Fault Detection for Motor Drive Control System of Industrial Robots Using CNN-LSTM-based Observers

Tao Wang, Le Zhang, and Xuefei Wang

***Abstract*—The complex working conditions and nonlinear characteristics of the motor drive control system of industrial robots make it difficult to detect faults. In this paper, a deep learning-based observer, which combines the convolutional neural network (CNN) and the long short-term memory network (LSTM), is employed to approximate the nonlinear driving control system. CNN layers are introduced to extract dynamic features of the data, whereas LSTM layers perform time-sequential prediction of the target system. In terms of application, normal samples are fed into the observer to build an offline prediction model for the target system. The trained CNN-LSTM-based observer is then deployed along with the target system to estimate the system outputs. Online fault detection can be realized by analyzing the residuals. Finally, an application of the proposed fault detection method to a brushless DC motor drive system is given to verify the effectiveness of the proposed scheme. Simulation results indicate the impressive fault detection capability of the presented method for driving control systems of industrial robots.**

***Index Terms*—Fault detection, Motor drive control system, Deep learning, CNN-LSTM, Industrial robot.**

I. INTRODUCTION

INDUSTRIAL robots at present days play an important role in the manufacturing industry. According to a report from the ASci Corporation, in the first half of 2021, China's industrial robot production continued to increase, reaching 173,600 sets. As one of the core subsystems of industrial robots, the driving control system has always attracted great attention. However, anomalies triggered by abnormal driving control systems take up a large proportion compared to other fault cases. To ensure a reliable working status and save valuable time, fault detection and diagnosis (FDD) for industrial robot driving control systems is essential[1]-[4].

Manuscript received August 04, 2022. revised October 25, 2022; accepted December 02, 2022. Date of publication June 25, 2023; Date of current version January 17, 2023.

This work was supported in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 21KJA470007.

Tao Wang, Le Zhang are now with the School of Intelligent Equipment Engineering, Wuxi Taihu University, Jiangsu International Cooperation Laboratory of 6G Network and Big Data Intelligent Application, Wuxi, 214064, China (e-mail: jefewt@126.com, zhangl1@wxu.edu.cn).

Xuefei Wang is with the School of Intelligent Equipment Engineering, Wuxi Taihu University, Wuxi, 214064, China (e-mail:1083568696@qq.com).

(Corresponding Author: Tao Wang)

Digital Object Identifier 10.30941/CESTEMS.2023.00014

In general, fault detection methods for robot driving control systems can be divided into three main groups, including model-based, signal-based and data-driven methods[5]. Model-based fault detection methods are accomplished based on the comparison between the estimated and measured signals, which are highly dependent on accurate design of the system model. However, modeling for nonlinear driving control systems is a challenging task due to the complex mapping between model parameters and physical operation[6]. Signal-based methods perform fault detection tasks by analyzing the time-domain signals without modeling. The main disadvantages are time-consuming computation and noise susceptibility[7]. Recently, data-driven methods, which are also known as knowledge-based methods, have been extensively studied due to their powerful modeling ability without any prior known models or parameters[8]. Deep learning technology has now been extensively researched in data-driven fault detection. Based on the historical data set, a deep CNN model was designed to realize thruster failure detection of dynamically positioned vessels[9]. In Ref.[10], a CNN-based fault detection and classification method was developed in the semiconductor manufacturing process. Zhang *et al.* took full use of raw time series data to train a one-dimensional CNN model, which showed good performance in fault diagnosis[11]. In Ref.[12], wavelet analysis was used to transform the raw data into time-frequency images that were regarded as inputs of a CNN model, which turned FDD problems into image pattern recognition issues. While Wen *et al.* developed a signal-to-image data preprocessing method and proposed a CNN-based fault diagnosis scheme[13]. However, for all these FDD methods, it is expected to be reliable in fault classification when the training data include multiple fault patterns. That is to say, fault patterns that are not included in the raw data would hardly be classified by the trained model. Although CNN-based models are able to extract patterns of local trends of time-series data, they fail to capture the sequence pattern information for long temporal dependencies[14]. The LSTM is a deep learning method based on the recurrent neural network (RNN) that overcomes the limitations of the RNN structure by introducing memories and gates. An initial fault detection algorithm for rolling bearings based on the LSTM was designed and verified by experiment results[15]. In Ref.[16], the high impedance fault detection of solar photovoltaics based on the LSTM

approach was presented. The discrete wavelet transform method was taken to extract data features, while the LSTM was employed as an intelligent classifier to perform prediction tasks. Cheng et al. proposed an LSTM-based method to detect anomalies for satellite power systems, where the LSTM was conducted to establish a prediction model[17]. To detect vibration signal faults for the rotating machinery, the LSTM method was adopted as a classifier in Ref.[18]. Generally, the LSTM networks are suitable for working with temporal correlations, especially long temporal dependencies. One challenge to LSTM networks is how to improve their ability to explore potential correlations for noncontinuous data. Therefore, some literary research tried a hybrid method based on CNN and LSTM to exploit the complementary advantages of both techniques. Li *et al.* proposed a hybrid CNN-LSTM network for predicting the PM2.5 concentration[19]. Similarly, a hybrid CNN-LSTM framework was developed to forecast the short-term individual household electric load[14]. In Ref.[20] a deep learning model based on CNN and LSTM networks was established for fault classification of vibration data of a helicopter gearbox mock-up system. The final results indicated the promising performance of the CNN-LSTM networks. However, to our best knowledge, the hybrid deep learning framework employed for fault detection of industrial robot driving control systems has rarely been found in the literature.

In this article, a hybrid deep learning-based observer is presented to estimate the outputs of a nonlinear industrial robot driving control system by combining CNN and LSTM networks. Herein the fault detection is treated as a time-series prediction problem. At first, an offline prediction model is built by training the CNN-LSTM hybrid network with preprocessing data without faults. CNN layers utilize convolution, pooling, and padding operations to extract the internal representation of time series data and capture important attributes, which are subsequently learned by LSTM layers to identify short-term and long-term dependencies. The trained model then works as an online observer to estimate the outputs of the driving control system. As the observer is trained with normal data, the residuals between the estimated outputs and the measured outputs contain related information when the driving control system occurs faults. Thus, a fault detection scheme for industrial robot driving control systems is designed based on the evaluation of the residuals.

