International Journal of Control, Automation and Systems 20(6) (2022) 1827-1840
http://dx.doi.org/10.1007/s12555-021-0323-6
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Anomaly Detection with GRU Based Bi-autoencoder for Industrial Multimode Process

Xinyao Xu , Fangbo Qin , Wenjun Zhao , De Xu* , Xingang Wang , and Xihao Yang

Abstract: The anomaly detection for multimode industrial process is a challenging problem, because the multiple operation modes present various main distributions of monitored variables, and the dynamic sequential characteristics exist within each operation mode. This paper proposes an anomaly detection method based on sequence-to-sequence gated recurrent units (SGRU). First, to better model both the cross-mode trends and mode-specific sequential characteristics, a main reconstruction module and residual reconstruction module are integrated to improve the ability to represent complex process. Both modules are implemented by SGRUs. Second, a reconstruction error prediction module is designed to estimate the mean values of mode-specific reconstruction errors, which helps to determine the more reliable alarm thresholds. Third, the two anomaly indicators are utilized to represent the deviation degree of monitored variables against the normal conditions, according to the statistical errors and biases of reconstructions, respectively. The effectiveness of the proposed method is validated on simulations with multimode process, and on the practical data set collected from the Cleaning-in-Place multimode process of an aseptic beverage filling line in a real factory.

Keywords: Anomaly detection, autoencoder, gated recurrent unit, multimode process.

1. INTRODUCTION

Modern industrial production facilities have complex processes and large production scales. Anomaly detection and forewarning are vital for preventing equipment failures, product quality problems, and even severe accidents. Since extensive sensors are deployed in industrial process monitoring, massive data can be collected for real-time anomaly detection [1]. Anomaly detection is usually realized by monitoring the deviation of current status from normal conditions.

There could be a single operation mode or multiple operation modes in an industrial process. The statistical methods are usually applied to monitor the single-mode process [2–9], among which the multivariate statistical process monitoring (MSPM) methods are widely used. Neural networks are widely used to describe complex systems that are difficult to be modeled mathematically. For example, fuzzy neural networks are promising to describe the complex systems with uncertain factors [10–12]. Neural network based methods are also proposed to monitor the industrial process with single operation mode [13,14].

Real industrial production lines are generally adjusted due to many factors such as production schedules and market changes. Hence, an industrial process can include a variety of operating modes. For the industrial processes with multiple operation modes, working conditions can be changed significantly according to the production strategies. Thus, the monitored variables fluctuate in multiple working ranges, making data multimodally distributed. K-Nearest Neighbor [15] and kernel function method [16] can be used to remove the mode-specific characteristics. Then, traditional methods with unimodal data distribution assumptions can be applied to monitor the processes. Deep neural network based autoencoders are widely applied to monitor multimode processes [17–20] as well. To model the dynamic characteristics of industrial processes, the sliding window strategy is generally used to sample the sequences with sequential characteristics. Wu *et al.* proposed an adaptive method to detect abnormal tendencies [20]. The window-based local adaptive standardization (LAS) strategy is developed to remove mode-specific characteristics of data, then the process is monitored with a Long Short-Term Memory (LSTM) network based au-

Manuscript received April 18, 2021; revised June 27, 2021; accepted August 4, 2021. Recommended by Associate Editor Lei Liu under the direction of Editor Bin Jiang. This work was supported by National Key R&D Program of China (2018YFD0400902) and State Key Laboratory of Smart Manufacturing for Special Vehicles and Transmission System (GZ2019KF008).

Xinyao Xu, Fangbo Qin, De Xu, and Xingang Wang are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mails: {xuxinyao2018, qinfangbo2013, de.xu, xingang.wang}@ia.ac.cn). Wenjun Zhao and Xihao Yang are with State Key Laboratory of Smart Manufacturing for Special Vehicles and Transmission System, Baotou City, Inner Mongolia 014000, China (e-mails: {zhaowenjun9930, yangxihao0316}@126.com).
* Corresponding author.

toencoder [21]. The model keeps good performance when encountering new operation modes. Multiple models can be used to monitor the multimode process [22], by using an individual model for each operation mode. Decision-function based methods firstly classify the mode types, then conduct the estimation according to the corresponding mode [23–26]. However, the inaccurate classification might lead to false detection. Bayesian-fusion methods fuse the results from multiple models in probabilistic ways, to provide more robust results [27,28]. When the distributions of different modes overlap with each other, the multiple-model based methods might have degraded accuracy, because of the ambiguity of mode identification.

Due to the complexity of multimode industrial process, the monitoring of the multimode industrial process is still challenging. Many industrial processes are consecutive production processes with various dynamic sequential characteristics. However, many methods only focus on the industrial processes' statistical features but neglect the modeling of sequential characteristics. The sequential characteristics are important to reflect the working states of the industrial process, which should be modeled and monitored properly. Moreover, the task goals of different operation modes are different, so that the monitored variables usually switch among different working ranges. The changes of working ranges lead to varying sequential characteristics, which increases the complexity of the process monitoring.

Autoencoders have good representation ability for sequential data, which is promising for modeling industrial processes with complex characteristics. The sequence-to-sequence based model [29] reconstructs the process with only an initial state that contains the process dynamics. However, 1) it's challenging for a single model to concurrently capture the cross-mode trends and the multiple mode-specific sequential characteristics under different ranges. 2) Reconstruction errors under different operation modes have different distributions, which might lead to unreliable alarm thresholds.

Considering the above limitations, we propose a novel anomaly detection method to detect the anomalies in multimode industrial process. The main contributions are listed as follows:

First, the Bi-SGRU model is proposed to analyze the multimode industrial process with mode-specific sequential characteristics. 1) A main reconstruction module is utilized to model the overall features. 2) A residual reconstruction module is utilized to capture the residual features neglected by the main reconstruction module. Bi-SGRU is the integration of the two modules, which focus on the cross-mode trends and mode-specific sequential details, respectively. Thus, the two-stage architecture enables the model to gain better reconstruction precision for multimode process with smaller network size.

Second, the model is further applied to detect anomalies in multimode processes. 1) A reconstruction error prediction module is designed to estimate the mode-specific reconstruction errors, to determine the alarm thresholds more reliably when operation mode varies. 2) Two anomaly indicators are proposed to indicate the deviations of reconstruction errors and biases, respectively. The final detection result is determined jointly by the two indicators, which is more accurate than the result using the basic Mahalanobis distance.

The remainder of this paper is organized as follows: Section 2 introduces preliminaries of the work. The proposed method is presented in Section 3. Simulations and experiments are conducted in Section 4. Finally, the conclusion is given in Section 5.

2. PRELIMINARIES

2.1. Anomaly detection with autoencoder

As a popular unsupervised framework, autoencoder has been widely applied to extract representations from massive unlabeled data collected from industrial facilities [30,31]. Similar to Principal Component Analysis (PCA) and other dimension-reduction algorithms, autoencoder networks are designed to extract concise low-dimension features from complex processes. The original inputs can be recovered from these features. Many works apply autoencoder-based methods to monitor the status of industrial equipment and processes [13,14,17–20,32]. Generally, these methods construct feature projection functions and inverse projection functions based on neural networks, which are trained only on normal samples. The distributions of reconstruction errors given the normal samples are generally unimodal. Thus, given an abnormal sample as input, the reconstruction error is expected to deviate from the learned unimodal distribution, so that the anomaly can be detected. The higher reconstruction error indicates the larger deviation. Mahalanobis distance is a general indicator to estimate the reconstruction errors. The Mahalanobis distance score s between $\boldsymbol{X}$ and its reconstruction $\hat{\boldsymbol{X}}$ can be used as the anomaly indicator, as calculated by

$$\boldsymbol{e} = \left|\hat{\boldsymbol{X}} - \boldsymbol{X}\right|, \tag{1}$$

$$s = (\boldsymbol{e} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{e} - \boldsymbol{\mu})^{\mathrm{T}}, \tag{2}$$

where $\boldsymbol{e}$ is the absolute difference between input $\boldsymbol{X}$ and its reconstruction $\hat{\boldsymbol{X}}$. s is the anomaly score, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the mean vector and covariance matrix of $\boldsymbol{e}$ over all the training samples. Some works also took models' training losses as indicators [14,20]. The alarm thresholds of the anomaly indicators are estimated by Kernel Density Estimation (KDE),

$$p(s) = \frac{1}{hn}\sum_{i=1}^{n} K\left(\frac{s - s_i}{h}\right), \tag{3}$$

$$\int_{-\infty}^{T_s} p(s)\,ds = \alpha, \tag{4}$$

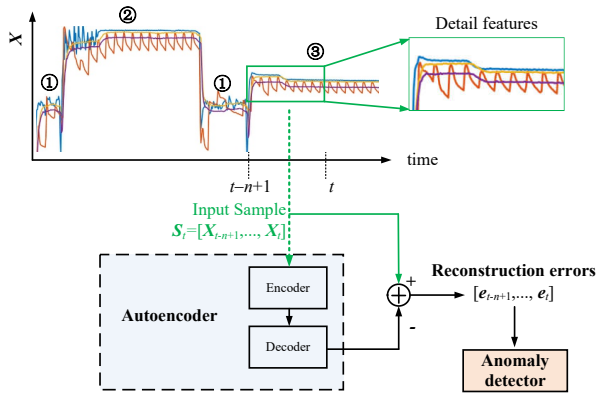


Fig. 1. Sliding window and autoencoder-based anomaly detection for multimode process. The circled numbers on top of the curve show the three different operation modes in the industrial process.

