer (DHL-AE) is proposed and applied for anomaly detection monitoring. Two case studies including Tennessee Eastman process and Wind data are used to show the effectiveness of the proposed algorithm. The results demonstrate that compared with classical AE approaches that are most commonly used, DHL-AE exhibits the best overall performance in anomaly detection monitoring.

Keywords
Autoencoder, Dynamic hidden layer, Anomaly detection, monitoring.

1. Introduction
With the accelerated development of Industry 4.0, considerable technological progress such as artificial intelligence and real-time signal processing methods has equipped existing industrial systems with highly sophisticated technologies in diverse fields, leading to significant enhancement on the complexity of real-world industrial applications, especially chemical industry. The collected data from complicated practical industry processes primarily consists of multivariate time series data with high dimensionality. To ensure the operational safety and security of complex industrial systems, anomaly detection has been a critical issue for identifying abnormal observations or events that deviate from normal patterns, which may cause potential problems in industrial practice (Zhou et al. 2020).

Several classical machine learning and deep learning methods have been widely adopted in industry and academia. More concretely, multivariate statistical approaches such as clustering (Xu and Wunsch 2005), principal component analysis (PCA) (Wold et al. 1987), and Gaussian mixture model (GMM) (Reynolds 2009) have been used to detect abnormal points that exhibit substantial deviations from the normal data. In terms of machine learning-based algorithms, support vector machines (SVM) (Hearst et al. 1998), random forests (Breiman 2001), and neural networks (Bishop 1994) are also suitable for anomaly detection by model training. Deep learning methods have also been employed for identifying abnormal events or observations that deviate significantly from the expected or typical behavior of the data (Pang et al. 2021). Due to the ability to capture complex patterns, feature extraction and dimensionality reduction, autoencoder (AE) technology has been considered as a popular choice for anomaly detection in chemical practice (Sakurada and Yairi 2014). However, they pay less attention to dynamic behaviors existing in industrial datasets. Thus, it still remains a challenging issue carrying out the reliable results of anomaly detection from high-dimensional data that changes over time.

For reasons of identifying anomalies in real-world data with high dimensional and time-dependent properties, some classical algorithms such PCA and AE, and their corresponding variants can be employed for dimensionality reduction and feature extraction. Specifically, to handle time-dependent samples, dynamic PCA has been created to extract dynamics in the original data and detect irregular observations or samples (Tsung 2000). To overcome the assumption of most AE algorithms that data points are considered as independent samples, considering the dynamic relations existing in the original collected data from industrial applications, some pre-processing approaches for time series data have been adopted for the construction of frameworks combined with AE methods, such as mutual information (Yin and Yan 2019) and dynamic thresholding mechanism (Tayeh et al. 2022). However, these proposed schemes focus on the pre-processing methods, without any consideration for the improvement of AE structure. Pre-processing increases the complexity of algorithms, rendering disadvantages involving high computational costs, prolonged running time, difficulty of implementation, and higher demand on data quality (Bishop and Nasrabadi 2006).

Compared with PCA, AE methods possess the strengths including handling missing data, learning hierarchical representations, new data generation, and data processing capabilities (Hinton and Salakhutdinov 2006). Thus, AE is adopted to build up the anomaly detection scheme proposed in this work. Inspired by autoregressive integrated moving average (ARIMA) model (Schaffer et al. 2021), the hidden layer of AE is improved to handle the issue of dynamics in our work. Then, a modified autoencoder method with dynamic hidden layer (DHL-AE) is proposed to address both high dimensionality and dynamic relations simultaneously and the corresponding anomaly detection scheme is designed to improve the performance of identifying abnormal points. The effectiveness and superiority of the proposed algorithm for anomaly detection monitoring can be demonstrated by case studies of Tennessee Eastman process (TEP) and Wind data.

The main contributions of this article are listed as follows:

- An improved autoencoder with dynamic hidden layer (DHL-AE) algorithm is developed to address high-dimensional and time-dependent issues existing in the real-world data simultaneously.
- An anomaly detection monitoring scheme based on DHL-AE is constructed and its validity and superiority are proved by case studies.

The remainder of this article is organized as follows. Section 2 briefly reviews some preliminaries. Section 3 presents the details of the proposed DHL-AE algorithm and the corresponding anomaly detection monitoring framework, respectively. The superiority of the proposed methods is analyzed with two case studies in Section 4. Finally, conclusions are drawn, and the direction of future work is discussed in the last section.

2. Preliminaries

2.1 Autoencoder

As a specific type of neural networks, AE has been created by encoding the input into a compressed representation and decoding it back to reconstruct the input that is nearly identical to the original input to greatest extent possible (Rumelhart et al. 1985).

The purpose of AE is to learn a lower-dimensional and meaningful representation of the input in an unsupervised way via minimizing the reconstruction error based on a loss function in training (Hinton and Salakhutdinov 2006). Typically, the mean squared error (MSE) between the original input and the corresponding reconstructed output is adopted to be the loss function. Assuming $\mathbf{x} \in \mathbb{R}^n$ as the input, the loss function is expressed as

