



Chinese Society of Aeronautics and Astronautics  
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn  
www.sciencedirect.com



## FULL LENGTH ARTICLE

# An attention graph stacked autoencoder for anomaly detection of electro-mechanical actuator using spatio-temporal multivariate signals

Jianyu WANG, Heng ZHANG, Qiang MIAO \*

College of Electrical Engineering, Sichuan University, Chengdu 610065, China

Received 7 October 2023; revised 3 November 2023; accepted 21 February 2024

### KEYWORDS

Anomaly detection;  
Spatio-temporal information;  
Multivariate time series signals;  
Attention graph convolution;  
Stacked autoencoder

**Abstract** Health monitoring of electro-mechanical actuator (EMA) is critical to ensure the security of airplanes. It is difficult or even impossible to collect enough labeled failure or degradation data from actual EMA. The autoencoder based on reconstruction loss is a popular model that can carry out anomaly detection with only consideration of normal training data, while it fails to capture spatio-temporal information from multivariate time series signals of multiple monitoring sensors. To mine the spatio-temporal information from multivariate time series signals, this paper proposes an attention graph stacked autoencoder for EMA anomaly detection. Firstly, attention graph convolution is introduced into autoencoder to convolve temporal information from neighbor features to current features based on different weight attentions. Secondly, stacked autoencoder is applied to mine spatial information from those new aggregated temporal features. Finally, based on the benchmark reconstruction loss of normal training data, different health thresholds calculated by several statistic indicators can carry out anomaly detection for new testing data. In comparison with traditional stacked autoencoder, the proposed model could obtain higher fault detection rate and lower false alarm rate in EMA anomaly detection experiment.

© 2024 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Electro-mechanical actuators (EMAs) are relatively compact and can offer high power-to-weight ratios and motion velocities, which are finding increasing use in landing gear and aileron of airplanes.<sup>1</sup> To remedy the lack of fault mechanism and knowledge about EMAs, Balaban et al.<sup>1</sup> explored the fault and failure information about EMAs based on failure modes and criticality analysis, literature review and industry experience.<sup>2</sup> Four fault modes, i.e., sensor, mechanical, motor and electrical, are summarized. In each fault mode, the fault,

\* Corresponding author.

E-mail address: [mqiang@scu.edu.cn](mailto:mqiang@scu.edu.cn) (Q. MIAO).

Peer review under responsibility of Editorial Committee of CJA.



Production and hosting by Elsevier

<https://doi.org/10.1016/j.cja.2024.03.024>

1000-9361 © 2024 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Please cite this article in press as: WANG J et al. An attention graph stacked autoencoder for anomaly detection of electro-mechanical actuator using spatio-temporal multivariate signals, *Chin J Aeronaut* (2024), <https://doi.org/10.1016/j.cja.2024.03.024>

failure, relative probability, relative criticality and model type are introduced for different components. In addition, multivariate time series signals from different sensors are usually applied to monitor the state of EMA. For example, the component like ballscrew in EMA is non-redundant, and it will trigger serious consequences when it is stuck during the flight. Thereby, to repair these damaged components, it is necessary to explore the effective state monitoring approaches.<sup>3</sup>

As a core technology in airplane health monitoring system, prognostic and health management (PHM)<sup>4</sup> can carry out anomaly detection, fault diagnosis, life prediction tasks, etc., for health monitoring of EMA. Fault diagnosis aims to identify fault type, position, or degree, while life prediction refers to performance degradation estimation and remaining useful life forecast for key component. In early studies, the model-based strategy was popular in EMA fault diagnosis and life prediction research.<sup>5–6</sup> However, it is difficult to develop a high-fidelity model for EMA with complex structures and variable working conditions.<sup>7</sup> With the development of machine learning, the data-driven strategy draws more attention in recent years. It can extract robust features or carry out end-to-end learning from monitoring signal, which has been validated to be effective in many efforts.<sup>8–15</sup> Abundant labeled failure data or degradation data are indispensable for developing data-driven models. However, it is challenging to collect labeled failure data and degradation data from actual airplane component<sup>16</sup> because the actual EMA operates under normal conditions in most cases and failures are less likely to happen or even not allowed to be simulated. Further, the collection of degradation data is extremely time-consuming because the degradation process of EMA is very slow, and even, this collection process will be interrupted when overhaul is implemented. Hence, it is difficult to carry out fault diagnosis and life prediction for EMA when failure data or degradation data are absent. Anomaly detection, as one preliminary approach in PHM, can detect anomaly behavior of monitoring signal. Further, anomaly detection model can be developed by only normal data and no longer need to collect failure or degradation data,<sup>17</sup> which is more valuable for its practical application in EMA health monitoring.

Some efforts about anomaly detection in key component are summarized as follows. Ruiz-Carcel and Starr<sup>18</sup> extracted five statistical characteristics from EMA position error and current signal, and further, two indicators, i.e., Hotelling  $T^2$  and squared prediction error, were applied to detect anomaly signal based on the indicators difference between anomaly data and normal data. Mazzoleni et al.<sup>19</sup> directly extracted root mean square and crest factor features from EMA current signal and then utilized relative density-ratio estimation to identify anomaly signal. Further, Pearson correlation technology was applied to calculate the correlation among multiple signals from normal working condition, and then, probability density function could build a distribution benchmark to distinguish anomaly signal.<sup>20</sup> In the above references, specific features usually need to be extracted from original signal through abundant expert knowledge or effective feature extraction algorithms. In addition, the robustness of those extracted features from original signal is also critical to the performance of anomaly detection task, which makes it difficult to generalize them in different working conditions. Neural network model can adaptively mine features from original data, and then, dif-

ferent evaluation criterions can be applied to detect those features from normal or anomaly state. The prediction-based neural network estimates the ideal signal to guide the anomaly detection task.<sup>21,22</sup> Once the residual value between the ideal signal and actual signal exceeds the specific threshold, it will trigger an anomaly alarm. However, it is time-consuming and labor-intensive to develop a robust model for multivariate time series signals.<sup>23</sup> Generative adversarial network takes advantage of discriminator to distinguish normal and anomaly sample.<sup>24</sup> The limitation of it is that additional simulative anomaly samples are usually required for discriminator-based neural network to guide the training process, while it is unrealistic to simulate all fault types in actual EMA.

The reconstruction loss-based neural network can obtain the reconstruction loss between input sample and reconstructed sample through unsupervised learning. Autoencoder is one of the most popular anomaly detection approaches and its reconstruction loss from normal data can be regarded as the benchmark to identify the anomaly signal based on different statistical indicators. Lin et al.<sup>25</sup> calculated the reconstruction loss of normal vibration signal and further constructed health indicator to detect the anomaly behavior in machine tool. Fan et al.<sup>26</sup> calculated the square prediction error of reconstruction loss to detect the anomaly behavior in monitoring signal of Tennessee Eastman process. Many sensors are usually applied to monitor the state of key component; however, the aforementioned efforts only select one kind of signal or one-dimensional reshaped signal to carry out anomaly detection task. It is difficult to capture the spatio-temporal information from multivariate time series signals. Further, some researchers utilized a  $m \times l$  matrix ( $m$  stands for the number of sensors and  $l$  is the number of features in each sensor) as the input of anomaly model to extract more effective features through dynamic principal component analysis<sup>27</sup> and dynamic stacked sparse autoencoder.<sup>28</sup> The curse of dimensionality problem is inevitable when  $m$  and  $l$  become large, and more complex network structure like convolutional autoencoder needs to be developed. In addition, they also fail to consider the weight difference of historical features in each signal. Hence, it is difficult for traditional autoencoder to mine the spatio-temporal information from multivariate time series signals.

In recent years, graph neural network (GNN) has aroused wide public concern for processing multivariate time series signals. The signal from each sensor can be regarded as a node and the connection between two different sensors is regarded as an edge in graph data, which can be applied to capture the spatio-temporal information of multivariate time series signals. Different features in each signal can also be regarded as the different nodes in graph architecture to learn the temporal correlation in each time series signal. Deng and Hooi<sup>29</sup> proposed a GNN anomaly detection framework to learn the relationships among different sensors, and then, the proposed model could predict future behavior and detect anomaly behavior of monitoring signal. Zhao et al.<sup>30</sup> utilized a graph attention network consisting of two graph attention layers to learn the spatial information among different sensors and temporal information in each time series signal. Further, a forecasting model was applied to detect anomaly value in monitoring signal. The monitoring signals are continuous and some anomaly behaviors like deviation anomaly, static

anomaly, point anomaly and so on, exist in continuous signal, so that it is feasible to use prediction-based GNN to detect the anomaly behavior in those signals. However, it is difficult to predict multiple time series signals simultaneously for EMA.

The high reliability of EMA in landing gear and aileron of airplane makes it challenging to acquire failure data while only normal data are available for developing anomaly detection model. In addition, more than seven monitoring signals are applied to monitor the health state of EMA, which bring the abundant operation information of EMA. It is labor-intensive and time-consuming to select the suitable features for different monitoring signals. The autoencoder based on reconstruction loss can extract adaptive features from original signal,<sup>25</sup> while it fails to capture the spatio-temporal information from multivariate time series signals. Inspired by the idea of GNN aforementioned, this paper proposes an attention graph stacked autoencoder to capture spatio-temporal information from multivariate time series signals for EMA anomaly detection. The attention graph convolution is introduced to aggregate temporal information from neighbor features to current feature, while stacked autoencoder can learn the spatial information among different sensors. Based on the weight parameter in two process, the proposed method can learn the spatio-temporal information from multivariate time series signals. Benchmark reconstruction loss of normal training data is further applied to detect the anomaly sample based on different statistical indicators. Main contributions of this paper are concluded as follows:

- (1) Based on the spatio-temporal information extraction process, the multi-dimensional time series signals are reduced to one-dimensional reconstruction loss.
- (2) Only normal operating data are applied to train the proposed model and their benchmark reconstruction loss can conduct anomaly detection task for new testing signals.
- (3) In comparison with stacked autoencoder without the consideration of attention graph convolution, abundant EMA anomaly detection case studies under various working conditions are conducted to indicate the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 introduces the details of the proposed method. Section 3 includes data description and preprocessing step of EMA. Section 4 gives several case studies about the proposed method. Section 5 draws a conclusion.