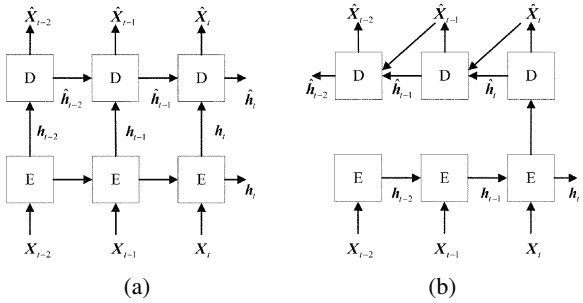


Fig. 2. Comparison of two autoencoder architectures. The RNN structures are unrolled over time steps and the sequence length is assumed as 3. D and E mean the encoder and decoder, respectively. (a) Standard architecture, (b) Sequence-to-sequence architecture.

where $p(s)$ is the probability density of anomaly score s, s_i is the ith anomaly score, n is the number of samples, h is the bandwidth parameter, and $K(\cdot)$ is the kernel function. In this paper, we used the radial basis kernel function. T_s is the alarm threshold, determined by the predefined confidence level α.

To model the sequential characteristics, the sliding window strategy is utilized, as illustrated by Fig. 1. The n sequential data points $[\boldsymbol{X}_{t-n+1}, \boldsymbol{X}_{t-n+2}, \cdots, \boldsymbol{X}_t]$ within the time window are used to monitor the status at time step t.

2.2. GRU based autoencoder

In recent years, various neural network models have been widely applied to describe multivariate complex processes. Among them, Recurrent Neural Network (RNN) is one type of architectures specifically designed for sequential data processing [33]. It can take into account both the morphological characteristics and dynamic trends of the process data simultaneously. GRU is an RNN variant with greater description capability and more concise structure. It is the ideal model for multimode process description. There are two typical RNN based autoencoder architectures, whose unrolled structures are shown in Fig. 2. For expression briefness, we assume the length of the sequential data is 3. Under the first architecture, the encoder E firstly converts the time series input $\boldsymbol{S}_t = [\boldsymbol{X}_{t-2}, \boldsymbol{X}_{t-1}, \boldsymbol{X}_t]$ to the time series of hidden states $[\boldsymbol{h}_{t-2}, \boldsymbol{h}_{t-1}, \boldsymbol{h}_t]$, which is further decoded as the series of hidden states $[\hat{\boldsymbol{h}}_{t-2}, \hat{\boldsymbol{h}}_{t-1}, \hat{\boldsymbol{h}}_t]$ by the decoder D, as illustrated in Fig. 2(a). Finally, the reconstruction $[\hat{\boldsymbol{X}}_{t-2}, \hat{\boldsymbol{X}}_{t-1}, \hat{\boldsymbol{X}}_t]$ is recovered from $[\hat{\boldsymbol{h}}_{t-2}, \hat{\boldsymbol{h}}_{t-1}, \hat{\boldsymbol{h}}_t]$ by the linear transformation $\boldsymbol{W}\boldsymbol{h}+\boldsymbol{b}$. The second RNN based autoencoder architecture has the sequence-to-sequence form [29]. Differently, it firstly infers the hidden state $\boldsymbol{h}_t$ at the latest time step t, according to the time series input $\boldsymbol{S}_t = [\boldsymbol{X}_{t-2}, \boldsymbol{X}_{t-1}, \boldsymbol{X}_t]$. Then, the hidden states $[\hat{\boldsymbol{h}}_{t-2}, \hat{\boldsymbol{h}}_{t-1}, \hat{\boldsymbol{h}}_t]$ are recovered based on only the single vector $\boldsymbol{h}_t$, as is illustrated in Fig. 2(b). The final reconstruction is also obtained by the linear transformation. In comparison, using the first architecture, the reconstruction at time step t highly relies on the current input, lacks the awareness of the dynamic sequential characteristics. Using the second architecture, the model is forced to encode the dynamic sequential characteristics into a single vector, so that the time series can be reconstructed from this vector. Therefore, we prefer to use the sequence-to-sequence architecture to form the autoencoder with better representation ability of sequential characteristics.

3. METHODS

3.1. Bi-SGRU autoencoder for multimode process

In many industrial processes, the multimode operations, such as heating, sterilization, and cooling, are combined to realize the whole production. Between the different operation modes, the monitored variables have different distributions. Moreover, within a single operation mode, the monitored variables can also present the specific sequential characteristics, which are called detail features and are important for anomaly recognition, as shown in Fig. 1. The variation scale of the detail features might be relatively small compared to the cross-mode variation scale. Thus, it is difficult to jointly reconstruct the main cross-mode features and the detail features with a single autoencoder.

Inspired by the idea of boosting algorithm, we propose a Bi-SGRU autoencoder to reconstruct the sequential variables monitored in multimode process, which consists of a main reconstruction module, a residual reconstruction module, and a reconstruction error prediction module, as shown in Fig. 3.

First, the main reconstruction module is a SGRU based

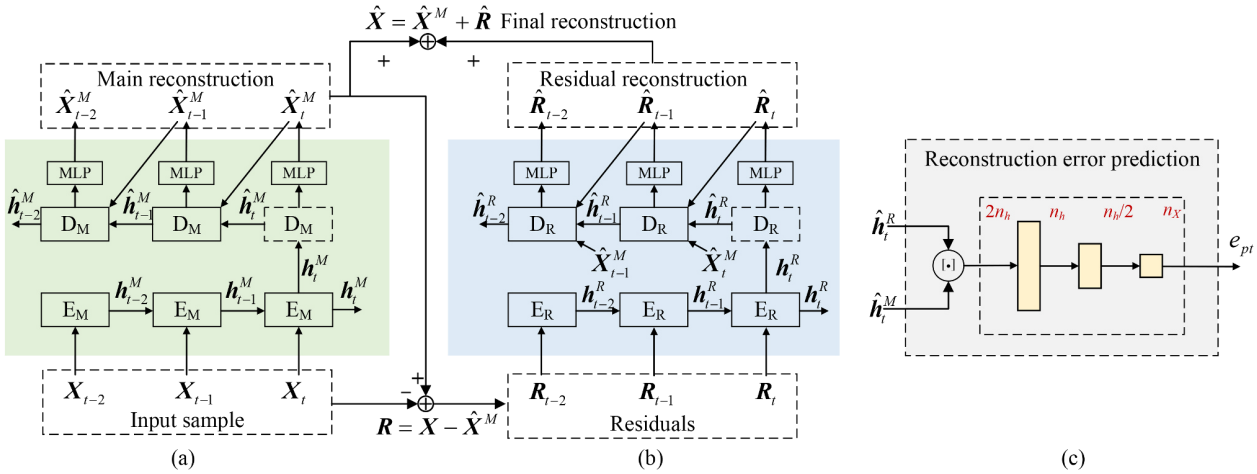


Fig. 3. Architecture of Bi-SGRU. The sequential length is assumed as 3. (a) Main reconstruction module. E_M, D_M are its encoder and decoder, respectively. (b) Residual reconstruction module. E_R, D_R are its encoder and decoder, respectively. $\boldsymbol{h}_i^R$, $\boldsymbol{h}_i^M$, $\hat{\boldsymbol{h}}_i^R$, and $\hat{\boldsymbol{h}}_i^M$ are the hidden states. (c) Reconstruction error prediction module, $\boldsymbol{e}_{p_t}$ denotes the predicted error at time t. n_h and n_X are the dimensions of the hidden state $\boldsymbol{h}$ and the input $\boldsymbol{X}$, respectively.

autoencoder for main feature reconstruction. For expression briefness, the input sequential length is assumed as 3. The main reconstruction module's input and output are $\boldsymbol{S}_t = [\boldsymbol{X}_{t-2}, \boldsymbol{X}_{t-1}, \boldsymbol{X}_t]$ and $\hat{\boldsymbol{S}}_t^M = [\hat{\boldsymbol{X}}_{t-2}^M, \hat{\boldsymbol{X}}_{t-1}^M, \hat{\boldsymbol{X}}_t^M]$, respectively. The main reconstruction is realized by

$$\boldsymbol{h}_t^M = f_{me}(\boldsymbol{S}_t), \tag{5}$$

$$\hat{\boldsymbol{X}}_{t-k}^M = \begin{cases} f_{MLP}\left[f_{md}\left(\hat{\boldsymbol{h}}_{t-k+1}^M, \hat{\boldsymbol{X}}_{t-k+1}^M\right)\right], & (k>0), \\ f_{MLP}\left(\boldsymbol{h}_t^M\right), & (k=0), \end{cases} \tag{6}$$

where $f_{me}(\cdot)$, $f_{md}(\cdot)$ and $f_{MLP}(\cdot)$ denote the functions implemented by the SGRU encoder, SGRU decoder and multi-layer perceptron (MLP) net of the main reconstruction module, respectively. $f_{me}(\cdot)$ involves the whole encoding process. $f_{md}(\cdot)$ and $f_{MLP}(\cdot)$ only involve the calculation of single time step.