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \tag{1}$$

where L expresses the loss function and is the corresponding reconstructed output vector of $\mathbf{x}$.
The encoder projects the input into the hidden layer to obtain a compressed representation $\mathbf{z} \in \mathbb{R}^k$, and the decoder reconstructs the output of AE based on $\mathbf{z}$ as $\hat{\mathbf{x}}$.
To construct the encoder by the projection $\varphi: \chi \to F$ and decoder by the projection $\psi: F \to \chi$, the objective of AE is defined as (Goodfellow et al. 2016)

$$\underset{\varphi,\psi}{\operatorname{argmin}} \|\mathbf{x} - \psi[\varphi(\mathbf{x})]\|^2 \tag{2}$$

where $\varphi(\mathbf{x})$ denotes the process of encoding, and $\mathbf{z} = \varphi(\mathbf{x})$. $\psi[\varphi(\mathbf{x})]$ represents the process of decoding, and $\hat{\mathbf{x}} = \psi[\varphi(\mathbf{x})]$.
More concretely, encoding and decoding can be expressed as

$$\begin{aligned} \mathbf{z} &= f_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \\ \hat{\mathbf{x}} &= f_2(\mathbf{W}_2\mathbf{x} + \mathbf{b}_2) \end{aligned} \tag{3}$$

where f_1 and f_2 represent the activation functions used to construct the hidden and output layers, respectively. $\mathbf{W}_1$ and $\mathbf{b}_1$ are the weight and bias of encoder, while $\mathbf{W}_2$ and $\mathbf{b}_2$ are the weight and bias of decoder.

2.2 Autoregressive Integrated Moving Average Model

Autoregressive Integrated Moving Average (ARIMA) models have been commonly utilized in time series analysis. Due to its flexibility and applicability, ARIMA models have been developed and applied for processing data with trends, irregular fluctuations, or even seasonality. Typically, an ARIMA model consists of three parts, involving autoregressive part, integrated part, and moving average part. The autoregressive part captures the temporal dependence of current value based on its past values, while the differencing process depends on the integrated part. Additionally, the error term is processed as a linear combination of past error terms by the moving average part (Schaffer et al. 2021). Given an input vector $\mathbf{v} \in \mathbb{R}^n$, which can also be expressed as time series, the autoregressive part at time t (t = 1, 2, ..., n) is expressed by

$$\mathbf{v}_t = \sum_{i=1}^{p} \beta_i \mathbf{v}_{t-i} + \boldsymbol{\varepsilon}_t \tag{4}$$

where $\mathbf{v}_t$ denotes the aggressive part $\mathbf{v}$ existing in the input $\mathbf{z}$ at time t. p is the number of time lags, and β_i (i = 1, 2, ..., p) are parameters of the autoregressive part. $\boldsymbol{\varepsilon}_t$ denotes the white noise error term at time t, which is typically assumed to be independent and conform to a normal distribution with zero mean.
The moving average part is given by

$$\mathbf{e}_t = \boldsymbol{\mu} + \sum_{i=1}^{q} \theta_i \boldsymbol{\varepsilon}_{t-i} + \boldsymbol{\varepsilon}_t \tag{5}$$

where $\mathbf{e}_t$ denotes the moving average part $\mathbf{e}$ existing in the input $\mathbf{z}$ at time t. $\boldsymbol{\mu}$ represents the mean of $\mathbf{e}$. q is the order of the moving average part. θ_i (i = 1, 2, ..., q) are parameters of the moving average part. $\boldsymbol{\varepsilon}_t$ is the white noise error term at time t (t = 1, 2, ..., n) for the moving average part as well.
Combining Eqs. (4) and (5), the ARIMA model is defined by

$$\mathbf{z}_t = \left(1 - \sum_{i=1}^{p} \beta_i B^i\right)^{-1} (1 - B)^{-d} \left(1 - \sum_{i=1}^{q} \theta_i B^i\right) \boldsymbol{\varepsilon}_t \tag{6}$$

where $\mathbf{z}_t$ represents the component of time series $\mathbf{z}$ at time t. B denotes the backshift operator, which operates on an element of the compressed representation to construct the previous element and can be expressed as $B\mathbf{z}_t = \mathbf{z}_{t-1}, \forall t > 1$. d denotes the degree of differencing, which is the number of times the time series $\mathbf{z}$ has that had past values subtracted. Here, p, d, and q are non-negative integers.

2.3 Anomaly Detection

Anomaly detection refers to the identification of anomalies, which are defined as abnormal events, observations, or samples that significantly deviate from the normal or expected behavior of given data to indicate the potential problems in the dataset (Aggarwal and Aggarwal 2017). The early detection of anomalies plays an essential role in industrial practice to raise the system safety, prevent failures, reduce downtime, and improve production in manufacturing, since anomalies are considered as indicative of potential issues involving safety risks, equipment failures, and maintenance

problems (Chalapathy and Chawla 2019). To detect anomalies in a given dataset, the model that captures the normal observations or behaviors is constructed by the training period, and the test data is applied for the identification of abnormal points which are not fitted with the training model.

Anomaly detection using dimensionality reduction has been developed based on the assumption that variables in the input data are correlated with each other and their features can be captured by projecting them into a lower dimensional subspace where the samples are significantly different from the original ones (Sakurada and Yairi 2014). In the training period, the input matrix $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\right] \in \mathbb{R}^{n \times m}$ with normal samples is considered as the training set, where *n* and *m* denote the number of samples and variables in the original data. Then, the corresponding compressed representation matrix $\mathbf{Z} = \left[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(n)}\right] \in \mathbb{R}^{n \times k}$ is reconstructed as $\hat{\mathbf{Z}} = \left[\hat{\mathbf{z}}^{(1)}, \hat{\mathbf{z}}^{(2)}, \ldots, \hat{\mathbf{z}}^{(n)}\right] \in \mathbb{R}^{n \times k}$ to capture the features of $\mathbf{X}$, where *k* is the dimension of the compressed representation. The output of the model is reproduced by decoder as $\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \ldots, \hat{\mathbf{x}}^{(n)}\right] \in \mathbb{R}^{n \times m'}$ $(m' \leq m)$, and MSE is used as the reconstruction error (RE) between input and output is calculated as

$$\boldsymbol{\varepsilon}^{(t)} = \left\| \mathbf{x}^{(t)} - \hat{\mathbf{x}}^{(t)} \right\|^2 \tag{7}$$

where $\boldsymbol{\varepsilon}^{(t)}$ denotes the RE of an input vector $\mathbf{x}^{(t)}$ at time *t* (*t* = 1, 2, ..., *n*). RE is obligated to update the parameters of encoder and decoder and calculate the anomaly score for anomaly detection.