## 2. Proposed method

In Section 2.1, the overview of the proposed method is given firstly. In Section 2.2, attention graph convolution that can strengthen temporal information aggregation from neighbor features is illustrated. In Section 2.3, stacked autoencoder is applied to extract spatial information from new aggregation features of multiple sensors. In Section 2.4, several statistic thresholds of reconstruction loss from normal training data are regarded as benchmark value to detect anomaly samples.

### 2.1. Overview of proposed method

It is assumed that multivariate time series signals with  $m \times n$  dimensionality are applied to monitor the state of EMA, where  $m$  stands for the number of sensors and  $n$  stands for the number of features of each signal. Hence, multivariate time series signals from multiple sensors bring a spatial distribution characteristic, while the signals in each sensor have temporal relationship between historical features and future features. To mine the spatio-temporal information from multivariate time series signals of EMA, the proposed anomaly detection framework based on attention graph stacked autoencoder is shown in Fig. 1.

Step 1: We utilize attention graph convolution to carry out the temporal information aggregation for each signal. Assume that  $x_i^m$  stands for the  $i$ -th feature from the  $m$ -th sensor. The previous  $k$  features (i.e.,  $x_{i-1}^m, x_{i-2}^m, \dots, x_{i-k}^m$ ) and future  $k$  features (i.e.,  $x_{i+1}^m, x_{i+2}^m, \dots, x_{i+k}^m$ ) are regarded as neighbors for  $x_i^m$ . Hence, new  $x_i^m$  will be aggregated and updated by previous and future features through different weights.

Step 2: We utilize a stacked autoencoder to learn the spatial information among different sensors. The input of stacked autoencoder is a  $m \times 1$  matrix (i.e.,  $X_i = [x_i^1, x_i^2, \dots, x_i^m]^T$ ). To make the model stronger, pre-training is applied to initialize each encoder and decoder firstly, and then, fine-tuning strategy is further applied to optimize the whole network. In addition, it needs to be pointed out that attention graph convolution will also be introduced into stacked autoencoder to achieve temporal information aggregation of input features.

Step 3: Based on the benchmark reconstruction loss of normal training data, seven statistical indicators that can reflect the waveform change will be applied to calculate the different thresholds based on a sliding window. Once the new indicators of reconstruction loss of testing data captured by the sliding window with width  $l$  exceed those benchmark thresholds, it will be regarded as the anomaly sample.

### 2.2. Temporal information aggregation by attention graph convolution

Based on a graph data comprising of multiple nodes and connection edges, graph convolution can aggregate the information of neighbor nodes to current node, where each node includes features and an edge stands for the relationship between two nodes. It can propagate node information and learn new graph representation for those graph data.<sup>31,32</sup> For instance, a graph is defined as  $G = (X, A, E)$ .  $X$  represents a  $m \times n$  node feature matrix (i.e.,  $X \in \mathbb{R}^{m \times n}$ ), where  $m$  is the number of nodes and  $n$  is the number of features.  $A$  represents a  $m \times m$  adjacency matrix (i.e.,  $A \in \mathbb{R}^{m \times m}$ ) whose value  $A_{ij}$  denotes that there is an connection edge between the  $i$ -th and  $j$ -th node ( $i \neq j$ ).  $E$  is a set of edges ( $A_{ij} = (i, j) \in E$ ). Further, a degree matrix  $D \in \mathbb{R}^{m \times m}$  is introduced to represent the number of neighbor nodes of each node.  $D$  is a diagonal matrix, which can be deduced as follows:

$$D_i = \sum_j A_{ij} \quad (1)$$

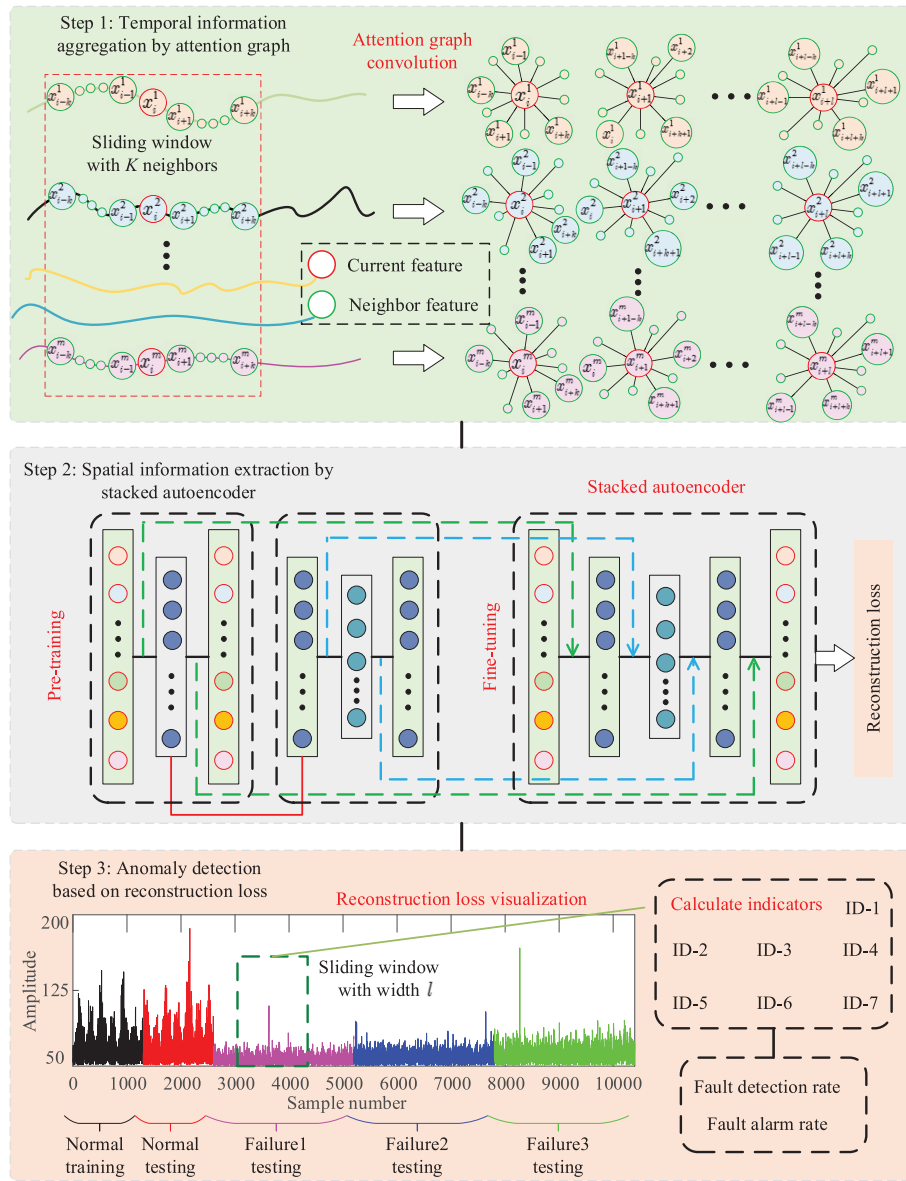


Fig. 1 Framework of proposed method.

Like the traditional neural network, GNN can map features from one layer to next layer. The propagation of GNN is divided into two steps, i.e., message passing and message aggregation. Let us assume that  $X^p$  denotes a node feature matrix in the  $p$ -th layer. The message passing value  $Y^p$  from their neighbor nodes can be deduced as follows:

$$Y^p = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X^p \quad (2)$$

$$\tilde{A} = A + I \quad (3)$$

$$\tilde{D} = \sum_j \tilde{A}_{ij} \quad (4)$$

When  $i=j$  and  $A_{ij} \neq 0$ , it denotes that there is a self-loop of current node feature. Hence,  $I$  is an identity matrix, and then, new adjacency matrix  $\tilde{A}$  and degree matrix  $\tilde{D}$  can be deduced

in Eq. (3) and Eq. (4), respectively.  $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$  is a normalized adjacency matrix. Further, the new node feature matrix  $X^{p+1}$  in  $(p+1)$ -th layer can be deduced by message aggregation, as shown in

$$X^{p+1} = \sigma(Y^p W^p) \quad (5)$$

where  $W^p$  stands for the weight matrix and  $\sigma(\cdot)$  is activation function. The rectified linear unit (ReLU) is regarded as the mapping function in this paper.