To clearly explain the fault detection strategy, a brief review of CNN and LSTM networks is conducted to develop a nonlinear observer with a hybrid deep learning framework in Section II, which lays the foundation for the fault detection scheme presented in Section III. Section IV presents the case validation to evaluate the proposed method for the industrial robot driving control system. The conclusion and future jobs are described in Section V.

II. Nonlinear Observer with Hybrid Framework

Thanks to the powerful approximation capability to complex nonlinear functions, deep learning algorithms have been extensively applied to the FDD research field[21]-[24]. Owing to their promising performance, CNN and LSTM networks probably take an overwhelming position in deep learning study and application. In this section, a nonlinear observer is established based on hybrid CNN and LSTM networks to estimate the outputs of the industrial robot driving control system. Fig. 1 depicts the framework of the nonlinear observer. It can be seen from Fig.1 that the control signals and measurement signals are fed into the hybrid deep learning network. With the help of normal data, the observer can be trained to extract features of the driving control system and establish the nonlinear mapping between reference signals and system output. To improve the efficiency and precision of the proposed observer, the implementation is described in the following steps.

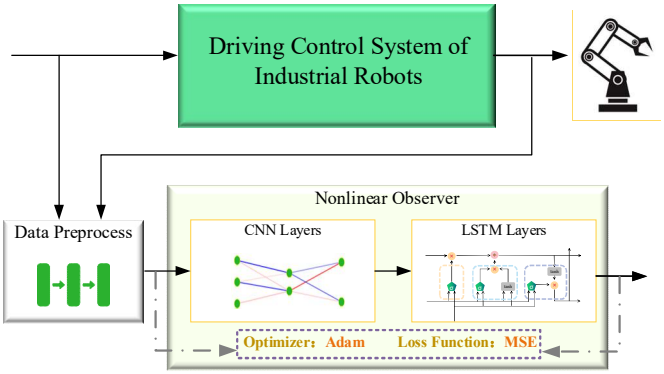

Fig. 1. Framework of the nonlinear observer based on the hybrid neural networks.

A. Data Preprocessing

As mentioned above, one of the main tasks of the observer is to exploit the hidden mapping between inputs and outputs. However, the complex working condition of the driving control system indicates the nonlinear properties of the mapping denoted as (1), where $\boldsymbol{u}$ stands for the input vector, $\boldsymbol{y}$ is the output vector, while $f(\cdot)$ represents the nonlinear mapping function.

$$\boldsymbol{y} = f(\boldsymbol{u}) \tag{1}$$

In this work, the raw data acquired from the driving control system contains motor speed v, electromagnetic torque T_e, load torque T_L, phase current I_a, I_b, I_c and control signal Ctl. The motor speed would vary considerably across different loads as well as control laws. Therefore, the speed observer based on deep learning networks should be designed to excavate the expression between the speed and the related parameters, which is shown as (2),

$$\hat{v}_k = f\left(I_{ak\text{-}1}, T_{ek\text{-}1}, T_{Lk\text{-}1}, Ctl_{k\text{-}1}, v_{k-1}\right) \tag{2}$$

where subscript k means the time step, $\hat{}$ stands for the estimated output. Given the ability of the LSTM to extract sequence pattern information as well as long-term dependencies, the sample of motor speed at the previous time step is strongly associated with the current prediction. This is the reason that v_{k-1} is adopted as an input parameter in (2). Moreover, the Min-Max normalization technique conducted as (3), is used to deal with the problem of dimensional inconsistency.

$$x_{inorm} = \frac{x_i - x_{min}}{x_{min} - x_{max}} \tag{3}$$

$x_{\min}$, $x_{\max}$ are the minimum and maximum elements of the vector $\boldsymbol{x}$ respectively, while x_{inorm} is the normalized result within the range of 0 to 1. To form the training data that can be handled by the CNN and LSTM models, each raw data sequence is reshaped to a suitable size matrix $N \times m$. Then another $N \times m$ continuous sample is formed as a new matrix in an overlapping manner as shown in Fig. 2. Furthermore, 10% of the raw data are taken as the testing data by repeating the above processing method.

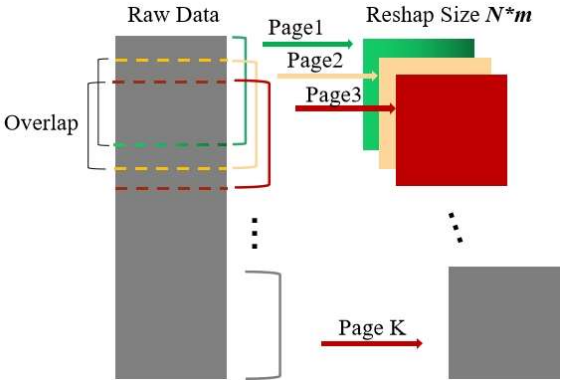

Fig. 2. Data Segmentation.

B. Model Training

As one of the core modules of the nonlinear observer, the hybrid deep learning framework based on CNN and LSTM networks takes the responsibility of modeling the driving control system. Fig. 3 shows the framework of the proposed CNN-LSTM.

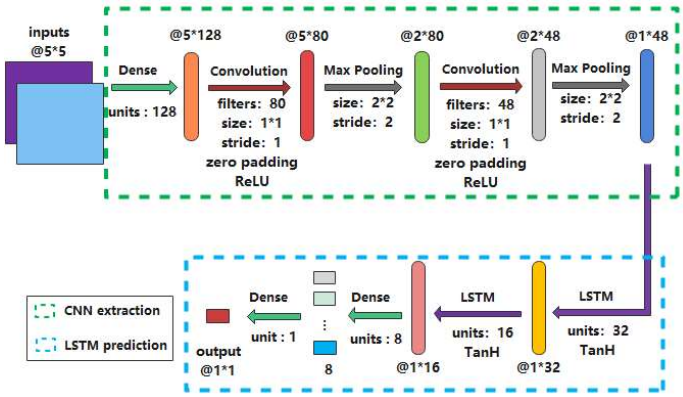

Fig. 3. Architecture of the CNN-LSTM Network.

1) Feature Extraction by CNN

CNN is always used to extract spatial features of the raw data considering its multiple filter structure. It can be seen from Fig. 3 that the CNN designed in this paper contains one dense (or fully connected) layer, two convolution layers, and two pooling layers.