When $k=0$, since $\hat{\boldsymbol{X}}_{t+1}^M$ is lacked in the first inferring step of decoder, $\hat{\boldsymbol{X}}_t^M$ is directly calculated by the MLP net with $\hat{\boldsymbol{h}}_t^M = \boldsymbol{h}_t^M$, as shown by the dashed rectangle in Fig. 3(a).

Second, the residual reconstruction module is designed to capture the residual detail features neglected by the main reconstruction module. The residual feature $\boldsymbol{R}_t$ is given by

$$\boldsymbol{R}_t = \boldsymbol{X}_t - \hat{\boldsymbol{X}}_t^M. \tag{7}$$

Thus, after the main features are removed from the raw input, the residual features are more significant for modeling. The residual reconstruction module is also implemented by SGRU, whose structure is different from the main reconstruction module. As shown in Fig. 3(b), the reconstructed main feature $\hat{\boldsymbol{X}}_{t-k+1}^M$ is also fed to the decoder to predict the residual feature at time $t-k$. The sequential characteristics of different modes are different. To provide the awareness of the specific mode for residual decoding, $\hat{\boldsymbol{X}}_{t-k+1}^M$ and $\hat{\boldsymbol{R}}_{t-k+1}$ are concatenated and input to the decoder of $\hat{\boldsymbol{R}}_{t-k}$. The residual reconstruction is expressed by

$$\boldsymbol{h}_t^R = f_{re}\left(\boldsymbol{S}_t - \hat{\boldsymbol{S}}_t^M\right). \tag{8}$$

$$\hat{\boldsymbol{X}}_{t-k+1}^{new} = \left[\hat{\boldsymbol{X}}_{t-k+1}^M, \hat{\boldsymbol{R}}_{t-k+1}\right], \tag{9}$$

$$\hat{\boldsymbol{R}}_{t-k} = \begin{cases} f_{MLP}\left(f_{rd}\left(\hat{\boldsymbol{h}}_{t-k+1}^R, \hat{\boldsymbol{X}}_{t-k+1}^{new}\right)\right), & (k>0), \\ f_{MLP}\left(\boldsymbol{h}_t^R\right), & (k=0), \end{cases} \tag{10}$$

where $f_{re}(\cdot)$ and $f_{rd}(\cdot)$ denote the functions implemented by the SGRU encoder and SGRU decoder of the residual reconstruction module.

Third, the final reconstruction is the sum of the main reconstruction and the residual reconstruction, as given by

$$\hat{\boldsymbol{X}}_t = \hat{\boldsymbol{X}}_t^M + \hat{\boldsymbol{R}}_t. \tag{11}$$

3.2. Reconstruction error prediction for anomaly detection

For industrial multimode process, the feature scales and noise levels of the same monitored variables are usually different under different operation modes. Consequentially, the reconstruction error distribution varies between different operation modes, which affects the calculation of alarm thresholds of anomaly indicators.

Therefore, we additionally design a reconstruction error prediction module, utilizing hidden states of the decoders of the main and residual reconstruction modules to predict

the mean values of the reconstruction errors,

$$\boldsymbol{e}_{p_t} = f_{REP}\left(\left[\hat{\boldsymbol{h}}_t^M, \hat{\boldsymbol{h}}_t^R\right]\right), \tag{12}$$

where f_{REP} denotes the function of reconstruction error prediction module.

The actual reconstruction error is given by the absolute difference

$$\boldsymbol{e}_t = \left|\hat{\boldsymbol{X}}_t - \boldsymbol{X}_t\right|. \tag{13}$$

By subtracting the predicted mean error from the actual error, $\boldsymbol{e}_t - \boldsymbol{e}_{p_t}$ is considered zero-centered, which can be used to determine the alarm thresholds more accurately.

3.3. Training loss

The main reconstruction module, residual reconstruction module, and reconstruction error prediction module are optimized one by one: First, the main reconstruction module is trained individually. Second, the residual reconstruction module is trained while the weights of the main reconstruction module are frozen. Finally, the reconstruction error prediction module is trained, while the weights of the above two modules are frozen. The training loss functions for the three modules are as follows:

$$L_M = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i^M\right)^2, \tag{14}$$

$$L_R = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i^M - \hat{\boldsymbol{R}}_i\right)^2, \tag{15}$$

$$L_{REP} = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i^M - \hat{\boldsymbol{R}}_i\right| - \boldsymbol{e}_{p_i}\right)^2, \tag{16}$$

where n is the sequence length of the input sample.

3.4. Measurement of anomalies

The anomaly degree of samples can be indicated by their reconstruction errors. The higher reconstruction error indicates the larger probability of anomaly. The Mahalanobis distance is a popular indicator [29,32]. However, Mahalanobis distance might be affected by the fluctuations and noises. As shown by the 2D example in Fig. 4, the samples in set 1 (blue) and set 2 (red) have similar average Mahalanobis distance values to the zero-centered basic Gaussian distribution (green). The samples in set 1 are generated from the basic Gaussian distribution, and the average distance is 1.846. The samples in set 2 are generated from a different Gaussian distribution with a biased center, and the average Mahalanobis distance is 1.772. Therefore, the distinction ability of Mahalanobis distance might degrade under noisy fluctuations.

To overcome the problem, we proposed the two indicators, the statistical reconstruction error based distance s_{et} and the statistical reconstruction bias based distance s_{bt},

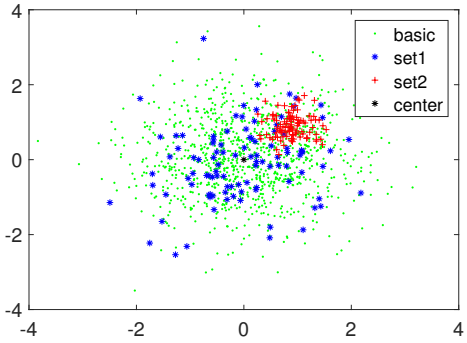


Fig. 4. Example to illustrate the limitation of Mahalanobis distance based indicator. The average deviation scores of the samples in set 1 and set 2 are similar using Mahalanobis distance.

which are calculated by

$$\boldsymbol{e}_{et} = \frac{1}{n}\sum_{k=0}^{n-1}\left(\left|\hat{\boldsymbol{X}}_{t-k} - \boldsymbol{X}_{t-k}\right| - \boldsymbol{e}_{p_{t-k}}\right), \tag{17}$$

$$s_{et} = \left(\boldsymbol{e}_{et} - \boldsymbol{\mu}_1\right)\boldsymbol{\Sigma}_1^{-1}\left(\boldsymbol{e}_{et} - \boldsymbol{\mu}_1\right)^{\mathrm{T}}, \tag{18}$$

$$\boldsymbol{e}_{bt} = \frac{1}{n}\sum_{k=0}^{n-1}\left(\hat{\boldsymbol{X}}_{t-k} - \boldsymbol{X}_{t-k}\right), \tag{19}$$

$$s_{bt} = \left(\boldsymbol{e}_{bt} - \boldsymbol{\mu}_2\right)\boldsymbol{\Sigma}_2^{-1}\left(\boldsymbol{e}_{bt} - \boldsymbol{\mu}_2\right)^{\mathrm{T}}, \tag{20}$$

where $\boldsymbol{e}_{et}$, $\boldsymbol{e}_{bt}$ represent the statistical error and statistical bias of reconstruction. s_{et} and s_{bt} are anomaly indicators. $\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}_2$ are the mean vectors and covariance matrices of $\boldsymbol{e}_{et}$ and $\boldsymbol{e}_{bt}$, respectively.

The alarm thresholds of s_{et} and s_{bt} are estimated by Kernel Density Estimation. Besides, the smoothing filter is applied to the sequential anomaly scores, as realized by

$$S_{et} = \frac{\sum_{k=0}^{m-1} c^k s_{et-k}}{\sum_{k=0}^{m-1} c^k}, \tag{21}$$

$$S_{bt} = \frac{\sum_{k=0}^{m-1} c^k s_{bt-k}}{\sum_{k=0}^{m-1} c^k}, \tag{22}$$

where m is the length of the filtering time window, and c is a decay parameter. S_{et} and S_{bt} are the final anomaly scores output by the filters.

The anomaly detection process is summarized in Fig. 5. Anomaly signals are emitted when one or both indicator values exceed the corresponding alarm thresholds. In Fig. 5, the upper thresholds and the lower thresholds of monitored variables are predefined, which are calculated by the working ranges of those variables.