Inspired by the training process of GNN, we can also construct graph data for each monitoring signal. Hence, the previous  $k$  features (i.e.,  $x_{i-1}^m, x_{i-2}^m, \dots, x_{i-k}^m$ ) and future  $k$  features (i.e.,  $x_{i+1}^m, x_{i+2}^m, \dots, x_{i+k}^m$ ) are regarded as neighbors for current feature  $x_i^m$  in the  $m$ -th sensor, as shown in Fig. 1. In other words, we want to aggregate the information of those neighbors to current feature, which can help to learn the temporal correlation information in each monitoring signal. In each



graph data, the current feature  $x_i^m$  will be regarded as the central node and the other  $2k$  features will be regarded as neighbor nodes. There are only edge connections between center node and those neighbor nodes. Normalized adjacency matrix  $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$  in Eq. (2) assumes that the same weight is given for those neighbor nodes so that message passing will treat all neighbor features equally important. However, it fails to learn the dynamic temporal correlation information among those neighbor features. It is necessary to consider the importance of those neighbors and develop different contributions of them to current node. This process can be regarded as an attention process. Hence, a new weighted adjacency matrix  $\tilde{A}$  is introduced to assign different weights for different nodes,<sup>33</sup> which are shown as follows. Firstly, we utilize Euclidean distance ( $\text{dist}_{ij}^p$ ) to measure the distance between central node and each neighbor node, which can be deduced as follows:

$$\text{dist}_{ij}^p = \begin{cases} \|x_i^p - x_j^p\|_2^2 & \text{if } x_j^p \text{ is a neighbor of } x_i^p \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Secondly, softmax function is applied to normalize Euclidean distance  $\text{dist}_{ij}^p$  to assign different weights for adjacency matrix. The new weighted adjacency matrix  $\tilde{A}$  can be deduced as follows:

$$\tilde{A}_{ij}^p = \begin{cases} 1 & \text{if } i = j \\ \frac{\exp\left(-\frac{\text{dist}_{ij}^p}{\beta_i^p}\right)}{\sum_{j=1}^{2*k} \exp\left(-\frac{\text{dist}_{ij}^p}{\beta_i^p}\right)} & \text{if } i \neq j \text{ and } x_j^p \text{ is a neighbor of } x_i^p \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$\beta_i^p = \frac{1}{2 * k} \sum_{j=1}^{2*k} \text{dist}_{ij}^p \quad (8)$$

Finally, referring to attention graph propagation from the  $p$ -th layer to the  $(p+1)$ -th layer, temporal information from neighbor features can be aggregated to a new time series signal. The propagation of attention graph convolution can be deduced as follows:

$$X^{p+1} = \sigma\left(\tilde{A} X^p W^p\right) \quad (9)$$

### 2.3. Spatial information extraction by stacked autoencoder

Autoencoder is an unsupervised neural network that can mine the features from original samples through encoder and restore those features to original samples through decoder. Benefited from its same input and output, reconstruction loss minimization can be applied to guide the training process of autoencoder. In addition, reconstruction loss can also be applied in anomaly detection task, which is due to the fact that different reconstruction losses can be acquired in normal data and anomaly data. A typical autoencoder comprises an encoder and a decoder. Assume that input sample is defined as  $x_i = [x_i^1, x_i^2, \dots, x_i^m]^T$ , and the process of encoder can be deduced as follows:

$$z_i = f_1(W_1 x_i + b_1) \quad (10)$$

In Eq. (10),  $z_i$  stands for the extracted features (i.e.,  $z_i = [z_i^1, z_i^2, \dots, z_i^o]^T$ );  $m$  and  $o$  are the number of features in input sample and extracted features;  $W_1$  and  $b_1$  denote the weight and bias in encoder, respectively;  $f_1(\cdot)$  is a activation function. The reconstructed sample  $\hat{x}_i = [\hat{x}_i^1, \hat{x}_i^2, \dots, \hat{x}_i^m]^T$  can be acquired by decoder, as shown in

$$\hat{x}_i = f_2(W_2 z_i + b_2) \quad (11)$$

where  $W_2$  and  $b_2$  denote the weight and bias in decoder, respectively;  $f_2(\cdot)$  is a activation function. Further, the optimization of autoencoder can be guided by the following loss function:

$$\text{Loss} = \min \sum_{i=1}^m \|x_i - \hat{x}_i\|^2 \quad (12)$$

Here, we use mean square loss to measure the difference between  $x_i$  and  $\hat{x}_i$ . The weight  $W$  and bias  $b$  will be optimized by back propagation algorithm, as shown in

$$W = W - \eta \cdot \frac{\partial \text{Loss}}{\partial W} \quad (13)$$

$$b = b - \eta \cdot \frac{\partial \text{Loss}}{\partial b} \quad (14)$$

where  $\eta$  denotes the learning rate.

The stacked autoencoder is constructed by stacking multiple autoencoders, which can develop a deeper neural network and extract high-level representation. The encoder in each autoencoder will be stacked to extract the hidden features from last layer, and then, the symmetrical decoder will restore the those features to original data in turns. The vanishing gradient or falling in a local optimum are easy to happen when training this stacked autoencoder directly. Hence, two phases, i.e., pre-training and fine-tuning, are usually applied to make the stacked autoencoder stronger. In pre-training phase, the weight and bias in each autoencoder will be initialized. It needs to be pointed out that the extracted features from last autoencoder will be regarded as the input of next autoencoder. In fine-tuning phase, those initialized parameters in those autoencoders will be transferred to a stacked autoencoder, and then, this new network can be optimized again, as shown in Fig. 1.

In this paper, features from different sensors will be regarded as the input of the stacked autoencoder. Each input  $x_i$  is a  $m \times 1$  matrix, where  $m$  denotes the number of sensors. The stacked autoencoder can adaptively learn the spatial relationship of those features. To mine the spatio-temporal information from multivariate time series signals, attention graph convolution mentioned in Section 2.2 will be introduced into each layer of stacked autoencoder to aggregate the temporal information of neighbors to current feature, and then, the stacked autoencoder can learn the spatial information from those new aggregated features. Finally, the  $m \times n$  multivariate time series signals can be converted to a  $1 \times n$  reconstruction loss matrix.

### 2.4. Anomaly detection based on reconstruction loss

Since the combination of attention graph convolution and stacked autoencoder can extract spatio-temporal information from multivariate time series signals, we utilize the reconstruction loss in Eq. (12) to develop an anomaly detection frame-

work for EMA. Only normal data are regarded as the training data, and their reconstruction loss will be regarded as the benchmark loss. To quantify the difference in reconstruction loss, we employ time domain indicators to calculate the waveform difference in a sliding window with width  $l$ , as shown in Fig. 1. Seven popular statistical indicators as shown in Table 1, i.e., peak to peak value (ID-1), mean value (ID-2), root mean square (ID-3), kurtosis (ID-4), shape indicator (ID-5), impulse indicator (ID-6) and crest indicator (ID-7),<sup>34</sup> will be explored in case study. Since we can capture many samples from reconstruction loss through a sliding window with width  $l$ , it will form an indicator database for each statistical indicator. Two quantile of each indicator database, i.e., 1% and 99% quantile, are regarded as thresholds in this paper. In other words, the new sample whose statistical indicator is not between 1% and 99% quantile thresholds will be regarded as an anomaly sample. Otherwise, it will be classified as a normal sample, as shown in

$$\text{Behavior} = \begin{cases} \text{Anomaly} & \text{if } ID_j^{\text{test}} > ID_j^{\text{normal}(99\%)} \\ & \text{or } ID_j^{\text{test}} < ID_j^{\text{normal}(1\%)} \\ \text{Normal} & \text{otherwise} \end{cases} \quad (15)$$

where  $ID_j^{\text{test}}$  denotes the  $j$ -th indicator from test data;  $ID_j^{\text{normal}(99\%)}$  and  $ID_j^{\text{normal}(1\%)}$  represent 99% and 1% quantile threshold for the  $j$ -th indicator, respectively.

We utilize two anomaly detection criteria to evaluate the effectiveness of the proposed method. One is fault detection rate (FDR), which can calculate the percent of predicted

anomaly samples in actual anomaly samples. The other is false alarm rate (FAR), which can calculate the percent of samples incorrectly predicted as anomaly samples in actual normal samples. The FDR and FAR are defined in Eq. (16) and Eq. (17), respectively.

$$\text{FDR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (16)$$

$$\text{FAR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \times 100\% \quad (17)$$

where FDR and FAR can be deduced by Table 2. Further, we can also utilize the reconstruction loss visualization to find the anomaly data compared to the normal data, which will be shown in Section 4.4.

### 3. Experimental dataset and preprocessing step

The EMA dataset from NASA flyable EMA testbed is applied to test the performance of the proposed method. Section 3.1 introduces the EMA dataset. Section 3.2 illustrates the details of preprocessing step.

#### 3.1. Data description

EMA datasets from NASA flyable EMA testbed are multivariate time series signals.<sup>35</sup> As shown in Fig. 2, there are three actuators in EMA testbed. Actuator  $X$  is a testing actuator that is injected into different failures. Actuator  $Y$  is a health actuator without injecting any failure. Actuator  $Z$  can apply loads to the first two actuators. Four types of low-frequency signals with 100 Hz sampling frequency, i.e., motor voltage, motor current, motor temperature, and nut temperature, are applied to monitor the state of actuator  $X$  and  $Y$ . Three high-frequency signals from a three-axis ( $x$ ,  $y$  and  $z$  channel) acceleration sensor with 20 kHz sampling frequency are also applied to acquire the vibration signal. Eight working conditions are summarized in Table 3, with consideration of different driving command waveforms, amplitudes (frequency or duration), loads, and velocities. One normal state data and three failures (i.e., ballscrew jam, sensor fault and spall fault) exist in each working condition. Three failures will be regarded as the anomaly signal to test the performance of the proposed method.