The dense layer is designed with 128 units to reshape the inputs for convolution operations. By implementing function (4), the dense layer preliminarily establishes the feature maps.

$$\mathbf{y} = \boldsymbol{\omega}\mathbf{x} + \mathbf{b} \tag{4}$$

where $\boldsymbol{\omega}$ and $\mathbf{b}$ represent the weights and offset, respectively.

The convolution layer extracts essential features by sliding kernel filters over the input data. Furthermore, for a filter, it shares the same parameters during convolution operation. Meanwhile, the zero-padding method is introduced to prevent dimension loss problems. To enhance the capacity of the model to learn complex structures and nonlinear characteristics, the rectified linear unit (ReLU) is taken as the activation function.

$$\mathbf{h} = \max(0, \mathbf{x} \otimes \boldsymbol{\omega} + \mathbf{b}) \tag{5}$$

where $\otimes$ stands for convolution operation, $\boldsymbol{\omega}$ means the weights of the kernel filter, $\mathbf{b}$ is the offset and $\max(0, \bullet)$ indicates the ReLU activation function. The derivative of the ReLU function keeps constant 1 for a positive input, which effectively deals with the problem of vanishing and exploding gradients.

The pooling layer carries out the down-sampling operation on the feature map results of the convolution layer. The max-pooling operation enhances the local vision field of the model by aggregating similar features with the maximum value. Moreover, by sliding a sized 2*2 filter over the input, a reduced-dimensional feature map is obtained in this layer. The number of neuron parameters to be trained is significantly reduced, which in turn prevents the model training from overfitting.

2) Time Sequential Prediction by LSTM

Although CNN has excellent performance in spatial feature extraction, the time series features of the raw data are so temporal correlated and implicit that CNN can hardly capture them[25]. Consequently, the output of CNN is sequentially fed into the LSTM network to establish temporal-spatial correlations. The LSTM structure shown in Fig. 4 indicates that memory units learn time dependencies information by introducing three gates, i.e., forget gate, input gate, and output gate. As a core component of a memory unit, the memory cell takes the responsibility for state updates and transitions.

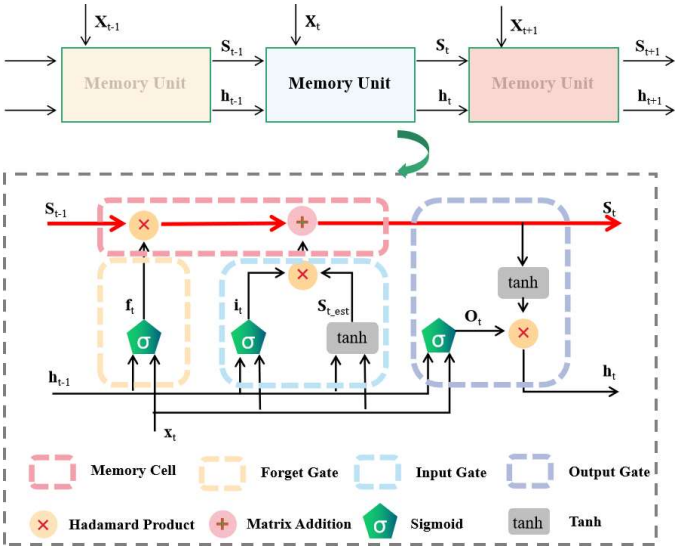

Fig. 4. Structure of LSTM.

The forget gate as the first gate performs the function of discarding redundant information by using the Sigmoid.

$$\mathbf{f}_t = \sigma(\boldsymbol{\omega}_\mathbf{f} \cdot [h_{t-1}, x_t] + \mathbf{b}_\mathbf{f}) \tag{6}$$

where σ stands for the Sigmoid function and generates a constant between 0 and 1 for each element in the state parameter $\mathbf{S}_{t-1}$. 0 means all information discarded and 1 means all information reserved. $\boldsymbol{\omega}_\mathbf{f}$, $\mathbf{b}_\mathbf{f}$ are the weights and the offset respectively.

The input gate works to select the latest information to be updated with the help of the Sigmoid and Tanh functions. The Tanh is taken to create a candidate vector $\mathbf{S}_{t\_est}$ that acts together with the Sigmoid output $\mathbf{i}_t$ to add the latest information into cell states.

$$\begin{cases} \mathbf{i}_t = \sigma(\boldsymbol{\omega}_\mathbf{i} \cdot [h_{t-1}, x_t] + \mathbf{b}_\mathbf{i}) \\ \mathbf{S}_{t\_est} = \tanh(\boldsymbol{\omega}_\mathbf{S} \cdot [h_{t-1}, x_t] + \mathbf{b}_\mathbf{S}) \end{cases} \tag{7}$$

The memory cell takes the Hadamard product and matrix addition to perform the discarding and update operations (8). The $\mathbf{S}_t$ is the latest cell state that contains previous time-related information. '$\circ$' denotes the Hadamard product.

$$\mathbf{S}_t = \mathbf{f}_t \circ \mathbf{S}_{t-1} + \mathbf{i}_t \circ \mathbf{S}_{t\_est} \tag{8}$$

Using Sigmoid and Tanh, the output gate gives the final filtered output that is sequentially linked to the next memory unit.

$$\mathbf{h}_t = \sigma(\boldsymbol{\omega}_\mathbf{o} \cdot [h_{t-1}, x_t] + \mathbf{b}_\mathbf{o}) \circ \tanh(\mathbf{S}_t) \tag{9}$$

Thanks to the remarkable structure of LSTM, the memory units can find the complex time series features in both the short and long term. It is depicted in Fig. 3 that the extracted features of the CNN module are trained by two LSTM layers to establish temporal-spatial correlations. Moreover, two dense layers are finally employed to figure out the probability of the expected item.

3) Combination Principle of CNN and LSTM

According to the hybrid CNN-LSTM framework shown in Fig. 3, there are 9 hidden layers to be trained. After the second max pooling operation in the CNN part, a sized 1*48 vector as the linkage is fed into the LSTM network. For effective model fitting of the combination network, those input data are split into the training set, validation set and testing set. While the mean square error (MSE) loss function, denoted as (10), is used to monitor the model validation loss.

$$J_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{10}$$

where N is the total number of samples. y_i stands for the system output. $\hat{y}_i = f(\boldsymbol{\omega}, \mathbf{b})$ is the predicted result, a function of weights and offsets. Thus, weights and offsets updates are conducted as (11).

$$\begin{cases} \boldsymbol{\omega}_{i+1} = \boldsymbol{\omega}_i - \eta \dfrac{\partial J_{MSE}}{\partial \boldsymbol{\omega}} \\ \mathbf{b}_{i+1} = \mathbf{b}_i - \eta \dfrac{\partial J_{MSE}}{\partial \mathbf{b}} \end{cases} \tag{11}$$

where $\dfrac{\partial J}{\partial \cdot}$ is the gradient express, η is the learning rate to set the step size for parameter update. A well-known optimizer, *Adam* is taken to optimize the learning rate. By repeating epochs, parameters such as weights of each layer will be well adjusted. Prediction and evaluation will be achieved on the test data by loading the last trained model.