4. SIMULATIONS AND EXPERIMENTS

Simulations and experiments were conducted to verify the effectiveness of the proposed model. The experiments were based on the temperature monitoring data,

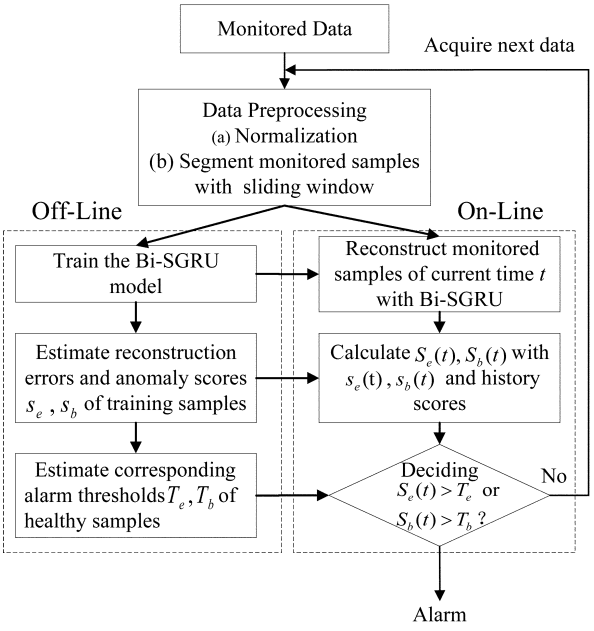


Fig. 5. The flow chart of the method.

Table 1. Parameter settings of three modes.

Mode no.	H_1	$\omega(/s)$	b	$H_2(/s)$	${}^2p_1(s)$
1	0.2	0.8π	0.5π	0.0	0.0
2	0.3	1.6π	0.25π	0.2	2.4
3	0.1	1.5π	-0.25π	0.2	0.8
Mode no.	${}^2p_2(s)$	H_3	${}^3p_1(s)$	${}^3p_2(s)$	*bias*
1	0.0	0.2	4.5	9.0	0.2
2	4.8	0.0	0.0	0.0	0.5
3	1.6	0.2	5.4	9.0	0.7

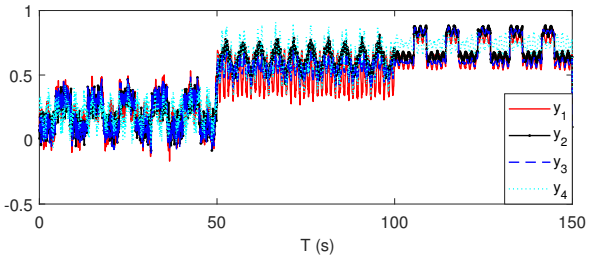


Fig. 6. Visualization of the simulation process.

which was collected from a practical aseptic beverage filling line. The hardware configurations include an Intel i9-9900 CPU and an NVIDIA 2080ti GPU. The deep learning framework is Pytorch 1.6.0.

4.1. Anomaly detection with multimode simulation process

The simulation process is generated by (23). It includes four observation variables $\boldsymbol{Y}(t) = [y_1(t), y_2(t), y_3(t), y_4(t)]$ and four source variables $\boldsymbol{U}(t) = [u_1(t),\ u_2(t),\ u_3(t),\ u_4(t)]$.

$$\boldsymbol{Y}(t) = (1-\gamma)\boldsymbol{Y}(t-1) + \gamma \boldsymbol{A}\boldsymbol{U}(t), \tag{23}$$

$$\boldsymbol{A} = \begin{bmatrix} 0.5 & 0.0 & 0.5 & 1 \\ 0.0 & 0.5 & 0.5 & 1 \\ 0.2 & 0.2 & 0.6 & 1 \\ 0.5 & 0.5 & 0 & 1 \end{bmatrix}, \tag{24}$$

where $\boldsymbol{A}$ is the transform matrix, γ is a weight, which was set as 0.8 in this simulation.

$$u_1(t) = H_1 \sin(\omega t + b) + n_1(t), \tag{25}$$

$$u_2(t) = \begin{cases} H_2 \Delta t + n_2(t), & \Delta t < {}^2p_1, \\ H_2(2\,{}^2p_1 - \Delta t) + n_2(t), & \Delta t \geq {}^2p_1, \end{cases} \tag{26}$$

$$(\Delta t = t - \lfloor t/{}^2p_2 \rfloor {}^2p_2),$$

$$u_3(t) = \begin{cases} -H_3 + n_3(t), & \Delta t < {}^3p_1, \\ H_3 + n_3(t), & \Delta t \geq {}^3p_1, \end{cases} \tag{27}$$

$$(\Delta t = t - \lfloor t/{}^3p_2 \rfloor {}^3p_2),$$

$$u_4(t) = bias, \tag{28}$$

where $u_1(t)$, $u_2(t)$, $u_3(t)$, $u_4(t)$ are signal sources: $u_1(t)$, $u_2(t)$, $u_3(t)$ are three dynamic sources: $u_1(t)$ is a sinusoidal signal. $u_2(t)$ is a saw-tooth signal with period ${}^2p_2$, signal reaches its peak value at $\Delta t = {}^2p_1$. $u_3(t)$ is a step signal with period ${}^3p_2$ and duty ratio $(1 - {}^3p_1/{}^3p_2)$. $\lfloor \cdot \rfloor$ denotes the rounding down operation. $u_4(t)$ is a static bias signal that is specific to each mode. $n_1(t)$, $n_2(t)$, $n_3(t)$ are mode-specific noise items with distribution $N(\mu, \sigma)$: $n_1(t) \sim N(0, 0.1)$, $n_2(t) \sim N(0, 0.05)$ and $n_3(t) \sim N(0, 0.01)$.

The parameters used in the simulation process include $[H_1,\ \omega,\ b,\ H_2,\ {}^2p_1,\ {}^2p_2,\ H_3,\ {}^3p_1,\ {}^3p_2,\ bias]$, whose values under the three operation modes are shown in Table 1. The physical units are listed in brackets beside the parameters' names. The simulation process is visualized in Fig. 6. The sampling interval is 0.05 seconds.

The Diff-PCA [15], LAS-VB [20], GRU-AE, SGRU [29] and the proposed Bi-SGRU were compared in this section. GRU-AE was a GRU autoencoder with the standard structure. Since the functions of LSTM and GRU are similar, the method with LSTM in [29] was replaced with SGRU in order to compare more fairly. Models' structure configurations are displayed in Table 2. The two numbers in the brackets denote the [input dimension, output dimension] of the network. The MLP based reconstruction error prediction module had three layers, whose dimensions were 48, 24, and 4, respectively. The filter size of Bi-SGRU was 20 and the decay parameter c was 0.8.

The training set was a normal sequence collected within 4000 seconds. Data switched among three modes periodically. Each mode continued for 100 seconds. Another normal sequence lasting for 4000 seconds was used to calculate the alarm thresholds. The initial states of the observed

Table 2. Configurations of different autoencoders.

Model	Encoder struc.	Decoder struc.
LAS-VB [20]	LSTM [4,48]	LSTM [48,4]
GRU-AE	GRU [4,48]	GRU [48,4]
SGRU [29]	GRU [4,48]	GRU [4,48]
Bi-SGRU (main)	GRU [4,48]	GRU [4,48]
Bi-SGRU (residual)	GRU [4,48]	GRU [8,48]

variables were set as $[y_1(0),\ y_2(0),\ y_3(0),\ y_4(0)] = [0,\ 0,\ 0,\ 0]$, and the initial time was set as $t_{ini} = 0$ second. The mode switching order of the process followed the cycle of "Mode 1→Mode 2→Mode 3". All the thresholds were calculated by Kernel Density Estimation with $\alpha = 0.95$. Diff-PCA's inputs were individual multivariate data, which were represented by 4D vectors. The number of principal components was determined by cumulative percent variance (CPV), which was set as CPV≥ 0.85. Except for Diff-PCA, the other four autoencoders used the sequential inputs segmented by a sliding window with length 80 and step 10. Thus, each sample was expressed as an 80*4 matrix. The first 80% samples in the training set were used for model training and the rest samples for validation. The Adam optimizer was adopted, and the max training iteration was 800. The initial learning rate was 0.001 and decayed by 0.8 every 80 iterations. The training process ended when the training procedure reached the max iteration or the loss on the validation set converged.

The sequential signal y_2 and its reconstruction with the proposed Bi-SGRU are shown in Fig. 7. The main reconstruction module reconstructed the main trends of the curve, but it could not capture the jagged detail features. With the residual reconstruction module, the residual features neglected by the main reconstruction module could be precisely reconstructed, so that the final reconstruction results reflected both the main trends and the detailed sequential characteristics. Fig. 8 shows the reconstruction errors $\boldsymbol{e}_e$ before and after using the reconstruction error prediction and subtraction. After predicted mean reconstruction errors were subtracted from $\boldsymbol{e}_e$, the results were zero-centered and more suitable for anomaly detection.