#### 3.2. Preprocessing step

The proposed attention graph stacked autoencoder based on reconstruction loss for anomaly detection task is a combination of attention graph convolution and stacked autoencoder. Attention graph convolution is introduced into each layer of encoder and decoder in stacked autoencoder. The stacked

**Table 1** Statistic indicators.

Indicator	Description
ID - 1 = $\max(x) - \min(x)$	It measures the difference between maximum value and minimum value of each signal $x$ .
ID - 2 = $\frac{1}{N} \sum_{i=1}^N x_i$	It perceives the mean value of each signal $x$ .
ID - 3 = $\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$	It reflects the magnitude of amplitude and intensity of impact of each signal $x$ .
ID - 4 = $\frac{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^4}{(\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2)^2}$	It calculates the sharpness of the peak of the data distribution based on fourth-order cumulation. The higher value represents the sharper peak of each signal $x$ , where $\mu$ is the mean value of each signal.
ID - 5 = $\frac{\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}}{\frac{1}{N} \sum_{i=1}^N  x_i }$	It is the quotient of root mean square and mean absolute value, which can balance the influence of those two indicators.
ID - 6 = $\frac{\max  x }{\frac{1}{N} \sum_{i=1}^N  x_i }$	It is the quotient of maximum absolute value and mean absolute value. The higher value represents the bigger impact amplitude in signal.
ID - 7 = $\frac{\max  x }{\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}}$	It is also applied to measure the percent of impact amplitude in each signal based on maximum absolute value and root mean square.

**Table 2** Confusion matrix.

Confusion matrix		Actual value	
		Anomaly	Normal
Predicted value	Anomaly	TP	FP
	Normal	FN	TN

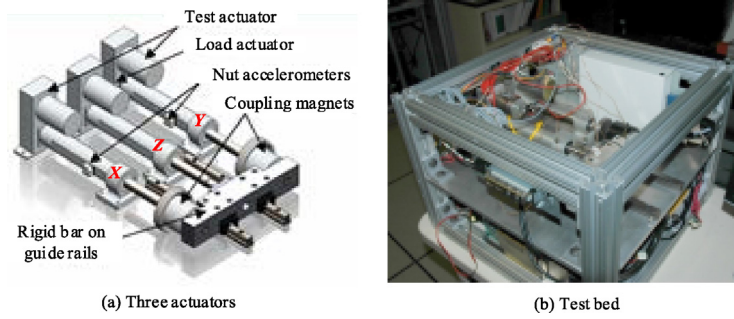


Fig. 2 EMA testbed.

Table 3 Experimental setup.

Command waveform	Amplitude (frequency or duration)	Load	Velocity	Working condition
Sinusoidal	40 mm, 0.5 Hz	0 lbs	0.04 m/s	A1
	40 mm, 1 Hz	0 lbs	0.08 m/s	A2
	40 mm, 1 Hz	10 lbs	0.08 m/s	A3
	80 mm, 0.25 Hz	-10 lbs	0.04 m/s	A4
Trapezoidal	40 mm, 22 s	-10 lbs	0.04 m/s	B1
	40 mm, 22 s	0 lbs	0.04 m/s	B2
	40 mm, 21 s	0 lbs	0.08 m/s	B3
	40 mm, 21 s	10 lbs	0.08 m/s	B4

autoencoder consists of five fully connected layers. The first two layers are regarded as two encoders. The last two layers are regarded as two decoders. The third layer is a hidden layer. The number of neurons from the first layer to the last layer are 7, 7, 3, 7 and 7, respectively. The activation function in the proposed model is a ReLU function. The learning rate of pre-training and fine-tuning of the stacked autoencoder are 0.005 and 0.0001, respectively. In addition, the training epochs of the pre-training and fine-tuning of the stacked autoencoder are 100 and 1000, respectively. The influence of neighbors  $K$  in attention graph convolution and sliding window width  $l$  in evaluating anomaly samples are further discussed in different case studies. To make the flowchart clearer, details about how to process EMA data and make them adapt well to the proposed method are illustrated as follows:

- (1) Down-sample signals: In each working condition, there are one normal data and three failure data. The data duration is 26 seconds. Since the sampling frequency of high-frequency signal is 20 kHz and the sampling frequency of low-frequency signal is 100 Hz, there are 520,000 features in each high-frequency signal and 2600 features in each low-frequency signal. We down-sample three high-frequency signals to make them have the same length as that in low-frequency signals. The mean value of a sliding window with width 200 is applied to replace original features in each high-frequency signal. In addition, motor temperature and nut temperature signal will be minus the environment temperature signal sampled in each experiment. Therefore, seven monitoring signals will be reconstructed to a  $7 \times 2600$  matrix in each data, where 7 stands for the seven sensors and 2600 stands for the number of features in each signal.

- (2) Train the proposed model: In each working condition, the normal state data are divided into normal training data and normal testing data. The normal training data correspond to the first 1300 features in each sensor, which are denoted as a  $7 \times 1300$  matrix, while the remaining normal data are regarded as normal testing data. Each failure testing data is a  $7 \times 2600$  matrix. The input of the stacked autoencoder is a  $7 \times 1$  matrix. Therefore, there are 1300 samples to train the stacked autoencoder. Finally, we can obtain a  $1 \times 1300$  reconstruction loss matrix as the benchmark loss to guide the anomaly detection task.
- (3) Detect anomaly behavior: Like the training process, we can also obtain a  $1 \times 1300$  and a  $1 \times 2600$  reconstruction loss matrix for normal testing data and each failure testing data, respectively. To carry out the anomaly detection for those reconstruction losses, a sliding window with width  $l$  is applied to capture reconstruction loss, and further construct a testing sample. Similar to that, this sliding window will also be applied in normal benchmark loss of normal training data. The interval of the sliding window is 10 in this paper. Hence, new training samples and testing samples will be extracted from reconstruction loss. Finally, we can follow Section 2.4 to carry out anomaly detection task.

#### 4. Case study

To illustrate the effectiveness of the proposed method, stacked autoencoder based on reconstruction loss is regarded as a comparison. The comparison method is named as Auto and the proposed method is named as Graph in the following case

studies. Section 4.1 gives an anomaly detection comparison about seven indicators under two different commands. Section 4.2 gives anomaly comparison results under the influence of the width of sliding window. Section 4.3 gives anomaly comparison results under the influence of the number of neighbor features. Section 4.4 gives an example of reconstruction loss visualization of two methods.

#### 4.1. Anomaly detection under two different commands

In this section, we conduct anomaly detection experiments under four working conditions and two different commands. The FDR of three failure testing data and the FAR of normal testing data are applied to evaluate the performance of Auto and Graph in each working condition. For example, three failure testing data and one normal testing data are defined as A1-1, A1-2, A1-3 and A1-4 in A1 working condition, respectively. In addition, the effectiveness of seven indicators are also compared in each experiment. To make the conclusion more convincing, we calculate three different average results in each table, as shown in Tables 4–7. Average1 is utilized to calculate the average results of seven indicators of three failure testing data and one normal testing data in each working condition, which can be applied to evaluate the overall performance of each method. Average2 is utilized to calculate the average FDR of 12 failure testing data in each indicator, which can be applied to compare the effectiveness of seven indicators. Average3 is utilized to calculate the average FAR of 4 normal testing data in each indicator, which can also be applied to compare the effectiveness of seven indicators. The number of neighbor features  $K$  is set as 1 and the width of sliding window  $l$  is set as 200 in this section.

##### 4.1.1. Sinusoidal command

The comparison results of Auto and Graph under sinusoidal command are shown in Table 4 and Table 5, respectively. Excellent model could obtain higher FDR in failure testing data and lower FAR in normal testing data. Firstly, from

the view of Average1, the FDR and FAR of two methods are drawn in Fig. 3(a) and (b), respectively. In Fig. 3(a), although the Graph performs worse than Auto in A1 working condition, it obtains the brilliant performance in A4 working condition. Graph obtains the better average FDR (69.34%) than that in Auto (58.10%). Hence, Graph can detect more failure samples than Auto. In Fig. 3(b), Graph obtains a lower average FAR (13.90%) than Auto (15.51%), which means that Graph can raise lower FAR in facing up with normal testing data. Secondly, from the view of Average2 and Average3, the FDR and FAR of two methods are drawn in Fig. 4(a) and (b), respectively. These two methods in sinusoidal command are defined as A-Auto and A-Graph in Fig. 4, which are corresponding to blue and red bar, respectively. In Fig. 4 (a), it can be found that ID-2 and ID-3 obtain higher accuracy than other five indicators (i.e., ID-1, ID-4, ID-5, ID-6, and ID-7). In addition, Graph can acquire better FDR than Auto in ID-2 and ID-3, which are corresponding to 93.19% and 99.41% in Table 5. In Fig. 4(b), Graph can acquire a lower FAR than Auto in ID-2 and ID-3. Although the lower FAR can be found in other several indicators, they will obtain the lower FDR in detecting the failure samples. In a word, it can be concluded that Graph performs better than Auto in most tasks under sinusoidal command, especially when ID-2 and ID-3 are applied to detect anomaly samples.