III. Online Fault Detection Scheme

In this section, an online fault detection scheme for industrial robot driving control systems is developed based on the nonlinear observer proposed in Section II.

A. Problem Statement and Principle

On the basis of the aforementioned structure, the control and measured signals are fed into the CNN-LSTM network for offline training, see as Fig. 5. With the help of the *Adam* optimizer, the nonlinear observer is trained to be consistent with the target system by minimizing the MSE loss function. Then, the trained CNN-LSTM observer is deployed along with the target system to carry out the prediction task. When anomalies or dynamic changes occur, actual outputs of the target system act in an abnormal manner. Given that the CNN-LSTM observer is trained by the normal data without anomalies, the prediction indicates outputs of the target system under normal operation. Thus, to implement fault detection, two problems have to be solved,

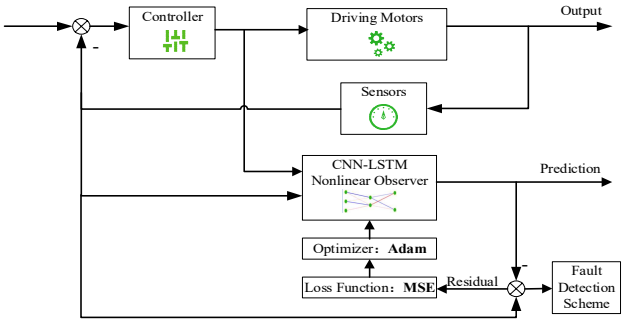

Fig. 5. Fault Detection Diagram Based on the Proposed Nonlinear Observer.

- As a crucial step of the fault detection method, residual generation has a profound impact on the final detection results. Therefore, building a residual model for the target system by using the CNN-LSTM network will be of great importance.
- For the collected residual signals, establishing a residual evaluation method is an effective way of indicating anomalies. Hence the evaluation criteria should be figured out for the decision of fault detection.

For the purpose of detecting anomalies, here an online fault detection scheme is proposed as Fig. 6 in consideration of the above two problems. In the phase of offline model training, data of regular operation are pretreated for CNN-LSTM network training to build an offline model. While in the phase of online fault detection, the trained model is loaded as an observer to identify and predict outputs of the target system. Finally, residual evaluation is conducted to accomplish fault detection tasks.

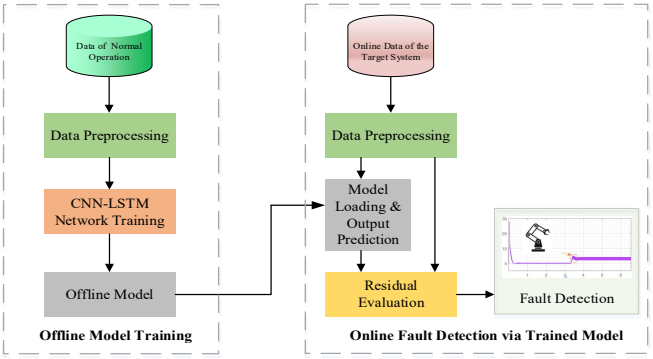

Fig. 6. Online Fault Detection Scheme.

B. Residual Modeling for the Target System

It can be concluded from the above description that the residual signals between the actual measured output and the predicted value are valuable as a diagnostic as they suggest failure information that could be incorporated into the target system.

Based on the scheme shown in Fig. 6, to establish the residual sequence, an offline trained CNN-LSTM network has to be deployed with the target system. The observer is taken to make online predictions. $\widehat{Y}_k$ denotes the predicted value. While Y_k is the actual output of the target system. The residual sequence is denoted as $\boldsymbol{\varepsilon} = Y_k - \widehat{Y}_k$. As mentioned before, the CNN-LSTM observer is offline trained to be consistent with the target system that is under normal operation. When the target system works in some abnormal way, the residual sequence will act abnormally. Therefore, anomalies detection can be implemented by evaluating the residual sequence, which is illustrated in detail on the following section.

C. Residual Evaluation

For the purpose of anti-interference performance, the PSNR (peak signal noise ratio) index is applied to explore the anomalies in the residual sequence.

1) Z-Score Normalization

Since the offline model is trained under fault-free conditions, the residual sequence converges to a range of stable minimum values when the target system runs well. The residual sequence can be written as a vector $\boldsymbol{\varepsilon} = \left[e_1, e_2 \cdots, e_k\right]^T$. Detecting anomalies under the normal residual distribution is sometimes difficult, especially when the external disturbances are not neglected. Consequently, Z-score normalization is utilized to score and rank outliers for the residual vector depending on how far the value of the sample is from the mean. Z-score normalization for $\boldsymbol{\varepsilon}$ is expressed as

$$\boldsymbol{\varepsilon}_{Z-score} = \left\| \left[\boldsymbol{\varepsilon} - \text{mean}(\boldsymbol{\varepsilon}) \cdot \boldsymbol{I}_{N\times 1}\right] ./ \left[\text{std}(\boldsymbol{\varepsilon}) \cdot \boldsymbol{I}_{N\times 1}\right] \right| \quad (12)$$

where mean($\boldsymbol{\varepsilon}$) and stand($\boldsymbol{\varepsilon}$) is the mean and standard deviation of the vector $\boldsymbol{\varepsilon}$ respectively, and $\boldsymbol{I}_{N\times 1}$ is a $N\times 1$ vector with elements of 1. 'A ./ B' is an operation that divides each element of A by the corresponding element of B. Here, the absolute value of each element of the $\boldsymbol{\varepsilon}_{Z-score}$ is calculated to indicate the outliers in the residual vector when the target driving system is working abnormally.