3. Anomaly Detection Framework Based on Autoencoder with Dynamic Hidden Layer

3.1 DHL-AE Algorithm

Based on the structure of AE, DHL-AE algorithm is constructed by processing dynamics in the hidden layer. Premised on the assumption of normalized input, the encoder maps the original input into the latent space in hidden layer, and the dynamics in the original data are captured by the compressed representation. After decoding, dynamic relations are also retained in the output, which is considered as the reconstructed input. Thus, to deal with dynamic datasets, it is reasonable to handle dynamic relations in the hidden layer.

For the dynamic input data $\mathbf{X}$, after normalization, the encoder projects $\mathbf{X}$ into the latent space in the hidden layer, obtaining $\mathbf{Z} = f_1(\mathbf{W}_1\mathbf{X} + \mathbf{b}_1)$. Considering dynamic relations existing in the input $\mathbf{X}$, motivated by ARIMA, the components of compressed representation in the hidden layer can be constructed as follows (Hyndman and Athanasopoulos 2018):

$$\begin{aligned} \hat{\mathbf{z}}^{(t)} &= \mathbf{v}^{(t)} + \mathbf{e}^{(t)} \\ \mathbf{v}^{(t)} &= \sum_{i=1}^{p} \beta_i \mathbf{z}^{(t-i)} \\ \mathbf{e}^{(t)} &= \sum_{i=1}^{q} \theta_i \boldsymbol{\varepsilon}^{(t-i)} + \boldsymbol{\varepsilon}^{(t)} \end{aligned} \tag{8}$$

where $\hat{\mathbf{z}}^{(t)} \in \mathbb{R}^k$ represents the compressed representation reconstructed considering dynamic relations in $\mathbf{z}^{(t)}$ obtained by encoding at time t, which consists of the autoregressive part $\mathbf{v}^{(t)}$ and the moving part $\mathbf{e}^{(t)}$.
Similar to Eq. (6), Eq. (8) can also be reorganized as

$$\hat{\mathbf{z}}^{(t)} = \left(1 - \sum_{i=1}^{p} \beta_i B^i\right)^{-1} (1 - B)^{-d} \left(1 - \sum_{i=1}^{q} \theta_i B^i\right) \boldsymbol{\varepsilon}^{(t)} \tag{9}$$

where $\boldsymbol{\varepsilon}^{(t)}$ is the white noise error at time *t* (*t* = 1, 2, ..., *n*) for the moving average part.
After the process in the hidden layer, the decoder reconstructs the output from $\hat{\mathbf{z}}$ via $\hat{\mathbf{x}} = f_2\left(\mathbf{W}_2\hat{\mathbf{z}} + \mathbf{b}_2\right)$.

The detailed DHL-AE algorithm is summarized as follows:

- Scale the input data $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\right]$ into zero mean and unit variance.
- The normalized input data is fed to DHL-AE.
- After encoding via $\mathbf{Z} = f_1(\mathbf{W}_1\mathbf{X} + \mathbf{b}_1)$, the input $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}$ can be projected to the latent space in the hidden layer as the compressed representation $\mathbf{Z} = \left[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(n)}\right]$.

- Decompose $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(n)}$ into two parts, involving the autoregressive part $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(n)}$ and the moving average part $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \ldots, \mathbf{e}^{(n)}$, leading to the reconstructed compressed representation $\hat{\mathbf{z}}^{(t)} = \mathbf{v}^{(t)} + \mathbf{e}^{(t)}$ ($t = 1, 2, \ldots, n$).
- The decoder reconstructs the input as the output of DHL-AE by $\hat{\mathbf{X}} = f_2(\mathbf{W}_2\hat{\mathbf{Z}} + \mathbf{b}_2)$, where $\hat{\mathbf{Z}} = \left[\hat{\mathbf{z}}^{(1)}, \hat{\mathbf{z}}^{(2)}, \ldots, \hat{\mathbf{z}}^{(n)}\right]$.

The structure of DHL-AE algorithm is visualized as Figure 1.