##### 4.1.2. Trapezoidal command

The comparison results of Auto and Graph under trapezoidal command are shown in Table 6 and Table 7, respectively. The anomaly detection tasks in trapezoidal command are more challenging than those in sinusoidal command, because most of FDR in Table 6 and Table 7 are lower than that in Table 4 and Table 5, and also, higher average FAR can be found in Table 6 and Table 7 than that in Table 4 and Table 5. To make this conclusion more intuitive, Average 1 of Table 6 and Table 7 are drawn in Fig. 3, and also, Average2 and Average3 of Table 6 and Table 7 are drawn in Fig. 4. These two methods in trapezoidal command are defined as B-Auto and B-Graph

**Table 4** FDR and FAR of seven indicators of Auto under sinusoidal command.

Auto		ID-1	ID-2	ID-3	ID-4	ID-5	ID-6	ID-7	Average1
FDR	A1-1	2.49%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	87.37%
	A1-2	8.30%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
	A1-3	24.07%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
FAR	A1-4	18.02%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.57%
FDR	A2-1	40.25%	100.00%	100.00%	7.47%	98.76%	13.69%	12.86%	45.92%
	A2-2	10.79%	100.00%	100.00%	2.49%	71.37%	4.98%	4.98%	
	A2-3	22.41%	100.00%	100.00%	27.80%	24.90%	10.79%	10.79%	
FAR	A2-4	0.00%	100.00%	100.00%	2.70%	43.24%	0.00%	0.00%	35.14%
FDR	A3-1	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	90.24%
	A3-2	67.22%	93.78%	83.82%	71.78%	89.21%	97.93%	97.10%	
	A3-3	16.60%	100.00%	97.93%	81.33%	100.00%	100.00%	98.34%	
FAR	A3-4	27.03%	23.42%	36.94%	9.01%	5.41%	9.01%	18.02%	18.40%
FDR	A4-1	0.00%	0.00%	0.00%	0.00%	46.06%	49.38%	10.79%	8.85%
	A4-2	9.96%	11.62%	11.20%	0.00%	0.00%	0.41%	0.00%	
	A4-3	24.90%	0.00%	0.00%	0.00%	20.75%	0.83%	0.00%	
FAR	A4-4	5.41%	9.91%	10.81%	0.00%	7.21%	8.11%	0.00%	5.92%
Average2		27.25%	75.45%	74.41%	49.24%	70.92%	56.50%	52.90%	58.10%
Average3		12.61%	33.33%	36.94%	2.93%	13.96%	4.28%	4.50%	15.51%



**Table 5** FDR and FAR of seven indicators of Graph under sinusoidal command.

Graph		ID-1	ID-2	ID-3	ID-4	ID-5	ID-6	ID-7	Average1
FDR	A1-1	91.70%	100.00%	100.00%	21.58%	16.18%	17.01%	17.84%	58.64%
	A1-2	93.36%	100.00%	100.00%	31.95%	64.32%	45.64%	11.62%	
	A1-3	100.00%	100.00%	100.00%	23.24%	58.09%	29.05%	9.96%	
FAR	A1-4	19.82%	29.73%	21.62%	19.82%	13.51%	19.82%	19.82%	20.59%
FDR	A2-1	100.00%	100.00%	100.00%	2.49%	0.00%	0.83%	2.07%	50.17%
	A2-2	100.00%	100.00%	100.00%	0.00%	70.95%	0.00%	0.00%	
	A2-3	91.70%	100.00%	100.00%	0.00%	76.35%	4.98%	4.15%	
FAR	A2-4	25.23%	5.41%	4.50%	0.90%	0.00%	18.92%	7.21%	8.88%
FDR	A3-1	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	87.06%
	A3-2	8.30%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
	A3-3	19.92%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
FAR	A3-4	18.02%	0.00%	0.00%	0.00%	0.00%	1.80%	18.02%	5.41%
FDR	A4-1	100.00%	100.00%	100.00%	48.96%	100.00%	100.00%	61.83%	81.49%
	A4-2	100.00%	100.00%	100.00%	57.68%	100.00%	95.85%	58.92%	
	A4-3	100.00%	18.26%	92.95%	52.28%	99.59%	71.37%	53.53%	
FAR	A4-4	82.88%	20.72%	39.64%	0.90%	0.00%	0.00%	0.90%	20.72%
Average2		75.41%	93.19%	99.41%	44.85%	73.79%	55.39%	43.33%	69.34%
Average3		36.49%	13.96%	16.44%	5.41%	3.38%	10.14%	11.49%	13.90%

**Table 6** FDR and FAR of seven indicators of Auto under trapezoidal command.

Auto		ID-1	ID-2	ID-3	ID-4	ID-5	ID-6	ID-7	Average1
FDR	B1-1	9.54%	51.04%	49.79%	44.40%	93.36%	48.96%	36.51%	36.55%
	B1-2	14.52%	48.13%	37.76%	33.20%	91.29%	24.48%	18.26%	
	B1-3	2.49%	38.17%	36.10%	35.27%	24.48%	13.69%	16.18%	
FAR	B1-4	40.54%	63.06%	63.06%	61.26%	55.86%	17.12%	34.23%	47.88%
FDR	B2-1	0.00%	100.00%	100.00%	53.11%	93.36%	68.88%	68.05%	53.90%
	B2-2	16.18%	100.00%	100.00%	18.67%	61.00%	30.71%	30.29%	
	B2-3	8.30%	100.00%	100.00%	17.84%	9.96%	28.22%	27.39%	
FAR	B2-4	0.90%	100.00%	100.00%	13.51%	0.00%	0.90%	0.90%	30.89%
FDR	B3-1	15.35%	47.72%	40.66%	26.97%	82.16%	48.55%	34.85%	30.49%
	B3-2	43.57%	31.54%	27.39%	4.15%	38.17%	29.88%	12.86%	
	B3-3	39.42%	22.82%	23.24%	9.13%	28.22%	21.16%	12.45%	
FAR	B3-4	45.95%	60.36%	62.16%	55.86%	63.06%	13.51%	27.03%	46.85%
FDR	B4-1	7.88%	61.83%	36.51%	9.96%	20.75%	11.20%	10.37%	18.22%
	B4-2	0.00%	51.45%	29.46%	1.66%	16.60%	0.41%	0.41%	
	B4-3	13.28%	40.25%	32.37%	4.56%	13.28%	11.20%	9.13%	
FAR	B4-4	4.50%	2.70%	1.80%	14.41%	20.72%	4.50%	4.50%	7.59%
Average2		14.21%	57.75%	51.11%	21.58%	47.72%	28.11%	23.06%	34.79%
Average3		22.97%	56.53%	56.76%	36.26%	34.91%	9.01%	16.67%	33.30%

in Fig. 4, which are corresponding to green and purple bar, respectively. Further, we compare the performance of two methods. Firstly, from the view of Average1 in Fig. 3 (a) and (b), Graph can still acquire higher average FDR (42.35%) and lower average FAR (21.69%) than that in Auto (corresponding to 34.79% and 33.30%), respectively. Secondly, from the view of Average2 and Average3, the FDR and FAR of B-Auto and B-Graph are drawn in Fig. 4 (a) and (b), respectively. It can be found that ID-2 and ID-3 still obtain higher FDRs than other five indicators in Fig. 4 (a) for both two methods. In addition, Graph also obtains higher FDRs than Auto based on ID-2 and ID-3, which are corresponding to 87.48% and 85.30%, respectively. Although ID-2 and ID-3 obtain higher FARs than the other five indicators,

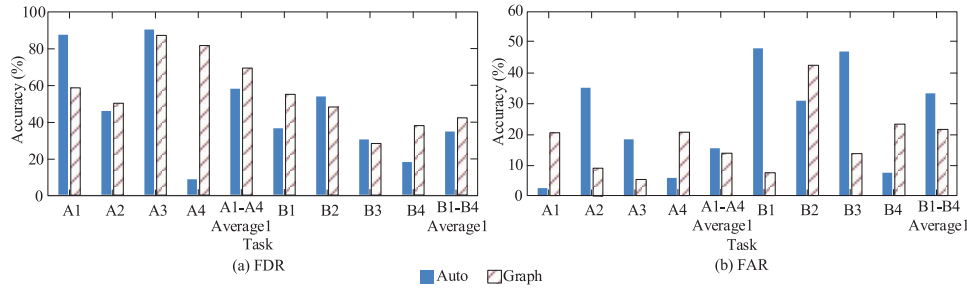
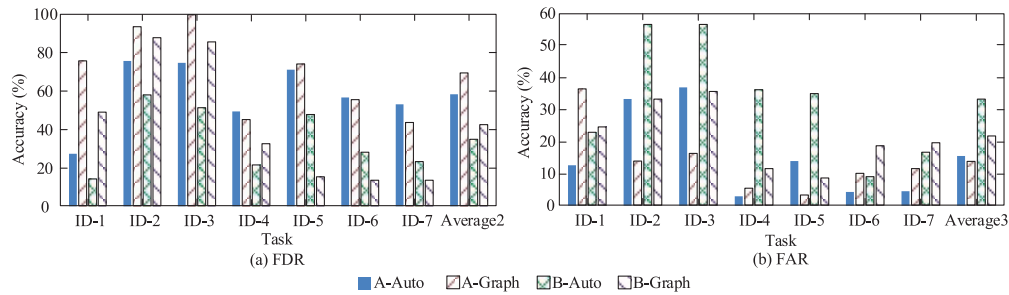
Graph can relieve this phenomenon compared to Auto. In a word, it can be concluded that the proposed method (Graph) performs better than Auto in most tasks under trapezoidal command. In addition, it is preferred to select ID-2 and ID-3 as the anomaly detection indicators in failure data, while we can combine other indicators to evaluate normal testing data since they acquire lower FAR in Fig. 4 (b), especially for ID-4, ID-5, ID-6 and ID-7.