2) PSNR Index

Noises are not eliminated in practical applications. That is to say, Z-Score normalization sometimes presents unreliable ranking results in case of noise disturbance. Therefore, the PSNR index, shown as (13), is taken based $\boldsymbol{\varepsilon}_{Z-score}$ to indicate the anomalies.

$$PSNR = \left[\boldsymbol{\varepsilon}_{Z-score} - \text{mean}(\boldsymbol{\varepsilon}_{Z-score}) \cdot \boldsymbol{I}_{N\times 1}\right] \times \left[\boldsymbol{\varepsilon}_{Z-score} - \text{mean}(\boldsymbol{\varepsilon}_{Z-score}) \cdot \boldsymbol{I}_{N\times 1}\right]^T ./ \left[\text{std}^2(\boldsymbol{\varepsilon}_{Z-score}) \cdot \boldsymbol{I}_{N\times 1}\right] \quad (13)$$

When the target driving system with noise runs normally, the PSNR index stays within a range of amplitude. However, faults and anomalies lead to abnormal performance for the PSNR index. Consequently, the fault detection is finally described as (14)

$$\begin{cases} PSNR \le T & fault-free \\ PSNR > T & faulty \end{cases} \quad (14)$$

T is the designed threshold for the decision rule. It is highly recommended that T is set as the mean value of PSNR elements from the fault-free training system.

IV. Cases Study

In this paper, a kind of brushless DC motor is taken as the driving system of industrial robots. In this section, a three-phase motor rated 1kW, 500Vdc, 3000rpm is fed by a six-step voltage inverter. The PI speed regulator is used to control the DC voltage, which is inverted to the three-phase voltage source of the stator windings. The stator phase resistance Rs is 2.875Ω, the stator phase inductance Ls is 0.0085H, the flux linkage is set to 0.175Wb, while the load torque is first set to 0 and steps to 1.5N.m at 1 second.

A. Nonlinear Observer Model Training

TABLE I
CONFIGURATION AND PARAMETER SETTINGS OF THE MODEL

Parameter	Setting
Optimizer	Adam
Loss Function	Mean Square Error (MSE)
Default Learning Rate	0.001
Batch Size	128
Epoch	90
Adjustment	Minimum learning rate: 1e-5 Monitor: validation loss
Total Hidden Layers	9 (seen in Fig. 3)
Split Ratio	80% training, 20% validation
Network Trainable Parameters	28,625

To verify the validity of the proposed CNN-LSTM observer, more than 133 thousand groups of the selected signals (seen formula (2)) are collected as a raw dataset in this section. Moreover, 119,995 reshaped matrices with 5x5 dimension are normalized to feed into the CNN-LSTM network for model training. The configuration and parameter settings of the proposed model are provided in Table I.

According to Table I, 20% of the inputs are separated from the raw data to validate the model training performance. By 90 epochs, the loss curve and mean absolute error (MAE) curve for training data and validation data are respectively shown in Fig.

7 and Fig. 8. For the proposed CNN-LSTM network training, it can be seen that the top-left loss curve in Fig. 7 and the top-left MAE curve in Fig. 8 of training data are both convergent, while the corresponding loss curves of validation data, see CNN-LSTM_Val_Loss in Fig. 7 and CNN-LSTM_Val_MAE in Fig. 8, stay in a small range. It means that the CNN-LSTM model is trained to be consistent with the target system. For the sake of comparison, the loss curves and MAE curves for training data and validation data based on CNN and LSTM network are also presented in Fig. 7 and Fig. 8. Obviously, loss curves (see CNN_Loss and CNN_Val_Loss in Fig. 7) and MAE curves (see CNN MAE and CNN_Val_MAE in Fig. 8) based on CNN network compares unfavourably with other ones. Moreover, the loss (see LSTM_Loss and LSTM_Val_Loss in Fig. 7) and MAE (see LSTM_MAE and LSTM_Val_MAE in Fig. 8) results based on LSTM network are close to those based

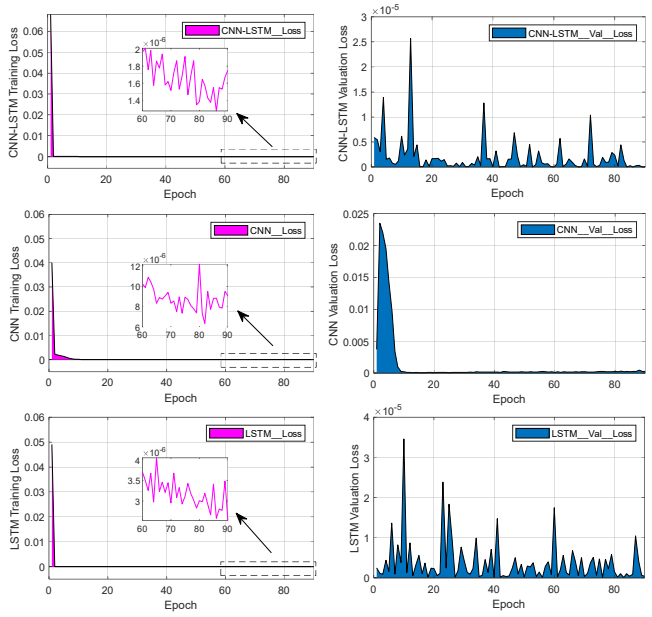

Fig. 7. Loss Curves of the Training and Validation Sample.

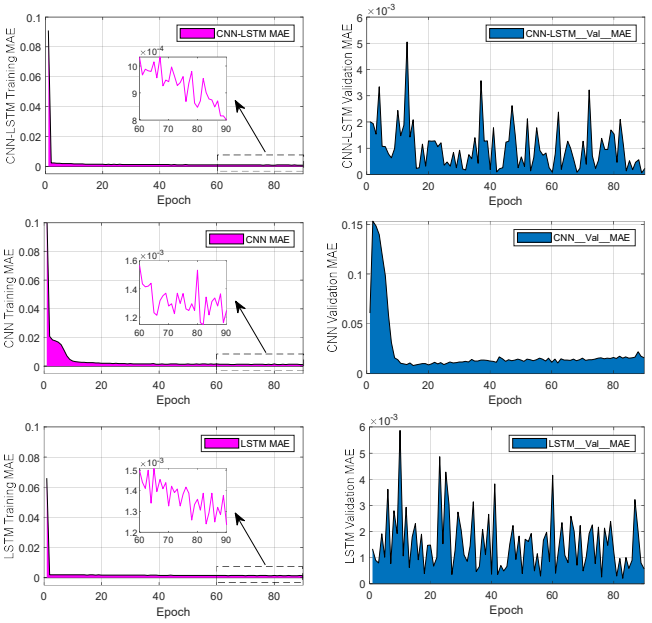

Fig. 8. MAE Curves of the Training and Validation Sample.