Four types of anomalies were designed, including two deviations of frequency parameters and two deviations of amplitude parameters. For each anomaly type, 10 deviation settings were used, which denote different degrees of deviation. Details of anomalies are described in Table 3. The length of the test sequence was 1100 seconds (2192 samples). The initial settings of the simulation system were the same as for the training set, except for the mode type. The formal records started from the 50th second (92nd sample) to avoid the initial unstable switching process. Anomalies started from the 300th second (592nd sample). The fault detection rate (FDR) was used as the evaluation metric.

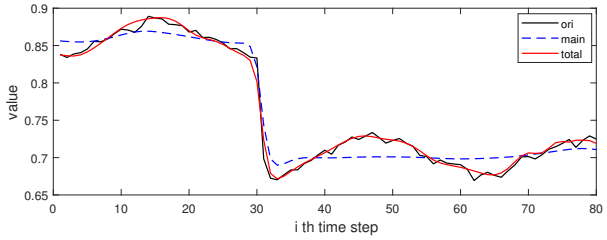


Fig. 7. Reconstructions of sequential signal y_2. 'ori' denotes the raw input sequence. 'main' denotes the signal reconstructed by the main reconstruction module. 'total' denotes the final reconstructions determined by both the main and residual reconstruction modules.

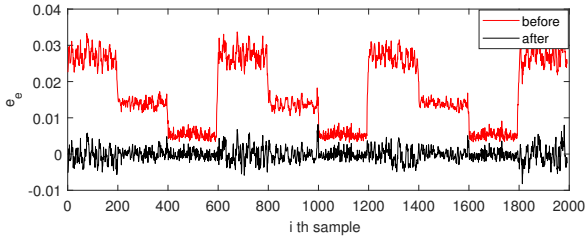


Fig. 8. Comparison of reconstruction error e_e of y_1 before and after using the reconstruction error prediction and subtraction.

Table 3. Deviations to generate simulated anomalies.

No.	Mode	Para.	Deviations
1	2	ω	-π, -0.8π, -0.6π, -0.4π, -0.2π, +0.2π, +0.4π, +0.8π, +1.2π, +1.6π
2	3	${}^3p_1$	-3.0, -2.4, -1.8, -1.2, -0.6, +0.6, +1.2, +1.8, +2.4, +3.0
3	2	H_1	-0.25, -0.20, -0.15, -0.10, -0.05, +0.10, +0.15, +0.20, +0.25, +0.30
4	1	*bias*	-0.15, -0.10, -0.08, -0.05, -0.02, +0.10, +0.20, +0.30, +0.40, +0.50

Table 4. Average FDR (%) of different models for 4 simulated anomalies.

Model	A1	A2	A3	A4
Diff-PCA (T2-diff)	0.0	0.0	0.0	19.37
Diff-PCA (q-diff)	6.70	0.33	24.09	42.99
LAS-VB	0.73	0.41	12.30	5.47
GRU-AE	36.82	19.54	52.90	61.82
SGRU	99.81	16.57	21.48	89.97
Bi-SGRU	99.96	85.59	78.69	94.32

The average anomaly detection rates of models under different simulation settings are shown in Table 4. 'Ai' denotes the anomaly type i. The proposed model achieved the best average performance. The detailed detection per-

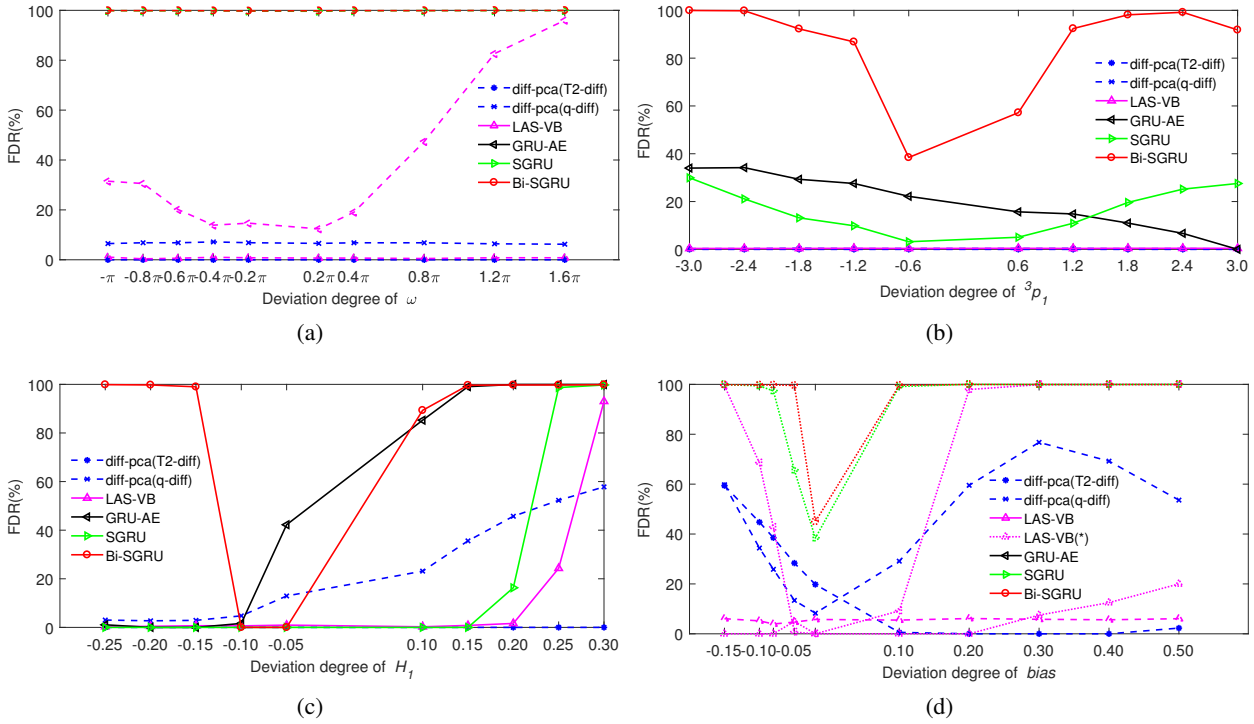


Fig. 9. Detection results for 4 kinds of anomalies under different simulated deviation degrees. (a) Anomaly 1, caused by frequency deviation. (b) Anomaly 2, caused by duty-ratio deviation. (c) Anomaly 3, caused by amplitude deviation. (d) Anomaly 4, caused by static bias deviation. 'LAS-VB(*)' shows the LAS-VB's detection results for the first 40 anomaly samples.

formances under different deviation degrees are shown in Fig. 9. Each sub-figure shows the models' performances for an anomaly under different deviation settings.

The deviations of frequency parameters are simulated as the abnormal sequence characteristics, which could not be modeled by Diff-PCA. Therefore, Diff-PCA failed to detect anomaly 1 and anomaly 2, as shown in Figs. 9(a) and 9(b). The LAS-VB could only detect the abnormal biases with significant deviation degree, as shown in Figs. 9(c) and 9(d). With the window-based standardization strategy LAS, the LAS-VB removed stable abnormal features of static bias signal u_4. Therefore, the LAS-VB only detected the first few abnormal samples with abnormal tendencies. Under the setting of anomaly 4 with the deviation 0.5, the detection accuracy of LAS-VB(*) was significantly higher than LAS-VB, as shown in Fig. 9(d). The GRU-AE with the standard autoencoder structure was compared to the SGRU with the sequence-to-sequence structure. The SGRU could detect the abnormal sequential characteristics caused by the deviation of frequency parameter ω. In contrast, the GRU-AE with the standard autoencoder structure had a lower detection accuracy, as shown in Fig. 9(a). In comparison, the Bi-SGRU had the best overall anomaly detection accuracy.

Fig. 10 displays the detection results of three different methods under the five anomaly deviation settings. The Diff-PCA could not distinguish the samples with abnormal sequential characteristics from normal ones, as shown in Fig. 10(a). The sinusoidal signal u_1 data points with the abnormal amplitude H_1 were partially misidentified as normal ones, causing the periodic alarms shown in Fig. 10(b).

For the SGRU, the training samples from different operation modes had reconstruction errors with different distributions, which affected the estimation of the alarm threshold. The calculated alarm threshold of SGRU was too high to detect the anomalies in Fig. 10(c). Meanwhile, the alarm threshold was too low for the samples from operation mode 1, the model gave frequently false alarms to the normal samples in Fig. 10(d). After the subtraction of predicted reconstruction errors, this influence of distribution changes on the alarm threshold calculation was greatly relieved. The comparison of Figs. 10(e) and 10(f) shows that the indicator S_b had the ability to detect the abnormal biases, which could not be reflected by S_e.

