



Long short-term memory neural network based fault detection and isolation for electro-mechanical actuators

Jing Yang*, Yingqing Guo, Wanli Zhao

School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, PR China

ARTICLE INFO

Article history:

Received 13 June 2018

Revised 23 November 2018

Accepted 9 June 2019

Available online 13 June 2019

Communicated by Shen Yin

Keywords:

Long short-term memory

Neural network

Fault diagnosis and isolation

Electro-mechanical actuators

ABSTRACT

In the new generation of aircraft, electro-mechanical actuators (EMA) have been replacing the conventional hydraulic versions. Despite the fact that a failure of this system can seriously affect the safety of vehicles and their operators, there are few studies that focus on fault diagnosis for the units. In this paper, we present an innovative fault detection and isolation method for the EMA. Our method is tested and verified against three types of failures. This novel fault diagnosis method works by creating a model for sensor data by using the known time series. It utilizes an advanced Long Short-term Memory (LSTM) neural network, which can effectively handle time series data in this domain. A modification to the LSTM network is applied in order to take advantage of the correlation between sensors. In addition, the algorithm uses a sliding window to improve performance of LSTM applied to fault isolation. Our research has revealed that the proposed algorithm is better able to detect faults when compared to traditional neural networks. We also compare our performance with the support vector machine algorithm and the typical LSTM algorithm. Ultimately, the proposed method performs superiorly for the task of fault isolation.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

With the advent of the next generation of aerospace systems equipped with fly-by-wire controls, Electro-mechanical Actuators (EMA) are quickly becoming components critical to the safety of aerospace vehicles. Actuation is used in all vehicles (aircraft, spacecraft, ground vehicles, etc.) to control the position and/or altitude of the vehicle, and to deploy or retract equipment, particularly for embedded optic instruments such as cameras and telescopes. As such, the actuation is a safety critical system, particularly when humans may be catastrophically affected by failures within the system. Therefore, actuator fault diagnosis is of great significance.

EMA fault diagnosis algorithms can be divided into model-based and data-driven algorithms. The model-based approach has the advantage of tracking individual components within the system, thereby identifying faults from a physical perspective in a straightforward manner. This allows for a very detailed and accurate fault identification, giving the operator the ability to see a particular component's state of health in real time. A careful analysis of EMA failures and the use of neural network methods for fault diagnosis is described in [1]. A fault detection and isolation

generator, designed by applying advanced null space computation techniques, is proposed in [2]. This approach monitors three sensors relevant for the control of the actuator. A set of five signal-based monitoring functions are designed using a detailed model of the system under consideration in [3]. The aforementioned model-based methods have done well with respect to diagnostic performance; however, there is significant trouble in capturing all of the relationships within a system with a demand for such high accuracy. Although feasible, the computational requirements are very high.

A data driven approach, on the other hand, has very little information about the system. These methods allow for sensor data to be used directly, and consequently, the data-based approach is less computationally expensive. This approach has been widely adopted for the task of fault diagnosis [4–7]. A modified version of the Principal Direction Divisive Partitioning (PDDP), based on a Chi-squared statistical test algorithm, is proposed in [8]. Algorithms based on wavelet analysis feature extraction and Artificial Neural Network (ANN) fault classification are proposed in [9,10]. In another study, variational mode decomposition (VMD), multifractal detrended fluctuation analysis (MFdfa) and a probabilistic neural network (PNN) were utilized for EMA fault diagnosis [11]. A fault diagnosis method for EMA based on wavelet packets and a self-organizing map (SOM) neural network was proposed in [12]. At present, the application of neural networks in EMA fault diagnosis lies in the training of classifiers, post-feature extraction.

* Corresponding author.

E-mail address: Yangjing86@mail.nwpu.edu.cn (J. Yang).

In summary, EMA fault diagnosis has the following problems:

- 1) Model-based methods achieve a high accuracy, but they are computationally expensive and an accurate model is difficult to obtain;
- 2) The data-driven method does not require an accurate model. However, the vibration signal of the EMA is nonlinear, non-stationary, and strong time-varying. Extant data-based methods do not generally perform well when building models to fit these characteristics.