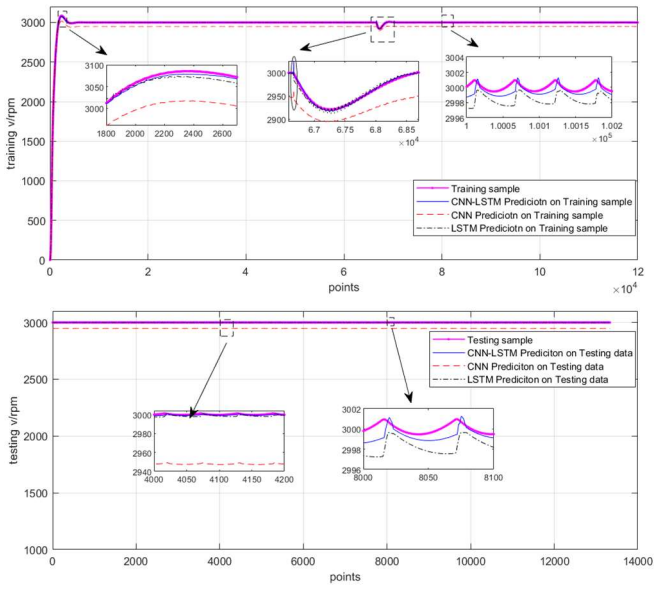

Fig. 9. Model Prediction on the Training and Testing Sample.

on CNN-LSTM network except the convergence results. The proposed CNN-LSTM model holds more stable and convergent loss and MAE curves compared to the LSTM network.

As mentioned in Section II, the rotating speed v is the output of the model. To illustrate the prediction performance of the proposed network, the prediction results based on CNN, LSTM and CNN-LSTM models are respectively presented in Fig. 9 for comparison.

For the target system with a step reference, not only the 119,995 raw training datasets, but also another 13,333 groups of reshaped testing data are utilized for the model prediction. Here, the graphs at the top of Fig. 9 are the prediction results of the training datasets. Obviously, CNN-based prediction is unfavourable in consideration of the undesirable prediction error compared with that of other two models. Though the LSTM model puts up a good performance, CNN-LSTM model holds more robustness and accurate prediction, which is indicated by the local subgraphs in Fig. 9. In addition, the prediction results on testing datasets, see graphs at the bottom of Fig. 9, demonstrate the excellent performance of the CNN-LSTM model as well.

The prediction results are abnormal when the load torque is working on the target system, which is seen as the points marked by an ellipse at the top of Fig. 9. That is because the model has not learned the dynamic changes in system parameters caused by the additional torque at that moment. Nevertheless, the network quickly optimizes its parameters to follow the changed target system.

B. Fault Detection via the Trained Observer

Industrial robots always work under complex conditions for a long time. As the core component, the driving control system of an industrial robot sometimes encounters anomalies. Here in this section, two common types of fault are taken to validate the proposed fault detection scheme. One is the constant sensor bias fault because of the sensitivity and resistance anomalies

caused by a change in working temperature. Another is the controller output stuck fault caused by an electrical anomaly of the DC source.

1) Constant Sensor Bias Fault

Suppose that the driving control system occurs a constant sensor bias fault since 0.4 seconds. The measured rotation speed feedback is always 200 rpm greater than the actual speed output. For the sake of comparison, the rotating speed predictions based on the proposed CNN-LSTM observer and other deep learning methods, i.e. CNN and LSTM, are presented in the first graph in Fig. 10. The other three graphs are the corresponding PSNR curves that indicate the fault detection results based on different methods. Meanwhile, Table II presents the model prediction performance and PSNR statistics, false alarm ratio (FR), and missing alarm ratio (MR) of different models.

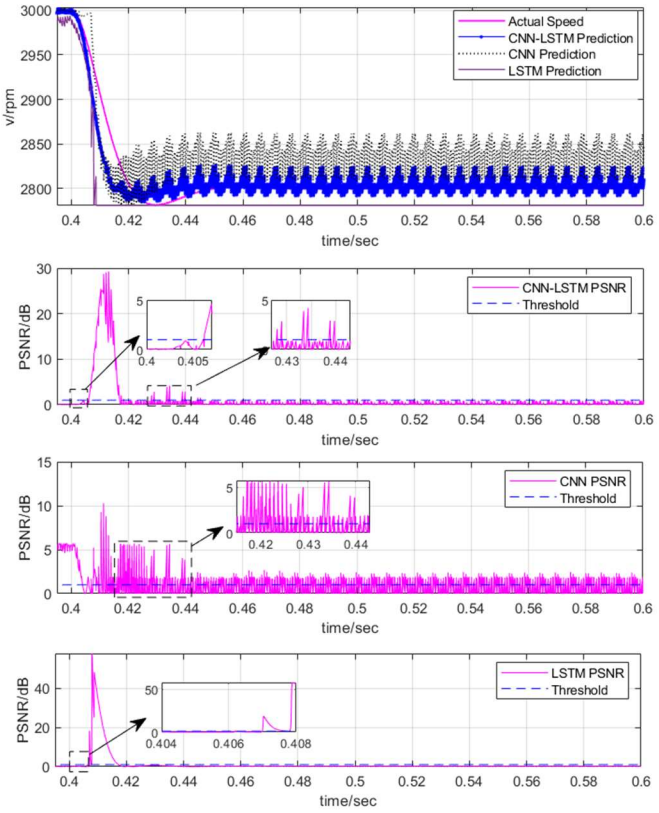

Fig. 10. Constant Bias Fault Detection Based on Different Methods.

TABLE II
MSE, MAE AND PSNR RESULTS FOR CONSTANT BIAS FAULT

Model	MSE	MAE	PSNR Statistics	
			FR	MR
CNN-LSTM	0.0045	0.0444	0	0.0921
CNN	0.0337	0.1674	0.2207	0.2105
LSTM	0.0161	0.0968	0	0.2850

The superiority of the proposed CNN-LSTM method is demonstrated by the graphs in Fig. 10 and the results in Table II.