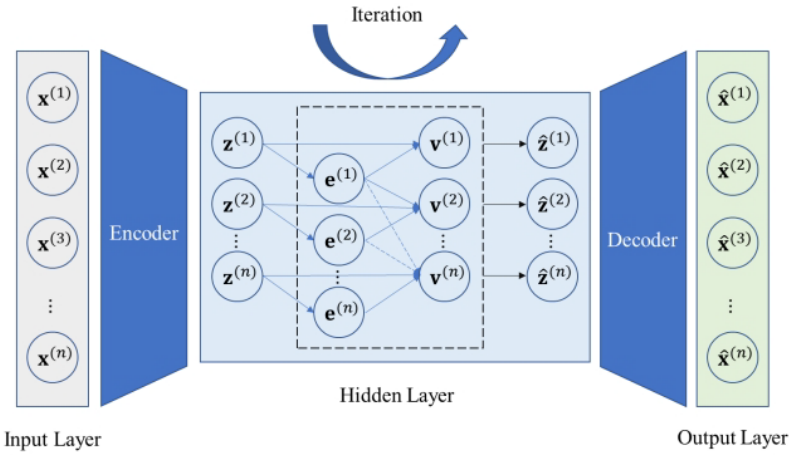


Figure 1. Structure of autoencoder with dynamic hidden layer

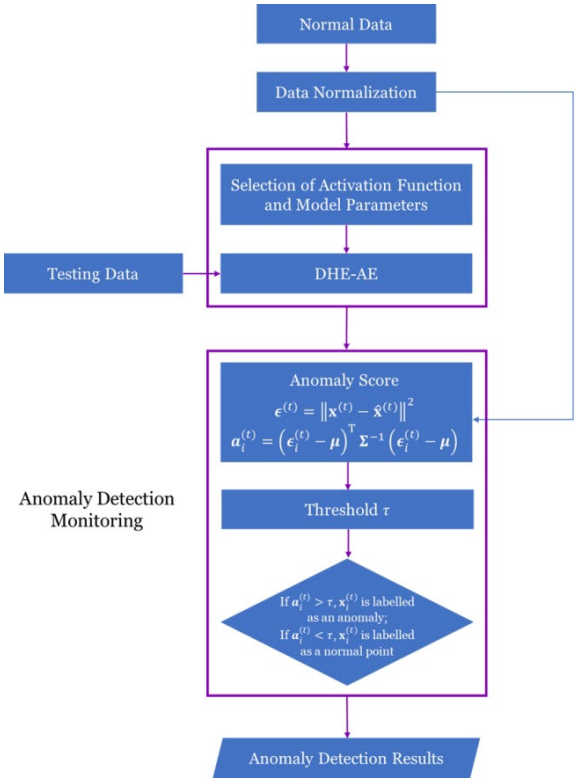


Figure 2. Anomaly detection monitoring scheme based on DHL-AE

3.2 Anomaly Detection Scheme Based on DHL-AE

To enhance the performance, improve the safety, and decrease the potential losses of the process, in this article, DHL-AE is integrated with anomaly detection to identify anomalies in the process monitoring.

Anomaly detection is based on the conversion from reconstruction errors to anomaly scores. The anomaly score for a datapoint $\mathbf{x}_i^{(t)}$ is computed as (Ahmad et al. 2020)

$$\mathbf{a}_i^{(t)} = \left(\boldsymbol{\varepsilon}_i^{(t)} - \boldsymbol{\mu}\right)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\varepsilon}_i^{(t)} - \boldsymbol{\mu}\right) \tag{10}$$

where $\mathbf{a}_i^{(t)}$ is the anomaly score for $\mathbf{x}_i^{(t)}$. $\boldsymbol{\varepsilon}_i^{(t)}$ represents the corresponding RE. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are obtained from a normal distribution using maximum likelihood estimation based on $\boldsymbol{\varepsilon}_i^{(t)}$ (Malhotra et al. 2016).

To determine whether a sample is an anomaly or not, the threshold of anomaly scores, which is denoted by τ, is learned using the normal data (Ahmad et al. 2020). If $\mathbf{a}_i^{(t)} > \tau$, the corresponding sample $\mathbf{x}_i^{(t)}$ is labelled as an anomaly, while if $\mathbf{a}_i^{(t)} < \tau$, $\mathbf{x}_i^{(t)}$ is considered as a normal point.

For better illustration, the integrated scheme is depicted in Figure 2. In the scheme, DHE-AE is first performed on the normal data after normalization to obtain the proper activation function and model parameters. Then samples of testing data are processed by anomaly detection monitoring to identify irregular datapoints.

4. Tennessee Eastman Process Case Study

In this section, the Tennessee Eastman Process (TEP) is applied for case study to demonstrate the effectiveness of the proposed anomaly detection monitoring framework, with 15 disturbances (IDV (1–15)) simulated in the dataset (Downs and Vogel 1993). XMEAS (1-9) are chosen as input variables, and 500 normal datapoints in d00 are selected as training data. Faulty data d02_te is utilized for testing, which represents the disturbance of IDV (2) and has 960 samples in total.

The activation function and model parameters are determined by the training data. The first three-quarters of the training data are designated as the training set, while the complement of the training set is assigned as the test set. Mean squared error (MSE), root mean squared error (RMSE), normalized root mean squared error (NRMSE), and mean absolute error (MAE) (Müller and Guido 2016) are considered as the candidates of model selection criteria to determine the best activation function and model parameters involving the dimension of output and compressed representation of DHL-AE. The activation function is selected from rectified linear unit (ReLU) (Arora et al. 2016), leaky rectified linear unit (Leaky ReLU) (Maas et al. 2013), and exponential linear unit (ELU) (Clevert et al. 2015). As mentioned in Section II, m' denotes the dimension of output $\hat{\mathbf{X}}$ after decoding, and k refers to the dimension of the compressed representation of input after encoding.

The results of activation function and parameter selection for modeling are shown in Table 1. In terms of activation function, Leaky ReLU and ELU perform better than ReLU, since they have lower MSE, RMSE, and NRMSE for all the metrics which are used for activation function and parameter selection. Considering the selection criteria and process time simultaneously, the combination of ELU as the activation function and RMSE as the section criteria is the best choice due to the smallest values of metrics including MSE, RMSE, NRMSE, and MAE, as well as the shortest process time. In this case, the parameters are determined as follows: the dimension of the output $m' = 8$, and the dimension of the compressed representation $k = 5$. Thus, it is concluded that for TEP, ELU is selected as the activation function, and the model parameters are selected as $m' = 8$ and $k = 5$.