Fig. 11 visualizes the data distributions of the normal data points and abnormal data points under different anomaly settings. The normal data points displayed a complex non-unimodal distribution in the original data space, as shown in Fig. 11(a). After reconstruction, the reconstruction errors of the normal data fitted an approximate unimodal distribution, as the red clusters in

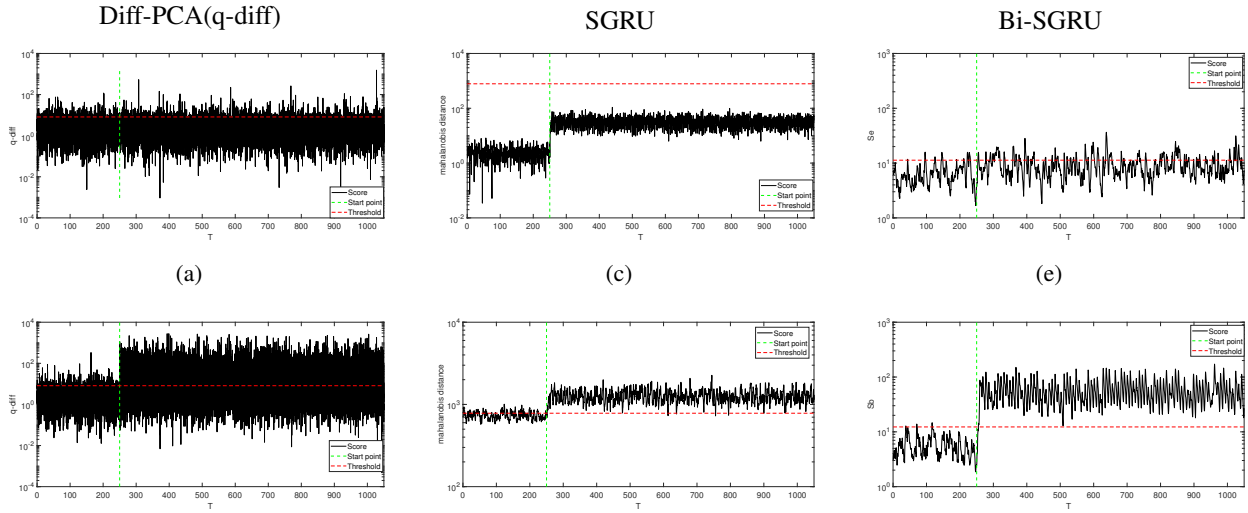


Fig. 10. Anomaly detection results using three different methods. The horizontal red dashed lines indicate the alarm thresholds. The vertical green lines indicate the start time of anomalies. (a) Diff-PCA (q-diff) (anomaly 1, deviation= 0.8π) (b) Diff-PCA (q-diff) (anomaly 3, deviation= 0.2). (c) SGRU (anomaly 3, deviation= -0.2) (d) SGRU (anomaly 4, deviation= -0.1). (e) Bi-SGRU (S_e) (anomaly 4, deviation= -0.05). (f) Bi-SGRU (S_b) (anomaly 4, deviation= -0.05).

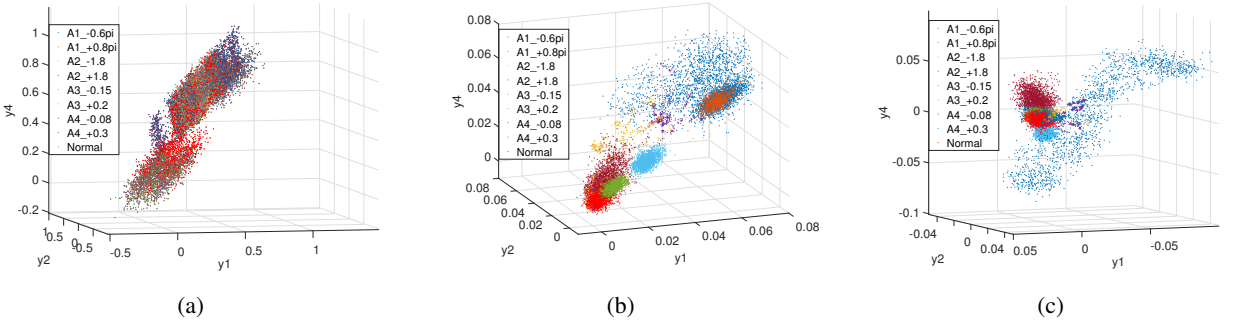


Fig. 11. Data distribution of original records, reconstruction errors, and reconstruction biases along axes y_1,y_2, and y_4. The normal data points are shown in red.There are 8 different anomaly settings, and the abnormal data points under each anomaly setting are shown by a different color. 'Ai_d' denotes the simulation setting of Anomaly type i with the deviation d. (a) Distributions of the original records. (b) Distributions of the reconstruction error e_e. (c) Distributions of the reconstruction bias e_b.

Fig. 11(b) and Fig. 11(c). Moreover, although abnormal data points heavily overlapped with normal data points in the original data space, they were distinguishable in reconstruction error space. Consequently, the anomalies could be identified by relatively simple borders.

The fitting abilities of SGRU and Bi-SGRU were further compared under the different hidden node number configurations. The network's size was denoted with the number of parameters, as the values under the points in Fig. 12. The fitting ability was evaluated by the reconstruction errors, indicated by Mean Square Error (MSE). The lower error means the better fitting ability. As shown in Fig. 12, the proposed Bi-SGRU provided significantly lower reconstruction errors with smaller network sizes, compared to the SGRU, which demonstrated that the Bi-SGRU had the improved fitting ability for the sequential data of the multimode process.

4.2. Anomaly detection experiment with real multimode process

The anomaly detection experiment on the real industrial multimode process was conducted on the Cleaning in place (CIP) process of an aseptic filling line of a beverage factory. CIP is the process of cleaning the inner surface of the production equipment and transportation pipelines. It is a necessary step before performing beverage filling operations. The CIP process is executed by the CIP automatic washer, which mainly consists of acid storage tanks,

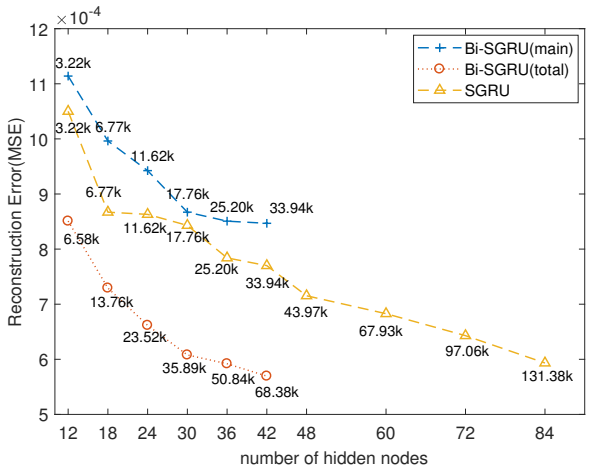


Fig. 12. Comparison of reconstruction errors of Bi-SGRU and SGRU. 'Bi-SGRU (main)' denotes Bi-SGRU's main reconstruction module. 'Bi-SGRU (total)' denotes the entire Bi-SGRU. The values under the data points show the models' parameter numbers, they reflect the model complexity.

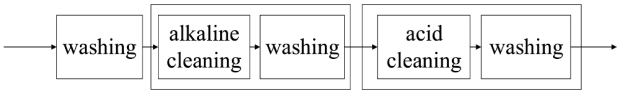


Fig. 13. CIP procedure with five steps.

lye storage tanks, water storage tanks, and related control valves.

As shown in Fig. 13, the CIP process mainly consists of five steps, including 3 operation modes. Firstly, the warm water flows through the equipment to rinse the residue inside equipment and pipelines. Secondly, NaOH solution in the lye storage tank flows through the equipment to decompose residual protein and fat scaling. Thirdly, warm water is flowed through again to remove the residual NaOH solution inside the equipment. Afterward, nitric acid solution is used to remove the scaling of the inorganic salts. Finally, a washing process is conducted to remove residual nitric acid solution. The temperature settings of the alkaline cleaning process and the acid cleaning process range between 80-85°C and 70-75°C, respectively.

Records of four temperature sensors were used in this experiment. These sensors corresponded to the CIP processes of a balance tank. Only the records during CIP processes were analyzed. The sampling interval was 10 seconds. Fig. 14 shows the normal temperature trends of a CIP process. Values had been normalized to [0, 1] in advance. The process could be divided into three stable cleaning modes and the transitions among them. The valve monitored by signal 2 opened periodically, which caused periodic fluctuations of signal 2. During the transition between the first washing process and the alkaline washing process, water was gradually flowed out from the turbines

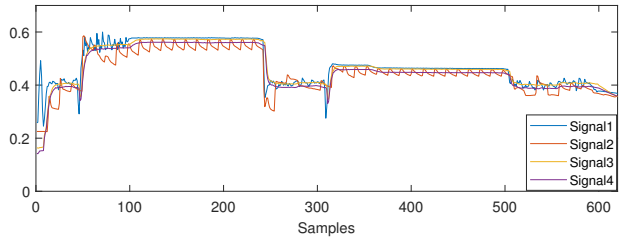


Fig. 14. Normal trends of 4 temperature sensors.

Table 5. Configurations of different autoencoders.

Model	Encoder struc.	Decoder struc.
LAS-VB [20]	LSTM [4,72,36]	LSTM [36,72,4]
GRU-AE	GRU [4,72,36]	GRU [36,72,4]
SGRU [29]	GRU [4,72,72]	GRU [4,72,72]
Bi-SGRU (global)	GRU [4,72,72]	GRU [4,72,72]
Bi-SGRU (detail)	GRU [4,72,72]	GRU [8,72,72]

and tank by nitrate solution, which caused the temperature fluctuations during the transition.