- Thanks to the PI regulator, the faulty closed-loop driving control system can quickly resume a stable state. Moreover, for the target system, the proposed CNN-LSTM method performs much better than CNN and LSTM methods in rotation speed prediction, as its smaller MSE and MAE.
- Although the LSTM model holds lower MSE and MAE than those of the CNN model, the unappealing prediction graph in Fig. 10, marked as *LSTM Prediction*, deviates far away from the actual output curve.
- There is no false alarm for this fault based on the proposed fault detection scheme. In addition, the 9.21% MR is much smaller than that of CNN and LSTM, which proves the better performance of the proposed scheme in detecting anomalies in view of the dropped FR and MR indexes.

2) Controller Output Stuck Fault

Suppose that the controller output used to control the DC bus voltage is stuck at 350V since 0.4 seconds. Similar to the detection process of the constant sensor bias fault, the rotating speed prediction and the PSNR index based on different methods are shown in Fig. 11. The corresponding results are illustrated in Table III.

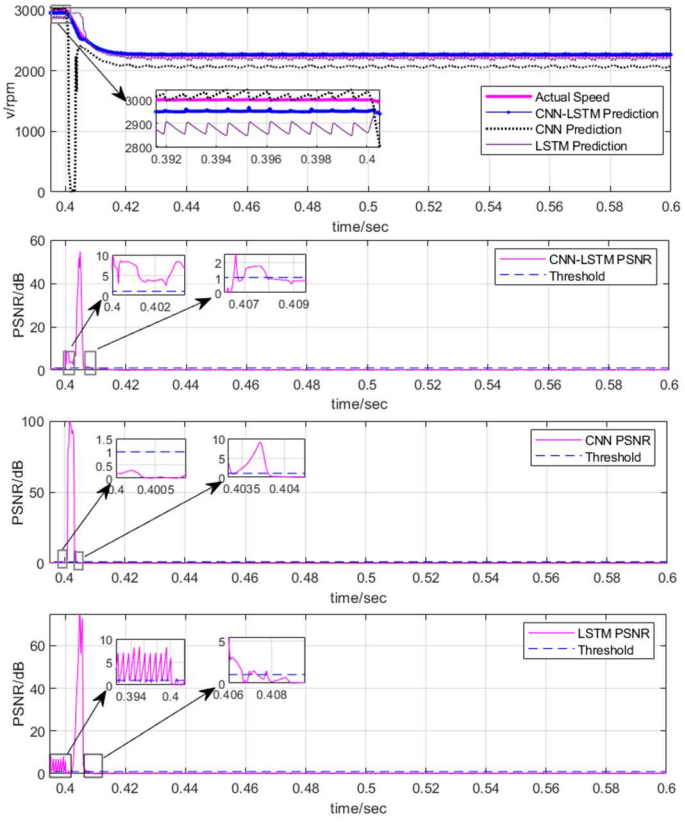

Fig. 11. Stuck Fault Detection Based on Different Methods.

TABLE III
MSE, MAE AND PSNR RESULTS FOR STUCK FAULT

Model	MSE	MAE	PSNR Statistics	
			FR	MR
CNN-LSTM	$4.6279*10^{-5}$	0.006	0.0009	0.1002
CNN	0.0121	0.0663	0	0.6427
LSTM	$4.2924*10^{-4}$	0.0175	0.0366	0.3403

According to the comparison, the proposed CNN-LSTM network shows better performance than other approaches in model prediction and fault detection. It can be concluded from the graphs and the statistics that

- The faulty system finally holds a new state when the controller is stuck. All three models show promising prediction indexes, MAE and MSE. In comparison, the proposed CNN-LSTM network is slightly better than others.
- As in the first graph in Fig. 11, the proposed CNN-LSTM method shows an improved prediction curve compared to other predictions. Additionally, the prediction based on the CNN model fluctuates in an irregular way when the anomaly occurs.
- Regarding the PSNR statistics in Table III, the FR and MR of the CNN-LSTM network show significant advantages compared to other methods, especially the great improvement in the missing fault alarm ratio.

In summary, for the driving control system, the proposed hybrid CNN-LSTM model shows more promising performance in output prediction than CNN and LSTM. Furthermore, the above detection results for different fault patterns indicate that the hybrid framework makes a significant improvement in feature extraction and time-sequential prediction in comparison with the basic CNN network and traditional LSTM structure.

V. Conclusion

A CNN-LSTM hybrid deep learning network is designed as a nonlinear observer for driving control systems of industrial robots in this paper. With the assistance of the strengths of CNN and LSTM, the proposed method enhances the modeling performance of the target system by connecting an LSTM network to a CNN network. Then anomaly detection of two common fault patterns can be realized via analyzing the residuals. The superiority of this approach can be concluded as three aspects compared with conventional methods. Firstly, the hybrid framework focuses on both spatial features and temporal correlation of the target system, which is demonstrated by the predictions for different cases. Secondly, the CNN-LSTM network is equipped with a powerful approximate ability to nonlinear systems, which ensures the accurate evaluation of the target system based on the trained model. Moreover, the fault detection scheme based on residuals utilizes Z-Score and PSNR to enhance the detection efficiency. The case study results reveal that the designed CNN-LSTM model is an effective and promising way of fault detection for the target system.

This study focuses only on the prediction of the rotation speed. In future work, estimation for other parameters, such as phase current and electromagnetic torque can be applied to make a weight-based decision. Additionally, the validation of the proposed scheme for different fault patterns on a semi-physical platform will be conducted in the future.

References

[1] Erjun, Y., H. Mingliang, and L. peng, "Design of high-precision and high-reliability aircraft stepper motor drive servo control system," in *Proc. IET Conference*, 2021, pp. 22-27

[2] Ghosh, S.S., S. Chattopadhyay, and A, "Das Harmonics Monitoring for Winding Fault Diagnosis of Brushless DC motor," in *Proc. IET Conference*, 2021, pp. 164-169.

[3] Zhang, X., et al. "Research on the control strategy of five-phase fault-tolerant servo system for aerospace," in *Proc. IET Conference*, 2021, pp. 1090-1094.

[4] Y. Zhao, X. Liu, H. Yu, and J. Yu, "Model-free adaptive discrete-time integral terminal sliding mode control for PMSM drive system with disturbance observer," *IET Electric Power Applications*, vol. 63, no. 10, pp. 1756-1765, Oct. 2020.

[5] Webert, Heiko, Tamara Döß, Lukas Kaupp, and Stephan Simons, "Fault Handling in Industry 4.0: Definition, Process and Applications," *Sensors*, vol. 22, no. 6, pp. 2205, 2022.