Table 1. Activation function and parameter selection for IDV (2) in TEP

Activation Function	Selection Criteria	*m'*	*k*	MSE	RMSE	NRMSE	MAE	Time/s
ReLU	MSE	6	13	0.2537	0.5037	1.3305	0.3145	3288.84
	RMSE	6	13	0.2537	0.5037	1.3305	0.3145	3535.73
	NRMSE	6	13	0.2537	0.5037	1.3305	0.3145	4398.42
	MAE	7	1	0.2178	0.4667	1.2327	0.3283	3254.39
Leaky ReLU	MSE	8	5	**0.2034**[a]	**0.4510**[a]	**1.1914**[a]	**0.3141**[a]	3716.17

	RMSE	8	5	**0.2034**[a]	**0.4510**[a]	**1.1914**[a]	**0.3141**[a]	3309.23
	NRMSE	8	5	**0.2034**[a]	**0.4510**[a]	**1.1914**[a]	**0.3141**[a]	3925.47
	MAE	8	5	**0.2034**[a]	**0.4510**[a]	**1.1914**[a]	**0.3141**[a]	3108.95
ELU	MSE	8	5	**0.2034**[a]	**0.4510**[a]	**1.1914**[a]	**0.3141**[a]	3298.11
	RMSE	8	5	**0.2034**[a]	**0.4510**[a]	**1.1914**[a]	**0.3141**[a]	**3006.76**[a]
	NRMSE	8	5	**0.2034**[a]	**0.4510**[a]	**1.1914**[a]	**0.3141**[a]	3122.58
	MAE	7	1	0.2178	0.4667	1.2327	0.3283	3603.27

a. The values of metrics and process time with best performance are displayed in bold font.

In this case study, the proposed anomaly detection monitoring framework based on DHL-AE is applied for the detection of the anomalies in d02_te. Taken as the ground truth, PCA-based anomaly detection monitoring is assigned as the criterion to classify a sample into normal or abnormal datapoints. To manifest the superiority and effectiveness of the DHL-AE based anomaly detection scheme, other AE methods involving convolutional autoencoder (CAE), denoising autoencoder (DAE), sparse autoencoder (Sparse AE), 2-layer stacked autoencoder (2-Layer Stacked AE), 4-layer stacked autoencoder (4-Layer Stacked AE), and variational autoencoder (VAE).

Figure 3 illustrates the anomaly detection monitoring results of DHL-AE, PCA, and other AE methods for comparison. For better understanding, the following metrics are introduced to demonstrate the comparison results statistically: accuracy, precision, recall, F1-score (Powers 2020), false discovery rate (FDR), false alarm rate (FAR) (Zhang and Zhu 2022), and missing alarm rate (MAR) (Izadi et al. 2009), and the corresponding results are presented in Table 2. Among the performance of the aforementioned algorithms, in terms of accuracy, precision, recall, and F1-score, the proposed algorithm performs best, and it also has the best performance on fault alarm rate (FAR) due to its lowest value. Although the performance of DHL-AE on fault detection rate (FDR) and missing alarm rate (MAR) is not the best, it is second only to CAE, DAE, and 4-Layer Stacked AE, and the difference is quite small. Thus, it is concluded that DHL-AE takes advantages over other AE methods in this case study.

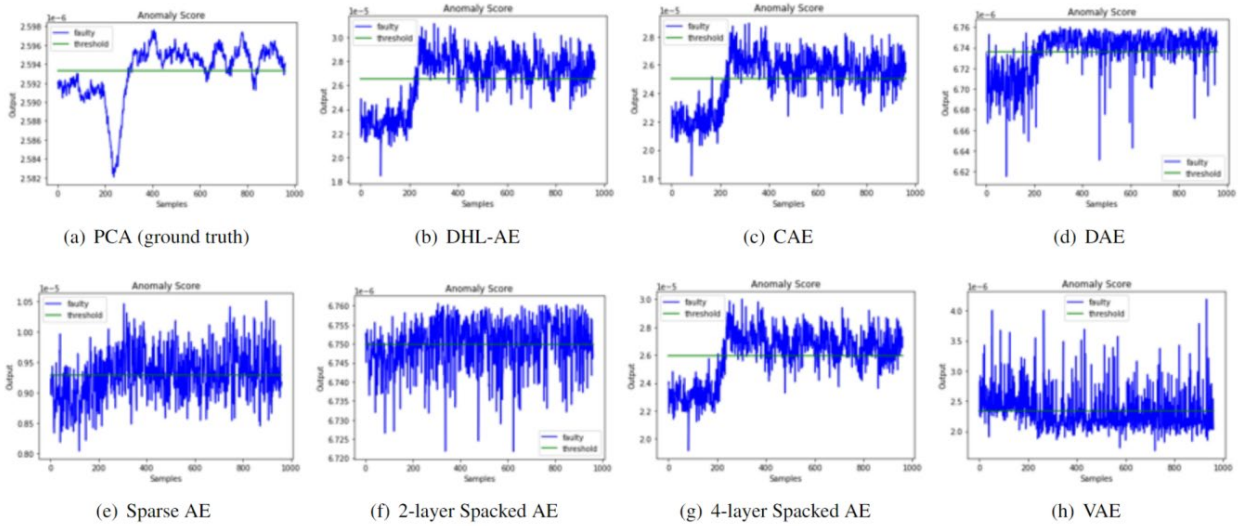


Figure 3. Anomaly detection monitoring results of algorithms for comparison for IDV (2) in TEP