#### 4.2. Influence of width of sliding window

When the width of sliding window  $l$  changes, different values of seven indicators can be acquired based on the captured features. Hence, we conduct an experiment to explore the influ-

**Table 7** FDR and FAR of seven indicators of Graph under trapezoidal command.

Graph		ID-1	ID-2	ID-3	ID-4	ID-5	ID-6	ID-7	Average1
FDR	B1-1	100.00%	100.00%	100.00%	92.12%	0.00%	0.00%	0.00%	55.09%
	B1-2	100.00%	100.00%	100.00%	100.00%	0.00%	0.00%	0.00%	
	B1-3	71.37%	100.00%	93.36%	100.00%	0.00%	0.00%	0.00%	
FAR	B1-4	18.92%	0.00%	14.41%	3.60%	5.41%	4.50%	5.41%	7.46%
FDR	B2-1	82.99%	100.00%	100.00%	13.28%	19.92%	14.94%	14.94%	47.99%
	B2-2	33.20%	100.00%	100.00%	26.14%	14.94%	34.44%	34.02%	
	B2-3	25.73%	100.00%	100.00%	24.07%	16.18%	26.97%	26.14%	
FAR	B2-4	20.72%	93.69%	93.69%	32.43%	4.50%	25.23%	25.23%	42.21%
FDR	B3-1	21.58%	100.00%	100.00%	5.39%	21.58%	4.56%	6.22%	28.31%
	B3-2	8.30%	99.17%	97.93%	6.22%	11.20%	7.05%	7.05%	
	B3-3	0.00%	48.96%	36.51%	2.49%	9.13%	0.41%	0.83%	
FAR	B3-4	21.62%	26.13%	21.62%	0.00%	5.41%	10.81%	10.81%	13.77%
FDR	B4-1	88.38%	100.00%	100.00%	9.96%	36.93%	35.68%	31.54%	38.02%
	B4-2	47.30%	85.06%	79.25%	6.22%	27.80%	19.92%	21.16%	
	B4-3	8.30%	16.60%	16.60%	2.49%	26.56%	18.67%	19.92%	
FAR	B4-4	36.94%	13.51%	12.61%	9.91%	18.92%	34.23%	36.94%	23.29%
Average2		48.93%	87.48%	85.30%	32.37%	15.35%	13.55%	13.49%	42.35%
Average3		24.55%	33.33%	35.59%	11.49%	8.56%	18.69%	19.59%	21.69%

**Fig. 3** Accuracy histogram of Average1 from Table 4 to Table 7.**Fig. 4** Accuracy histogram of Average2 and Average3 from Table 4 to Table 7.

ence of width of sliding window in this section. The width of sliding window is set as 200, 400, 600, 800 and 1000, respectively. The interval of the sliding window is still 10. The number of neighbor features  $K$  is still 1. It needs to be pointed out that we directly calculate Average1 in each experiment and adopt them in this case study. In addition, we further use Average4 to calculate the average value of Average1 for two methods.

The FDR results of Auto and Graph are shown in Table 8. As the window width increases, the FDR of the most tasks of

two methods also gradually increase. Although Graph achieves lower FDR than Auto in A1 task when  $l$  is 200, Graph can obtain more outstanding performance when  $l$  is set as a higher value, which are higher than 96% since  $l$  is set from 400 to 1000. In addition, the improvement of FDR from 200 to 400 are remarkable in two methods. Especially for A1, A2, A3, A4 and B1 tasks, Graph can acquire more than 90% FDR when  $l$  is equal to 400 compared to the lower FDR when  $l$  is 200. Graph obtains better Average4 than Auto whether  $l$  is equal to a small value or a high value in five dif-

ferent widths. Therefore, Graph can detect more anomaly samples and it is also preferred to choose a big window width to achieve accurate anomaly detection.

The FAR results of Auto and Graph are shown in Table 9. Obviously, bigger window width brings higher FAR for two methods in most tasks. However, we want to reduce the FAR when facing up with normal data. It is improper to choose a too big value like 800 or 1000 since Average4 exceeds 65% in two methods, which will bring a frequent false alarm for normal operating data. The small values like 200, 400 and 600 can reduce the false alarm compared to the high FAR based on window width 800 and 1000. It needs to be pointed out that Graph can acquire lower Average4 than Auto when we choose 200, 400 and 600, respectively. Hence, it indicates that the proposed method (Graph) performs better than Auto since it can reduce the false alarm when facing up with normal operating signal. In a word, it is preferred to choose 400 or 600 as the sliding window width for Graph, due to the fact that Graph can not only acquire higher FDR but also obtain lower FAR than Auto in those two situations.

#### 4.3. Influence of number of neighbor features

Since attention graph convolution is applied to aggregate the temporal information of neighbor features in the proposed method, various previous and future temporal information may bring a different effect on current feature by selecting different neighbors  $K$ . Hence, we conduct an experiment to explore the influence of number of neighbor features in this section. The number of neighbor features  $K$  are as 1, 5, 9, 13 and 17, which are defined as Graph-1, Graph-5, Graph-9, Graph-13 and Graph-17 for the proposed method, respectively. In addition, the width of sliding window with 200, 400, 600, 800 and 1000 are also considered in this section. It needs to be pointed out that we directly calculate Average4 mentioned in Section 4.3 in each experiment and use them in this case study. We use Average5 to calculate the average value of Average4 for two methods.

The FDR results of Auto and Graph are shown in Table 10. When the width of sliding window is fixed, different  $K$  will influence the proposed method. Graph will keep a high FDR in most tasks when  $K$  is equal to 1, 5 and 9, except for the specific case Graph-9 under window width 200. When  $K$  is set to a big value like 13 and 17, the performance of Graph will

drop a lot. In other words, attention graph convolution proposed in this paper can adapt well to small  $K$  value and capture the short temporal information, while is improper to aggregate too many neighbor features. In addition, it can obtain the same conclusion like that in Section 4.2, i.e., bigger window width will bring a higher FDR for two methods. In comparison with Auto, Graph can acquire better FDR when  $K$  is equal to 1, 5 and 9, respectively. Hence, it is preferred to choose a small  $K$  value like 1 and 5 to obtain better performance in detecting anomaly samples.

The FAR results of Auto and Graph are shown in Table 11. Taking the sliding window with width 200, 400 and 600 as examples, Graph-1, Graph-5 and Graph-9 can produce less false alarm than Auto since they acquire lower FAR. However, too big  $K$  like 13 and 17 will increase the FAR in Graph-13 and Graph-17, respectively. In addition, it is also not a good choice to choose a big sliding window width like 800 and 1000 for the Graph, since Graph acquires higher FAR than Auto in most experiment even when we utilize different  $K$ . From the view of Average5 in Table 11, Graph can obtain lower average FAR than Auto when  $K$  is equal to 1, 5 and 9 and window width is equal to 200, 400 and 600. In a word, it is also preferred to choose a small  $K$  value like 1 and 5 to decrease the FAR for the proposed method.

#### 4.4. Anomaly detection based on visual reconstruction loss

Since reconstruction loss is applied to guide the anomaly detection task in this paper, we give a visual reconstruction loss example about A1 task in this section. Firstly, we plot the original seven monitoring signals from normal training data, normal testing data and three failure testing data, as shown in Fig. 5(a) to (g), respectively. Secondly, we plot the reconstruction loss of Auto, as shown in Fig. 5(h). Thirdly, we plot the reconstruction loss of Graph with five  $K$  values (i.e., 1, 5, 9, 13 and 17) discussed in Section 4.3, which are corresponding to Fig. 5(i) to (m), respectively. In each subplot picture, normal training data and their reconstruction loss are marked by black color. Normal testing data and their reconstruction loss are marked by red color. Three failure testing data and their reconstruction loss are marked by pink, blue and green color, respectively. It is easy to distinguish the failure testing data in some original signals like Fig. 5(e) to (g). However, if we directly utilize all original signals to carry out the anomaly

**Table 8** FDR results of Auto and Graph under the influence of width sliding window.

Auto	A1	A2	A3	A4	B1	B2	B3	B4	Average4
200	87.37%	45.92%	90.24%	8.85%	36.55%	53.90%	30.49%	18.22%	46.44%
400	89.85%	56.54%	96.60%	89.49%	47.06%	60.12%	44.52%	35.04%	64.90%
600	89.98%	81.33%	97.65%	75.41%	70.05%	62.31%	53.85%	73.99%	75.57%
800	92.11%	84.66%	99.76%	94.95%	74.95%	63.90%	54.62%	75.53%	80.06%
1000	97.19%	90.62%	100.00%	90.71%	95.65%	82.17%	73.32%	96.66%	90.79%
Graph	A1	A2	A3	A4	B1	B2	B3	B4	Average4
200	58.64%	50.17%	87.06%	81.49%	55.09%	47.99%	28.31%	38.02%	55.85%
400	96.14%	100.00%	92.98%	93.43%	100.00%	60.16%	31.95%	46.86%	77.69%
600	98.96%	100.00%	91.66%	90.81%	100.00%	71.10%	42.24%	64.27%	82.38%
800	98.79%	100.00%	92.19%	94.45%	100.00%	92.42%	79.01%	85.87%	92.84%
1000	99.20%	100.00%	100.00%	94.47%	100.00%	95.92%	76.63%	93.55%	94.97%

**Table 9** FAR results of Auto and Graph under the influence of width sliding window.

Auto	A1	A2	A3	A4	B1	B2	B3	B4	Average4
200	2.57%	35.14%	18.40%	5.92%	47.88%	30.89%	46.85%	7.59%	24.40%
400	20.88%	46.62%	55.26%	1.10%	58.40%	41.44%	59.97%	37.21%	40.11%
600	24.14%	64.39%	74.85%	20.93%	67.61%	37.42%	62.78%	80.28%	54.05%
800	86.83%	75.35%	98.88%	17.65%	64.71%	36.97%	54.62%	98.04%	66.63%
1000	60.37%	66.36%	98.62%	48.39%	88.02%	85.71%	62.21%	100.00%	76.21%
Graph	A1	A2	A3	A4	B1	B2	B3	B4	Average4
200	20.59%	8.88%	5.41%	20.72%	7.46%	42.21%	13.77%	23.29%	17.79%
400	46.62%	27.00%	25.75%	58.71%	19.62%	57.77%	24.33%	40.35%	37.52%
600	75.25%	27.16%	32.19%	56.94%	20.32%	66.80%	44.87%	60.36%	47.99%
800	84.03%	70.03%	79.83%	89.08%	57.98%	78.99%	85.15%	83.75%	78.61%
1000	88.48%	75.58%	64.98%	86.18%	50.23%	91.24%	72.81%	90.32%	77.48%