[6] Purbowaskito, Widagdo, Po-Yan Wu, and Chen-Yang Lan, "Permanent Magnet Synchronous Motor Driving Mechanical Transmission Fault Detection and Identification: A Model-Based Diagnosis Approach," *Electronics*, vol. 11, no. 9, pp. 1356, 2022.

[7] H. Badihi, Y. Zhang, B. Jiang, P. Pillay and S. Rakheja, "A Comprehensive Review on Signal-Based and Model-Based Condition Monitoring of Wind Turbines: Fault Diagnosis and Lifetime Prognosis," in *Proc. of the IEEE*, vol. 110, no. 6, pp. 754-806, June 2022, doi: 10.1109/JPROC.2022.3171691.

[8] H. Chen, B. Jiang, S. X. Ding and B. Huang, "Data-Driven Fault Diagnosis for Traction Systems in High-Speed Trains: A Survey, Challenges, and Perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1700-1716, March 2022, doi: 10.1109/TITS.2020.3029946.

[9] P. Han, G. Li, R. Skulstad, S. Skjong and H. Zhang, "A Deep Learning Approach to Detect and Isolate Thruster Failures for Dynamically Positioned Vessels Using Motion Data," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-11, 2021, Art no. 3501511, doi: 10.1109/TIM.2020.3016413.

[10] K. B. Lee, S. Cheon and C. O. Kim, "A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 2, pp. 135-142, May 2017, doi: 10.1109/TSM.2017.2676245.

[11] W. Zhang, C. Li, G. Peng, Z. Zhang, "A Deep Convolutional Neural Network with New Training Methods for Bearing Fault Diagnosis under Noisy Environment and Different Working Load," *Mechanical Systems and Signal Processing*, vol. 100, pp. 439-453, Feb. 2018..

[12] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, Jan. 2019.

[13] L. Wen, X. Li, L. Gao and Y. Zhang, "A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990-5998, July 2018, doi: 10.1109/TIE.2017.2774777.

[14] M. Alhussein, K. Aurangzeb and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," in *IEEE Access*, vol. 8, pp. 180544-180557, 2020, doi: 10.1109/ACCESS.2020.3028281.

[15] H. Shi, L. Guo, S. Tan and X. Bai, "Rolling Bearing Initial Fault Detection Using Long Short-Term Memory Recurrent Network," in *IEEE Access*, vol. 7, pp. 171559-171569, 2019, doi: 10.1109/ACCESS.2019.2954091.

[16] V. Veerasamy et al., "LSTM Recurrent Neural Network Classifier for High Impedance Fault Detection in Solar PV Integrated Power System," in *IEEE Access*, vol. 9, pp. 32672-32687, 2021, doi: 10.1109/ACCESS.2021.3060800.

[17] F. Cheng et al., "Research on Satellite Power Anomaly Detection Method Based on LSTM," *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, 2021, pp. 706-710, doi: 10.1109/ICPECA51329.2021.9362601.

[18] Z. K. Abdul, A. K. Al-Talabani and D. O. Ramadan, "A Hybrid Temporal Feature for Gear Fault Diagnosis Using the Long Short Term Memory," in *IEEE Sensors Journal*, vol. 20, no. 23, pp. 14444-14452, Dec.1, 2020, doi: 10.1109/JSEN.2020.3007262.

[19] T. Li, M. Hua and X. Wu, "A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5)," in *IEEE Access*, vol. 8, pp. 26933-26940, 2020, doi: 10.1109/ACCESS.2020.2971348.

[20] T. Haj Mohamad, A. Abbasi, E. Kim and C. Nataraj, "Application of

Deep CNN-LSTM Network to Gear Fault Diagnostics," *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2021, pp. 1-6, doi: 10.1109/ICPHM51084.2021.9486591.

[21] A. M. Erfanian and A. Ramezani, "Using Deep Learning Network for Fault Detection in UAV," *2022 8th International Conference on Control, Instrumentation and Automation (ICCIA)*, 2022, pp. 1-5, doi: 10.1109/ICCIA54998.2022.9737206.

[22] Y. L. Guo, G. X. Wu, X. L. Liu and X. L. Xu, "Review of Fault Diagnosis Methods for Rotating Machinery Based on Deep Learning," in *Proc. of IET Conference*, 2021, pp. 175-180, doi: 10.1049/icp.2021.1316.

[23] Stefenon, S.F., et al. "Fault detection in insulators based on ultrasonic signal processing using a hybrid deep learning technique," *IET Science, Measurement & Technology*, vol. 14, no. 10, pp. 953-961, Dec. 2020, doi: 10.1049/iet-smt.2020.0083.

[24] Zhao, K. and L. Shi, "Application of deep neural networks for fault diagnosis in a hybrid AC/DC power grid," in *Proc. of IET Conference*, pp. 903-908, 2021.

[25] Z. Zhan, W. Chen, X. Wu, et al. "LSTM Network: A Deep Learning Approach for Short-term Traffic Forcast," *IET Intelligent Transport System*, vol. 11, no. 2, pp. 68-75, 2017.

Tao Wang was born in Jiangsu, China, in 1990. He received his B.E. degree in automation from Changzhou University, Jiangsu, China and M.S. degree in control theory and engineering from Nanjing University of Aeronautics and Astronautics, Jiangsu, China.

From 2015 to 2018, he was with ZTE Corporation, where he was an international solution manager. From 2018, he has been with Wuxi Taihu University. He is currently a lecturer of automation engineering. His current research interests include data-driven fault detection and diagnosis of industrial robots, the application of machine vision on manufacturing industry.

Le Zhang was born in Jiangsu, China, in 1980. He received his B.S., M.S., and Ph.D. degrees in electrical engineering from Nanjing University of Aeronautics and Astronautics, Jiangsu, China.

He has been with Wuxi Taihu University. He is currently an associate professor of automation engineering. His current research interests include smart grid, permanent magnet synchronous motor control.

Xuefei Wang was born in Jiangsu, China, in 1990. She received her B.E. in Electric and Electronic Engineering from Nanjing Normal University, Jiangsu, China and M.S degree in Power Distribution from Newcastle University, UK.

From 2016, she has been with Wuxi Taihu University. She is currently a lecturer of automation engineering. Her current research mainly in the field of digital electronic technology and FPGA.