Table 2. Statistical metrics results of algorithms for comparison for IDV (2) in TEP

Algorithms	Accuracy	Precision	Recall	F1-Score	FDR	FAR	MAR
DHL-AE	**0.7333**[a]	**0.7906**[a]	**0.7828**[a]	**0.7867**[a]	0.6499	**0.2172**[a]	0.3501
CAE	0.7271	0.7895	0.7711	0.7802	**0.6527**[a]	0.2289	**0.3473**[a]

DAE	0.7281	0.7898	0.7728	0.7812	**0.6527**[a]	0.2272	**0.3473**[a]
Sparse AE	0.5854[a]	0.7243	0.5489	0.7457	0.6471	0.4511	0.3529
2-Layer Stacked AE	0.6000	0.7102	0.6136	0.6584	0.5770[a]	0.3864	0.4230[a]
4-Layer Stacked AE	0.7104	0.7836	0.7446	0.6232	**0.6527**[a]	0.2554	**0.3473**[a]
VAE	0.5375	0.6755[a]	0.5075[a]	0.4415[a]	0.5882	0.4925[a]	0.4118

a. The values of metrics with best performance are highlighted in bold font, and the worst are displayed in red.

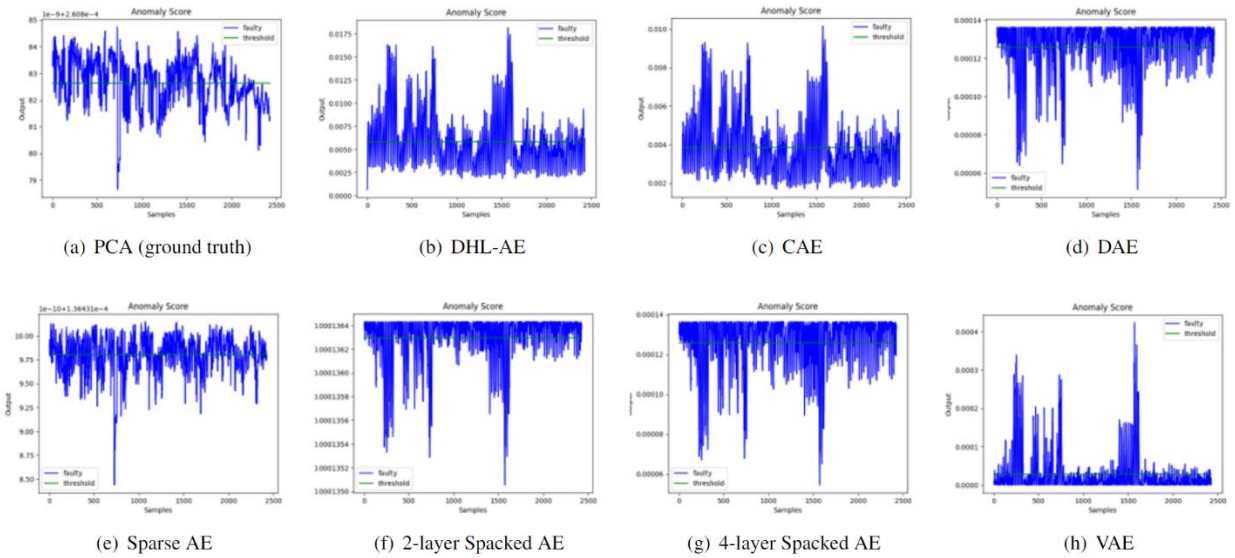
5. Wind 2017 Case Study

Table 3. Activation function and parameter selection for Wind 2017

Activation Function	Selection Criteria	*m'*	*k*	MSE	RMSE	NRMSE	MAE	Time/s
ReLU	MSE	1	9	0.1692	0.4113	1.3528	0.2308	4394.00
	RMSE	1	9	0.1692	0.4113	1.3528	0.2308	4394.00
	NRMSE	1	9	0.1692	0.4113	1.3528	0.2308	4394.00
	MAE	5	2	0.1159	0.3404	1.1198	0.1532	4642.00
Leaky ReLU	MSE	5	7	**0.1063**[a]	**0.3261**[a]	**1.0725**[a]	**0.1589**[a]	4465.00
	RMSE	5	7	**0.1063**[a]	**0.3261**[a]	**1.0725**[a]	**0.1589**[a]	**4523.00**[a]
	NRMSE	5	7	**0.1063**[a]	**0.3261**[a]	**1.0725**[a]	**0.1589**[a]	4539.00
	MAE	3	2	0.1229	0.3506	1.1531	0.1655	4419.00
ELU	MSE	3	2	0.2381	0.4880	1.6050	0.2471	4481.00
	RMSE	3	2	0.2381	0.4880	1.6050	0.2471	4492.00
	NRMSE	3	2	0.2381	0.4880	1.6050	0.2471	4503.00
	MAE	3	2	0.2381	0.4880	1.6050	0.2471	4546.00

a. The values of metrics and process time with best performance are displayed in bold font.

To illustrate the validity of the anomaly detection monitoring based on DHL-AE, a time-relevant dataset of only total wind supply with weather in 2017 (Alkabbani et al. 2021) is employed for case study in this section. 7283 samples are divided into two groups, 4854 samples for training and 2429 samples for testing. In addition, weather conditions are considered as input variables, including temperature, relative humidity, wind direction, wind speed, and pressure. For the selection of activation function and model parameters, as shown in Table 3, Leaky ReLU is the best choice for the activation function, and the model parameters are $m' = 5$ and $k = 7$.