We collected the records of 27 entire normal CIP processes on the production line, to construct the training set. After data normalization, the sliding window method was used to convert the original record data to sequential samples. The length of the sliding window was 40 and the sliding step was 3. Thus, about 2800 samples were extracted in total. Among them, 2500 normal samples were used for model training and another 300 normal samples were used for validation. The validation set was applied to estimate the alarm threshold as well. Another 60 completely documented records collected within 8 months were used for model testing. 20 records among them were normal and the rest included anomalies. Moreover, with time shifted, the working states of the production line drifted from the initial states. Within the 40 abnormal records, approximate 20 records with drifted values were considered as abnormal ones. Alarm to the abnormal record was considered as true positive detection. Alarm to the normal record was considered as false positive detection.

Since the transition duration between different operation modes is short, the samples describing transition are inadequate in number for training. As a result, the trained model might incorrectly output alarm during transitions. Therefore, only when at least L consecutive alarms are given, the alarm for the current record is considered valid.

The autoencoder-based anomaly detection methods used in Subsection 4.1 were compared with the aforementioned real data of CIP process. The configurations of model structures are listed in Table 5. The values in brackets denote the [input dimension, hidden dimension, output dimension] of SGRUs. The reconstruction error prediction network of Bi-SGRU was a MLP with three layers, whose dimensions were [72,36,4]. The training settings of

Table 6. Training parameter configurations.

Parameters	Settings
Optimizer	Adam optimizer
Maximum iteration	500
Batch size	40
Initial learning rate	0.001
Decay ratio of learning rate	0.8
Decay period	40 iterations

Table 7. Anomaly detection accuracies (%) of different models for abnormal records under different settings of L.

Model	$L=1$	$L=3$	$L=5$	$L=7$	$L=9$
LAS-VB	55	30	17.5	17.5	7.5
GRU-AE	100	100	100	97.5	87.5
SGRU	100	100	100	100	90
Bi-SGRU	100	100	100	97.5	97.5

Table 8. False detection rates (%) of different models for normal records under different settings of L.

Model	$L=1$	$L=3$	$L=5$	$L=7$	$L=9$
LAS-VB	70	0	0	0	0
GRU-AE	65	65	60	45	5
SGRU	90	50	50	30	0
Bi-SGRU	50	35	20	5	0

all methods were the same, as shown in Table 6. The alarm thresholds were calculated by Kernel Density Estimation with $\alpha = 0.99$. A filter window was applied to calculate the smoothed results. Its window size was 20. The decay parameter c was 0.8.

Tables 7 and 8 show the detection performances using the metrics of detection accuracy and false detection rate. Note that the alarm thresholds of GRU-AE and SGRU were calculated by all the 2800 samples using traditional indicators of Mahalanobis distance. As L increased, the false detection rate of each model decreased, meanwhile, some abnormal records were also missed in detection. Although the LAS-VB had the lowest false detection rates under the 5 settings, it had low detection accuracy for abnormal records as well. The Bi-SGRU achieved the highest detection accuracy for abnormal records, and the relative low false detection rates for normal records.

Fig. 15 shows the detailed detection results of three abnormal records. All the alarm thresholds were calculated with merely the validation set. As shown in Figs. 15(a)-15(c), the 3 sequential records had different abnormal sequential characteristics, which are abbreviated as 'R_a1', 'R_a2' and 'R_a3', respectively. Signal 2 in 'R_a1' had no response during the whole process, namely deviated from the normal working range. In Fig. 15(b), an obviously unstable transition occurred between the first washing process and the alkaline washing process. In Fig. 15(c), two anomalies occurred: First, the unstable records occurred during the alkaline washing process. Second, compared with the normal values during the water washing process (around 0.4), the records after the alkaline process dropped to 0.2 rapidly, showing the significantly irregular trends.

The rest sub-figures Figs. 15(d)-15(r) show the anomaly detection results of the four different models. Since the monitored sequences were segmented by a sliding window with size 40, the evaluation started from the 40th time step of the original records. For LAS-VB, the local adaptive standardization (LAS) strategy could remove the mode-specific statistical biases from the multimode process data, however, it might also remove the abnormal bias of Signal 2 in 'R_a1'. This limitation led to the false detection results in Fig. 15(d). Although LAS-VB precisely reconstructed process records, it failed to extract the valid features of process dynamics, and failed to detect the abnormal trend, as is shown by the similar detection results of normal and abnormal samples in Figs. 15(e) and 15(f). GRU-AE and SGRU based methods could detect all the anomalies. However, the low alarm thresholds caused many false alarms in 'R_a2' and 'R_a3', as shown in Figs. 15(g)-15(i). In comparison, Bi-SGRU provided the most precise indications of anomalies with sequential characteristics, as shown in Figs. 15(m)-15(r). All abnormal biases and trends were effectively detected with Bi-SGRU via the combination of S_e and S_b.

5. CONCLUSION

In this paper, a Bi-SGRU based anomaly detection method is proposed to detect the abnormal sequential characteristics in multimode industrial process. Bi-SGRU utilizes the main and residual reconstruction modules to concurrently model the cross-mode trends and the model-specific sequential characteristics. Moreover, a reconstruction error prediction module is designed to estimate the mean reconstruction errors under specific modes, to determine alarm thresholds more reliably under varying operation modes. The two Mahalanobis-based anomaly indicators are proposed according to the statistical errors and biases of reconstructions, respectively, and jointly utilized to reflect the anomaly degree. The simulations and experiments with multiple operation modes are conducted to verify the effectiveness of the proposed methods.

In future research, we will focus on the increment updating techniques of the anomaly detection model, aiming to enable the model to adapt to new operation mode timely and efficiently.

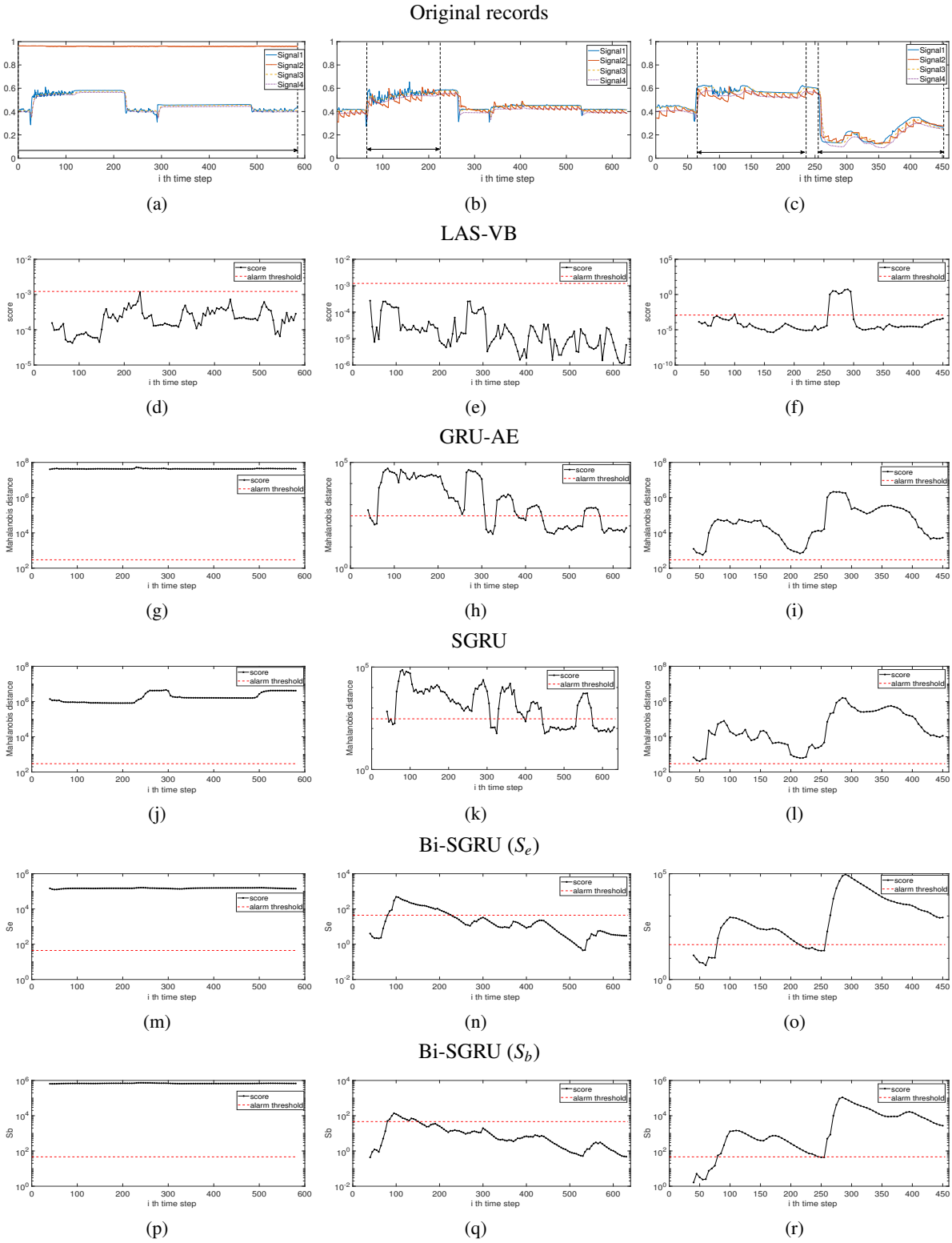


Fig. 15. Records with anomalies and models' detection results. The black solid line shows the anomaly scores. The red dashed line shows the alarm threshold. The first row show the original records. The second to fifth rows show the anomaly scores over time, given by the different methods including LAS-VB, GRU-AE, SGRU and Bi-SGRU. The three columns correspond to the three CIP processes, labeled as (R_a1), (R_a2), and (R_a3), respectively.