**Table 10** FDR results of Auto and Graph under the influence of number of neighbor features.

FDR	200	400	600	800	1000	Average5
Auto	46.44%	64.90%	75.57%	80.06%	90.79%	71.55%
Graph-1	55.85%	77.69%	82.38%	92.84%	94.97%	80.75%
Graph-5	52.49%	76.33%	84.67%	91.33%	95.06%	79.98%
Graph-9	22.00%	72.05%	80.90%	88.87%	94.40%	71.64%
Graph-13	36.20%	46.34%	50.87%	56.86%	87.50%	55.55%
Graph-17	22.74%	35.96%	45.64%	48.21%	83.30%	47.17%

**Table 11** FAR results of Auto and Graph under the influence of number of neighbor features.

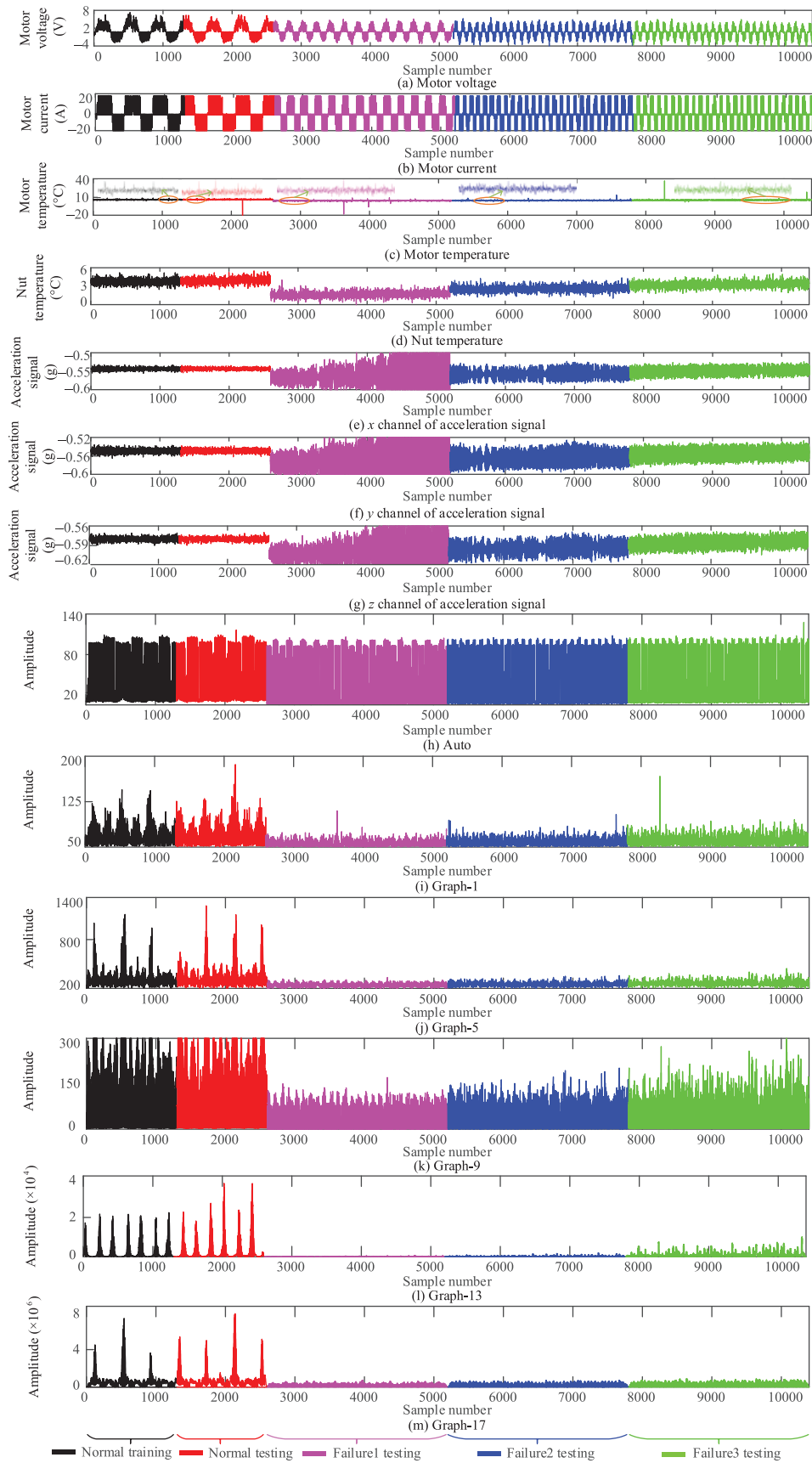
FAR	200	400	600	800	1000	Average5
Auto	24.40%	40.11%	54.05%	66.63%	76.21%	52.28%
Graph-1	17.79%	37.52%	47.99%	78.61%	77.48%	51.88%
Graph-5	14.61%	32.10%	52.31%	75.98%	83.70%	51.74%
Graph-9	22.28%	35.85%	52.46%	71.50%	76.50%	51.72%
Graph-13	25.74%	46.74%	60.41%	70.59%	88.94%	58.48%
Graph-17	32.27%	50.73%	57.57%	59.24%	89.46%	57.85%

detection task, it requires to obtain the enough expert knowledge and the analyses of seven sensors are time-consuming. Further, it is labor intensive to calculate different indicators for each monitoring signal. The anomaly detection based on one-dimensional reconstruction loss can reduce workload effectively. In comparison with Auto (Fig. 5(h)), the reconstruction loss plots of Graph (Fig. 5(i) to (m)) are more easily distinguishable for normal data and failure data. The shape of reconstruction loss of normal testing data is similar to that in normal training data, while the shapes of reconstruction loss of three failure testing data are different from that in normal training data. From the view of the influence of number of neighbor features, more significant difference can be found in Fig. 5(l) and (m). In addition, Graph can also obtain the significant difference between normal data and failure data in Fig. 5(i) and (j) that are corresponding to Graph-1 and Graph-5 respectively.

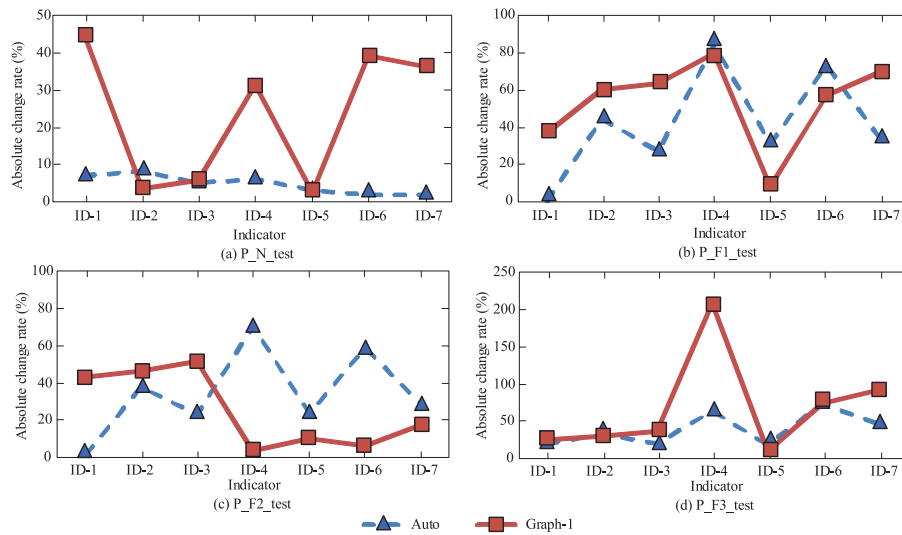
Furthermore, we also calculate seven indicators for Auto (Fig. 5(h)) and Graph-1 (Fig. 5(i)), respectively. It needs to

be pointed out that the width of sliding window is set to 1300 and 2600 in normal data and failure data, respectively. The experiment of seven values from seven indicators under normal training data is defined as  $N_{train}$ . Like this, one normal testing data and three failure testing data experiment are defined as  $N_{test}$ ,  $F1_{test}$ ,  $F2_{test}$ , and  $F3_{test}$ , respectively. In addition, in comparison with normal indicators from normal training data, absolute change rate of each indicator of four testing data are further calculated, which are corresponding to  $P_{N_{test}}$ ,  $P_{F1_{test}}$ ,  $P_{F2_{test}}$ , and  $P_{F3_{test}}$ , respectively. The comparison results of Auto and Graph-1 are shown in Table 12, which are corresponding to the blue and orange color in Fig. 6. A good indicator should achieve low absolute change rate in normal testing data while have high absolute change rate in failure testing data. Hence, in comparison with Auto, Graph can acquire low value ( $P_{N_{test}}$ ) in Fig. 6(a) and obtain high value ( $P_{F1_{test}}$ ,  $P_{F2_{test}}$ , and  $P_{F3_{test}}$ ) in Fig. 6(b)–(d) based on ID-2 and ID-3. Being similar to the conclusion in Section 4.1, ID-2 and ID-3 are two more effective