(a) PCA (ground truth) (b) DHL-AE (c) CAE (d) DAE

(e) Sparse AE (f) 2-layer Spacked AE (g) 4-layer Spacked AE (h) VAE

Figure 4. Anomaly detection monitoring results of algorithms for comparison in Wind 2017

The anomaly detection monitoring results of the aforementioned algorithms is shown in Figure 4. To further analyze the comparison of different algorithms based on the results of anomaly detection monitoring, as shown in Table 4, a series of metrics are used to express the performance of anomaly detection. Among the aforementioned algorithms, in terms of accuracy, precision, FDR, and MAR, the anomaly detection monitoring based on DHL-AE performs best. Its recall and F1-score are not the highest values, but not far from them; although the performance of DHL-AE on FAR is not the best, it is close to the best values. For methods other than DHL-AE, Sparse AE has the highest F1-score, but other metrics do not perform best; 2-Layer Stacked AE has the best recall and FAR values, while its accuracy, precision, and MAR has the worst performance. Besides, the recall, F1-score, and FAR of VAE have the worst performance. Thus, it is concluded that DHL-AE takes advantages over other AE methods in this case study.

Table 4. Statistical metrics results of algorithms for comparison in Wind 2017

Algorithms	Accuracy	Precision	Recall	F1-Score	FDR	FAR	MAR
DHL-AE	**0.6316**[a]	**0.7192**[a]	0.5270	0.6082	**0.7559**[a]	0.4370	**0.2441**[a]
CAE	0.6308	0.7120	0.5368	0.6121	0.7423	0.4632	0.2577
DAE	0.4977	0.5308	0.6416	0.5810	0.3270[a]	0.3584	0.6730
Sparse AE	0.5810	0.6109	0.6272	**0.6190**[a]	0.5261	0.3728	0.4739
2-Layer Stacked AE	0.4780[a]	0.5151[a]	**0.6492**[a]	0.5744	0.2748	**0.3508**[a]	0.7252[a]
4-Layer Stacked AE	0.5113	0.5417	0.6454	0.5891	0.3523	0.3546	0.6477
VAE	0.4734	0.5245	0.3174[a]	0.3955[a]	0.6856	0.6826[a]	0.3414

a. The values of metrics with best performance are highlighted in bold font, and the worst are displayed in red.

6. Conclusion

To deal with issues of high-dimensionality and time-dependence in the real-world datasets, DHL-AE is proposed in this article, and DHL-AE based anomaly detection monitoring framework is also developed to identify abnormal samples, which is conducive to ensuring the safety of industrial processes and solving the potential problem in the complex industrial applications. Tennessee Eastman process and Wind 2017 are used for the case study to show the effectiveness and superiority of the proposed DHL-AE method and the corresponding scheme for anomaly detection monitoring.

Acknowledgements
We would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Waterloo for funding this research.

References
Zhou, X., Hu, Y., Liang, W., Ma, J. and Jin, Q., Variational LSTM enhanced anomaly detection for industrial big data, *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 3469-3477, 2020.

Xu, R. and Wunsch, D., Survey of clustering algorithms, *IEEE Transactions on Neural Network*s, vol. 16, pp. 645-678, 2005.

Wold, S., Esbensen, K. and Geladi, P., Principal component analysis, *Chemometrics and Intelligent Laboratory Systems*, vol. 2, pp. 37-52, 1987.

Reynolds, D. A., Gaussian mixture models, *Encyclopedia of Biometrics*, vol. 741, pp. 659-663, 2009.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. and Scholkopf, B., Support vector machines, *IEEE Intelligent Systems and Their Application*s, vol. 13, pp. 18-28, 1998.

Breiman, L., Random forests, *Machine Learning*, vol. 45, pp. 5-32, 2001.

Bishop, C. M., Neural networks and their applications, *Review of scientific instruments*, vol. 65, pp. 1803-1832, 1994.

Pang, G., Shen, C., Cao, L. and Hengel, A. V. D., Deep learning for anomaly detection: A review, *ACM Computing Surveys*, vol. 54, pp. 1-38, 2021.

Sakurada, M. and Yairi, T., Anomaly detection using autoencoders with nonlinear dimensionality reduction, *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, Gold Coast, Australia, December 2, 2014, pp. 4-11, 2014.

Tsung, F. G., Statistical monitoring and diagnosis of automatic controlled processes using dynamic PCA, *International Journal of Production Research*, vol. 38, pp: 625-637, 2000.

Yin, J. and Yan, X., Mutual information–dynamic stacked sparse autoencoders for fault detection, *Industrial & Engineering Chemistry Research*, vol. 58, pp. 21614-21624, 2019.

Tayeh, T., Aburakhia, S., Myers, R. and Shami, A., An attention-based ConvLSTM autoencoder with dynamic thresholding for unsupervised anomaly detection in multivariate time series, *Machine Learning and Knowledge Extraction*, vol. 4, pp. 350-370, 2022.

Bishop, C. M. and Nasrabadi, N. M., *Pattern recognition and machine learning*, vol. 4, Springer, New York, 2006.

Hinton, G. E. and Salakhutdinov, R. R., Reducing the dimensionality of data with neural networks, *Science*, vol. 313, pp. 504-507, 2006.

Schaffer, A. L., Dobbins, T. A. and Pearson, S. A., Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions, *BMC medical research methodology*, vol. 21, pp. 1-2, 2021.