REFERENCES

[1] A. Diez-Olivan, J. D. Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0," *Information Fusion*, vol. 50, pp. 92-111, October 2019.

[2] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual Reviews in Control*, vol. 36, no. 2, pp. 220-234, December 2012.

[3] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418-6428, November 2014.

[4] J.-H. Cho, J.-M. Lee, S. W. Choi, D. Lee, and I.-B. Lee, "Fault identification for process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 60, no. 1, pp. 279-288, January 2005.

[5] R. Fezai, M. Mansouri, O. Taouali, M. F. Harkat, and N. Bouguila, "Online reduced kernel principal component analysis for process monitoring," *Journal of Process Control*, vol. 61, pp. 1-11, January 2018.

[6] C. Shang, X. Huang, J. A. Suykens, and D. Huang, "Enhancing dynamic soft sensors based on DPLS: A temporal smoothness regularization approach," *Journal of Process Control*, vol. 28, pp. 17-26, April 2015.

[7] C.-C. Hsu, M.-C. Chen, and L.-S. Chen, "A novel process monitoring approach with dynamic independent component analysis," *Control Engineering Practice*, vol. 18, no. 3, pp. 242-253, March 2010.

[8] P. P. Odiowei and Y. Cao, "Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 36-45, February 2010.

[9] J. Zheng and C. Zhao, "Enhanced canonical variate analysis with slow feature for dynamic process status analytics," *Journal of Process Control*, vol. 95, pp. 10-31, November 2020.

[10] R. Abiyev and S. Abizada, "Type-2 fuzzy wavelet neural network for estimation energy performance of residential buildings," *Soft Computing*, vol. 16, no. 4, pp. 1783-1793, May 2021.

[11] Y. Gao, J. Liu, Z. Wang, and L. Wu, "Interval type-2 FNN-based quantized tracking control for hypersonic flight vehicles with prescribed performance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1981-1993, March 2021.

[12] S. N. Qasem and A. Mohammadzadeh, "A deep learned type-2 fuzzy neural network: Singular value decomposition approach," *Applied Soft Computing*, vol. 105, p. 107244, July 2021.

[13] Z. Zhang, T. Jiang, S. Li, and Y. Yang, "Automated feature learning for nonlinear process monitoring – An approach using stacked denoising autoencoder and k-nearest neighbor rule," *Journal of Process Control*, vol. 64, pp. 49-61, April 2018.

[14] F. Cheng, Q. P. He, and J. Zhao, "A novel process monitoring approach based on variational recurrent autoencoder," *Computers & Chemical Engineering*, vol. 129, p. 106515, October 2019.

[15] C. Zhang, Q. Guo, and Y. Li, "Fault detection in the tennessee eastman benchmark process using principal component difference based on k-nearest neighbors," *IEEE Access*, vol. 8, pp. 49999-50009, 2020.

[16] R. Tan, J. R. Ottewill, and N. F. Thornhill, "Nonstationary discrete convolution kernel for multimodal process monitoring," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3670-3681, September 2020.

[17] Z. Deng, Y. Li, H. Zhu, K. Huang, Z. Tang, and Z. Wang, "Sparse stacked autoencoder network for complex system monitoring with industrial applications," *Chaos, Solitons & Fractals*, vol. 137, p. 109838, August 2020.

[18] P. Tang, K. Peng, J. Dong, K. Zhang, and S. Zhao, "Monitoring of nonlinear processes with multiple operating modes through a novel gaussian mixture variational autoencoder model," *IEEE Access*, vol. 8, pp. 114487-114500, 2020.

[19] F. Lv, C. Wen, and M. Liu, "Representation learning based adaptive multimode process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 181, pp. 95-104, October 2018.

[20] H. Wu and J. Zhao, "Self-adaptive deep learning for multimode process monitoring," *Computers & Chemical Engineering*, vol. 141, p. 107024, October 2020.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, November 1997.

[22] M. Quiñones-Grueiro, A. Prieto-Moreno, C. Verde, and O. Llanes-Santiago, "Data-driven monitoring of multimode continuous processes: A review," *Chemometrics and Intelligent Laboratory Systems*, vol. 189, pp. 56-71, June 2019.

[23] S. Zhao, J. Zhang, and Y. Xu, "Performance monitoring of processes with multiple operating modes through multiple PLS models," *Journal of Process Control*, vol. 16, no. 7, pp. 763-772, August 2006.

[24] Z. Hu, L. Yin, Z. Chen, W. Gui, C. Yang, and X. Peng, "An efficient multi-PCA based on-line monitoring scheme for multi-stages imperial smelting process," *International Journal of Control, Automation, and Systems*, vol. 11, no. 2, pp. 317-324, March 2013.

[25] Y.-J. Yoo, "Fault detection method using multi-mode principal component analysis based on gaussian mixture model for sewage source heat pump system," *International Journal of Control, Automation, and Systems*, vol. 17, no. 8, pp. 2125-2134, August 2019.

[26] H. Chen, B. Jiang, and N. Lu, "A multi-mode incipient sensor fault detection and diagnosis method for electrical traction systems," *International Journal of Control, Automation, and Systems*, vol. 16, no. 4, pp. 1783-1793, August 2018.

[27] S. Zhang and C. Zhao, "Stationarity test and Bayesian monitoring strategy for fault detection in nonlinear multimode processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 168, pp. 45-61, September 2017.

[28] H. Kodamana, R. Raveendran, and B. Huang, "Mixtures of probabilistic PCA with common structure latent bases for process monitoring," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 838-846, March 2019.

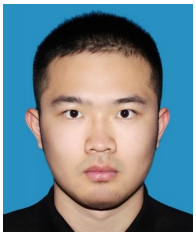
[29] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," *CoRR*, vol. abs/1607.00148, 2016. [Online]. Available: http://arxiv.org/abs/1607.00148

[30] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241-265, July 2018.

[31] W. Mao, J. Chen, X. Liang, and X. Zhang, "A new online detection approach for rolling bearing incipient fault via self-adaptive deep feature matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 443-456, February 2020.

[32] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 89-100, February 2018.

[33] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235-1270, July 2019.



Xinyao Xu received his B.Sc. degree in automation from Tianjin University, Tianjin, China, in 2018. He is currently pursuing a Ph.D. degree in control science and engineering at the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences (CASIA) and University of Chinese Academy of Sciences, Beijing, China. His research interests include industrial big data and fault diagnosis.

Fangbo Qin received his B.Sc. degree in automation from Beijing Jiaotong University, Beijing, China, in 2013. He received a Ph.D. degree in control science and engineering at CASIA and University of Chinese Academy of Sciences, Beijing, China, in 2019. He has been an Assistant Research Fellow at CASIA since 2019. His research interests include robot vision, robot manipulation, and deep learning.

Wenjun Zhao received his B.Sc. Degree in machinery manufacturing from Shenyang Ligong University, Shenyang, China, in 1985. He is currently a Senior Engineer at State Key Laboratory of Smart Manufacturing for Special Vehicles and Transmission System, Inner Mongolia First Machinery Group Co., Ltd. His research interests include digitization and intelligent manufacturing.

De Xu received his B.Sc. and M.Sc. degrees from the Shandong University of Technology, Jinan, China, in 1985 and 1990, respectively, and a Ph.D. degree from Zhejiang University, Hangzhou, China, in 2001, all in control science and engineering. He has been with CASIA since 2001. He is currently a Professor with the Research Center of Precision Sensing and Control, CASIA. His current research interests include robotics and automation such as visual measurement, visual control, intelligent control, welding seam tracking, visual positioning, microscopic vision, and micro-assembly.

Xingang Wang received his B.Sc. degree from Tianjin University, Tianjin, China, in 1995, and a Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China, in 2002. He is currently a Professor with the Research Center of Precision Sensing and Control, CASIA, Beijing, China. His current research interests include image processing and machine learning.

Xihao Yang received his B.Sc. Degree in Mechanical Engineering and Automation from Jilin University, Changchun, China, in 2012. He is currently an engineer in State Key Laboratory of Smart Manufacturing for Special Vehicles and Transmission System, Inner Mongolia First Machinery Group Co., Ltd. His research interests include automation system and intelligent manufacturing.