**Fig. 5** Reconstruction loss visualization of A1 of Auto and Graph based on different  $K$ .



**Fig. 6** Absolute change rate of one normal testing data and three failure testing data.

**Table 12** Different indicators and absolute change rate of Auto and Graph-1.

Auto	N_train	N_test	P_N_test	F1_test	P_F1_test	F2_test	P_F2_test	F3_test	P_F3_test
ID-1	106.91	114.28	6.89%	106.86	0.05%	108.36	1.36%	128.3	20.01%
ID-2	44.29	47.9	8.15%	24.82	43.96%	27.57	37.75%	29.98	32.31%
ID-3	62.01	65.01	4.84%	45.5	26.62%	47.75	23.00%	49.54	20.11%
ID-4	-1.73	-1.83	5.78%	-0.28	83.82%	-0.52	69.94%	-0.62	64.16%
ID-5	1.4	1.36	2.86%	1.83	30.71%	1.73	23.57%	1.65	17.86%
ID-6	2.51	2.47	1.59%	4.34	72.91%	3.98	58.57%	4.36	73.71%
ID-7	1.79	1.82	1.68%	2.37	32.40%	2.3	28.49%	2.64	47.49%
Graph-1	N_train	N_test	P_N_test	F1_test	P_F1_test	F2_test	P_F2_test	F3_test	P_F3_test
ID-1	148.44	214.1	44.23%	94.5	36.34%	84.95	42.77%	184.19	24.08%
ID-2	27.26	28.16	3.30%	10.96	59.79%	14.72	46.00%	19.33	29.09%
ID-3	35.96	37.96	5.56%	13.36	62.85%	17.57	51.14%	23.21	35.46%
ID-4	3.62	4.75	31.22%	6.48	79.01%	3.75	3.59%	11.14	207.73%
ID-5	1.32	1.35	2.27%	1.22	7.58%	1.19	9.85%	1.2	9.09%
ID-6	5.52	7.67	38.95%	8.65	56.70%	5.83	5.62%	9.59	73.73%
ID-7	4.18	5.69	36.12%	7.1	69.86%	4.88	16.75%	7.99	91.15%

indicators in most tasks. Hence, we recommend to combine the visual reconstruction loss, ID-2 and ID-3 indicators to carry out the anomaly detection task.

## 5. Conclusions

In this paper, we propose an attention graph stacked autoencoder to mine the spatio-temporal information from EMA multivariate time series signals. Only normal training data are applied to train the proposed model and failure samples can be detected based on the benchmark indicators of reconstruction loss from normal training data. Several conclusions can be drawn as follows: (1) Benefited from the spatio-temporal information extraction, the proposed method can acquire higher FDR and lower FAR than traditional stacked autoencoder in most tasks. (2) It is preferred to select ID-2 and ID-3 as the health indicators to guide the anomaly detection task in the proposed method. (3) It is preferred to choose suitable values for the width of sliding window and the number

of neighbor features in this experiment. (4) It is preferred to introduce reconstruction loss visualization in the proposed method to help to detect the anomaly signal. In the future, we will continue to focus on this topic and try to validate the effectiveness of anomaly detection methods in actual equipment.

## CRedit authorship contribution statement

**Jianyu WANG:** Formal analysis, Investigation, Methodology, Writing – original draft. **Heng ZHANG:** Supervision, Validation, Visualization. **Qiang MIAO:** Funding acquisition, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (No. 52075349), the National Natural Science Foundation of China (No. 62303335), the Postdoctoral Researcher Program of China (No. GZC20231779), and the Natural Science Foundation of Sichuan Province (No. 2022NSFSC1942).

## References

- Balaban E, Bansal P, Stoelting P, et al. A diagnostic approach for electro-mechanical actuators in aerospace systems. In: *2009 IEEE Aerospace Conference*. 2009. p. 1–13.
- Bodden DS, Clements NS, Schley B, et al. Seeded failure testing and analysis of an electro-mechanical actuator. In: *2007 IEEE Aerospace Conference*. 2007. p. 1–8.
- Chen BJ, Shen CQ, Shi JJ, et al. Continual learning fault diagnosis: a dual-branch adaptive aggregation residual network for fault diagnosis with machine increments. *Chin J Aeronaut* 2023;**36**(6):361–77.
- Zhao ZB, Wu JY, Li TF, et al. Challenges and opportunities of AI-enabled monitoring, diagnosis & prognosis: a review. *Chin J Mech Eng* 2021;**34**(1):34–56.
- Hussain Y, Burrow S, Henson L, et al. A review of techniques to mitigate jamming in electromechanical actuators for safety critical applications. *Int J Progn Health M* 2020;**9**(3):1–12.
- Zhang YJ, Peng Y, Liu DT. Degradation estimation of electro-mechanical actuator with multiple failure modes using integrated health indicators. *IEEE Sensors J* 2022;**22**(11):10783–93.
- Fu J, Maré JC, Fu YL. Modelling and simulation of flight control electromechanical actuators with special focus on model architecting, multidisciplinary effects and power flows. *Chin J Aeronaut* 2017;**30**(1):47–65.
- Liu KJ, Feng YW, Xue XF. Fault diagnosis of hydraulic retraction system based on multi-source signals feature fusion and health assessment for the actuator. *J Intell Fuzzy Syst* 2018;**34**(6):3635–49.
- Wang JY, Zeng ZG, Zhang H, et al. An improved triplet network for electromechanical actuator fault diagnosis based on similarity strategy. *IEEE Trans. Instrum. Meas* 2022;**71**:1–10.
- Wang JY, Miao JG, Wang JL, et al. Fault diagnosis of electrohydraulic actuator based on multiple source signals: An experimental investigation. *Neurocomputing* 2020;**417**:224–38.
- Siahpour S, Li X, Lee J. Deep learning-based cross-sensor domain adaptation for fault diagnosis of electro-mechanical actuators. *Int J Dynam Control* 2020;**8**(4):1054–62.
- Wang JY, Zhang YJ, Luo C, et al. Deep learning domain adaptation for electro-mechanical actuator fault diagnosis under variable driving waveforms. *IEEE Sensors J* 2022;**22**(11):10783–93.
- Zhang YJ, Liu LS, Peng Y, et al. Health indicator extraction with phase current for power electronics of electro-mechanical actuator. *Measurement* 2020;**159**:107787–800.
- Jiang XX, Huang Q, Shen CQ, et al. Synchronous chirp mode extraction: a promising tool for fault diagnosis of rolling element bearings under varying speed conditions. *Chin J Aeronaut* 2022;**35**(1):348–64.
- Qian WW, Li SM, Jiang XX. Deep transfer network for rotating machine fault analysis. *Pattern Recogn* 2019;**96**:106993–7006.
- Wang JY, Zhang H, Miao Q. Source free unsupervised domain adaptation for electro-mechanical actuator fault diagnosis. *Chin J Aeronaut* 2023;**36**(4):252–67.
- Dong L, Liu SL, Zhang HL. A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples. *Pattern Recogn* 2017;**64**:374–85.
- Ruiz-Carcel C, Starr A. Data-based detection and diagnosis of faults in linear actuators. *IEEE Trans. Instrum. Meas* 2018;**67**(9):2035–47.
- Mazzoleni M, Scandella M, Maccarana Y, et al. Condition assessment of electro-mechanical actuators for aerospace using relative density-ratio estimation. *IFAC* 2018;**51**(15):957–62.
- Krogerus T, Hyvönen M, Multanen P, et al. Joint probability distributions of correlation coefficients in the diagnostics of mobile work machines. *Mechatronics* 2016;**35**:82–90.
- Zhong J, Zhang YJ, Wang JY, et al. Unmanned aerial vehicle flight data anomaly detection and recovery prediction based on spatio-temporal correlation. *IEEE Trans. Rel* 2021;**71**(1):1–12.
- Li Z, Li J, Wang Y, et al. A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment. *Int J Adv Manuf Technol* 2019;**103**:499–510.
- Zhu HQ, Wang QL, Zhang FX, et al. A prediction method of electrocoagulation reactor removal rate based on long term and short term memory–Autoregressive integrated moving average model. *Process Saf Environ Prot* 2021;**152**:462–70.
- Plakias S, Boutalis YS. Exploiting the generative adversarial framework for one-class multi-dimensional fault detection. *Neurocomputing* 2019;**332**:396–405.
- Lin CY, Weng CP, Wang LC. Edge-based RNN anomaly detection platform in machine tools. *Smart Sci* 2019;**7**:139–46.
- Fan J, Wang W, Zhang H. AutoEncoder based high-dimensional data fault detection system. In: *2017 IEEE 15th International Conference on Industrial Informatics*. 2017. p. 1001–6.
- Shang J, Chen MY, Zhang HW. Fault detection based on augmented kernel Mahalanobis distance for nonlinear dynamic processes. *Comput Chem Eng* 2018;**109**:311–21.
- Jiang L, Ge Z, Song Z. Semi-supervised fault classification based on dynamic sparse stacked auto-encoders model. *Chemom Intel Lab Syst* 2017;**168**:72–83.
- Deng AL, Hooi B. Graph neural network-based anomaly detection in multivariate time series. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021. p. 4027–35.
- Zhao H, Wang YJ, Duan JY, et al. Multivariate time-series anomaly detection via graph attention network. In: *2020 IEEE International Conference on Data Mining (ICDM)*. 2020. p. 841–50.
- Li TF, Zhou Z, Li SN, et al. The emerging graph neural networks for intelligent fault diagnostics and prognostics: a guideline and a benchmark study. *Mech Syst Sig Process* 2022;**168**:108653–89.
- Xiao YM, Shao HD, Feng MJ, et al. Towards trustworthy rotating machinery fault diagnosis via attention uncertainty in transformer. *J Manuf Syst* 2023;**70**:186–201.
- Liu L, Zhao HT, Hu ZW. Graph dynamic autoencoder for fault detection. *Chem Eng Sci* 2022;**254**:117637–52.
- Nayana BR, Geethanjali P. Analysis of statistical time-domain features effectiveness in identification of bearing faults from vibration signal. *IEEE Sensors J* 2017;**17**(17):5618–25.
- Balaban E, Saxena A, Narasimhan S, et al. Experimental validation of a prognostic health management system for electro-mechanical actuators. In: *Infotech@Aerospace*. 2011. p. 1–14.