Rumelhart, D. E., Hinton, G. E. and Williams, R.J., Learning internal representations by error propagation, *California Univ San Diego La Jolla Inst for Cognitive Science*, 1985.

Goodfellow, I., Bengio, Y. and Courville A., *Deep learning*, MIT press, 2016.

Aggarwal, C. C. and Aggarwal, C. C., *An introduction to outlier analysis*, Springer International Publishing, 2017.

Chalapathy, R. and Chawla, S., Deep learning for anomaly detection: A survey, *arXiv preprint arXiv:1901.03407*, 2019.

Hyndman, R. J. and Athanasopoulos, G., *Forecasting: principles and practice*, OTexts, 2018.

Ahmad, S., Styp-Rekowski, K., Nedelkoski, S. and Kao, O., Autoencoder-based condition monitoring and anomaly detection method for rotating machines, *2020 IEEE International Conference on Big Data (Big Data)*, pp. 4093-4102, Atlanta, USA, December 10-13, 2020.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P. and Shroff G., LSTM-based encoder-decoder for multi-sensor anomaly detection, *arXiv preprint arXiv:1607.00148*, 2016.

Downs, J. J. and Vogel, E. F., A plant-wide industrial process control problem, *Computers & Chemical Engineering*, vol. 17, pp. 245-255, 1993.

Müller, A. C. and Guido, S., *Introduction to machine learning with Python: a guide for data scientists*, O'Reilly Media, Inc., 2016.

Arora, R., Basu, A., Mianjy, P. and Mukherjee, A., Understanding deep neural networks with rectified linear units, *arXiv preprint arXiv:1611.01491*, 2016.

Maas, A. L., Hannun, A. Y. and Ng, A. Y., Rectifier nonlinearities improve neural network acoustic models, *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, USA, 2013.

Clevert, D. A., Unterthiner, T. and Hochreiter, S., Fast and accurate deep network learning by exponential linear units (elus), *arXiv preprint arXiv:1511.07289*, 2015.
Powers, D. M., Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation, *arXiv preprint arXiv:2010.16061*, 2020.
Zhang, H. and Zhu, Q., Concurrent multilayer fault monitoring with nonlinear latent variable regression, *Industrial & Engineering Chemistry Research*, vol. 61, pp. 1423-1442, 2022.
Izadi, I., Shah, S. L., Shook, D.S. and Chen, T., An introduction to alarm analysis and design, *IFAC Proceedings Volumes*, vol. 42, pp. 645-50, 2009.
Alkabbani, H., Ahmadian, A., Zhu, Q. and Elkamel, A., Machine learning and metaheuristic methods for renewable power forecasting: a recent review, *Frontiers in Chemical Engineering*, vol.3, 2021.

Biographies

Haitian Zhang is a Ph.D. Candidate in Chemical Engineering (University of Waterloo, CA). Her research focuses on process monitoring and fault diagnosis based on machine learning algorithms in the field of process systems engineering. Prior to her doctoral study, she secured her Bachelor's and Master's degrees in Chemical Engineering from Xi'an Jiaotong University. She had two PRES conference papers and three journal papers in her masteral program. During the period of her doctoral study, she also published two papers related to multivariate statistical analysis.

Qinqin Zhu is an Assistant Professor in the department of Chemical Engineering (University of Waterloo, CA). She is also a faculty member at the Waterloo Artificial Intelligence Institute (Waterloo.AI), Waterloo Institute for Sustainable Energy (WISE) and Waterloo Institute for Nanotechnology (WIN). She received her PhD degree from the Chemical Engineering department at the University of Southern California, advised by Prof. Joe Qin. Prior to UW, she worked as a senior research scientist at Facebook Inc. in the United States. Her research mainly focuses on developing advanced statistical machine learning methods, process data analytics techniques and optimization algorithms in the era of big data with applications to statistical process monitoring and fault diagnosis. Her research addresses theoretical challenges and problems of practical importance in the area of process systems engineering. By leveraging the power of mathematical modeling and optimization, her group strives to develop advanced multivariate statistical analysis algorithms that enhance decision making in complex engineering systems.

Ali Ahmadian is a Postdoctoral Research Fellow in the Department of Chemical Engineering (University of Waterloo, CA). He received his PhD degree in Electrical Engineering in 2017. He has published more than 80 papers in journals and conference proceedings and one book in Springer. His main research interests include transportation electrification, energy storage, energy and environment, smart cities, and machine learning applications in modern energy systems.

Ali Elkamel is a Full Professor of Chemical Engineering (University of Waterloo, CA). He is also cross appointed in Systems Design Engineering. He holds a BSc in Chemical Engineering and BSc in Mathematics from Colorado School of Mines, MSc in Chemical Engineering from the University of Colorado, and PhD in Chemical Engineering from Purdue University. His specific research interests are in computer-aided modeling, optimization, and simulation with applications to energy planning, sustainable operations, and product design. His activities include teaching graduate and undergraduate courses, supervising post doctorate and research associates, and participation in both university and professional societal activities. He is also engaged in initiating and leading academic and industrial teams, establishing international and regional research collaboration programs with industrial partners, national laboratories, and international research institutes. He supervised over 120 graduate students (of which 47 are PhDs) and more than 45 post-doctoral fellows/research associates. He has been funded for several research projects from government and industry. Among his accomplishments are the Research Excellence Award, the Excellence in Graduate Supervision Award, the Outstanding Faculty Award, and IEOM Awards. He has more than 425 journal articles, 175 proceedings, 50 book chapters, and has been an invited speaker on numerous occasions at academic institutions throughout the world. He is also a co-author of six books.