In order to solve the above problems, we propose a data-driven EMA fault diagnosis algorithm based on the Long Short-term Memory (LSTM) neural network. Traditional ANN algorithms are prone to overfitting, and they routinely plateau at a local minimum. Additionally, selecting the structure of the network, including the number of hidden neurons, is difficult. These problems create issues of practicality when trying to apply the traditional neural network as a solution. In order to rectify this problem, Recurrent Neural Networks (RNN) [13–16] have been developed. This approach has shown improvement over the traditional ANN, and is more suitable for building time series models. It has also been used for fault diagnosis tasks. Traditional RNN is used in the construction of nonlinear models for actuator fault diagnosis [17,18]. Also, the Elman network has been improved in terms of a faster convergence and a generalization ability. This is attained through designing a self-feedback connection with fixed gain in the unit connection, and increasing the feedback of the output layer node [19]. However, if RNN allows for the storage of information, over time the network gradients tend to vanish. The LSTM network is a type of RNN, and it is also one of the best application [20–22]. It performs well in the field of speech recognition [23,24], text classification [25–28] and more [29]. In [30–32], the field of fault prediction is considered to be the main subject in nonlinear systems. To this point, there has been little research on the application of LSTM in fault diagnosis. Based on its excellent performance in time series problems [33], our research aims to apply LSTM for fault detection and isolation of the actuator. The goal is to build a time series model that also has the ability to classify faults, thereby solving problems in EMA fault diagnosis.

Methods for time series modeling are generally focused on explaining the structure of independent single sensor data, such as the fault diagnosis of rotating machinery. Missing from this analysis is the correlation that exists between the various sensors and the actuator. By considering data from only the single sensor during the process, it potentially overlooks information that can lead to valuable insights. One way to overcome this problem is to consider other sensors as part of the input to the model. Our results show that this leads to significant benefits in terms of time series modeling and fault detection.

The LSTM neural network is a commonly used technique for classification problems, and thus it can be applied to solve the fault isolation problem. To determine the type of failure, residuals of actual data and the estimated values of the time series model are employed. Training the classification model directly on the residual sequences is problematic for two reasons. First, the speed of the regression analysis is slow; moreover, the classification accuracy is poor. To improve the classification accuracy and training speed, a sliding window method is used to enhance the algorithm and improve the accuracy of fault isolation.

The proposed algorithm was validated using the test data in the Airborne Electro-Mechanical Actuator [34]. Compared with the Nonlinear Aggressive Neural Network (NARNN), RNN, and traditional LSTM neural networks, the proposed detection algorithm provides several advantages. For example, it has superior data modeling accuracy and fault detection. In addition, it has

significant benefits in terms of fault isolation as compared to the Support Vector Machine (SVM) and traditional LSTM algorithms.

Based on the above discussion, the main contributions of this work can therefore be summarized as follows:

- 1) The temporal characteristics of EMA observation data are considered, and a fault diagnosis framework based on LSTM for time series analysis is proposed. The framework effectively avoids the problem suffered by model-based fault diagnosis methods, where it is difficult to obtain an accurate model. Also, it obtains better fault diagnosis results than the traditional neural network;
- 2) This method applies the LSTM network to time series modeling and classification for fault diagnosis. Two modifications to the algorithm based on the actual EMA sensor data characteristics are proposed. The novel algorithms achieve better performance than the basic LSTM network in both applications.

The remainder of this paper is organized as follows. Section 2 presents the concept of the LSTM-RNN algorithm, as well as a method for LSTM-based time series modeling and classification. Section 3 presents our modified LSTM networks for EMA fault diagnosis and isolation usage. A complete fault diagnosis process is given. The FDI simulation results are presented in Section 4. Section 5 concludes the paper.

2. LSTM neural network based time series modeling and classification methods

2.1. Concept of LSTM

A recurrent neural network is an improved class of artificial neural network. It uses the temporal information of the input data, where connections between units form a directed cycle within the same layer. In contrast, a conventional neural network only has connections between layers. The units within in a single layer have no connection. The network does not transmit the temporal information, so the performance for time series data may be poor. However, if RNN allows the storage of information over time, the network gradients tend to vanish. If this happens then the RNN will be no longer be in a position to learn. The LSTM neural network, which is a deep-learning net, is explicitly designed to learn the long-term dependencies. It is capable of retaining information for long periods of time via the introduced gates. The *forget gate* can discard redundant information; the *input gate* is able to select key information to be stored in the internal state; and the *output gate* is used to identify output information.

The LSTM was proposed by Hochreiter and Schmidhuber [35] and was improved and promoted by Gers [20]. Compared to the standard RNN neural network layer, while the horizontal line runs through the top of the graph, the LSTM has three gates to protect and control the cell state. The first step in the LSTM is to decide what information will be discarded from the cell state. This decision was done through a layer called forgetting the door. The gate reads h_{t-1} and x_t and outputs a value between 0 and 1 for each number in the cell state C_{t-1} . 1 means ‘completely reserved’, 0 means ‘completely discarded’. The next step is to ascertain what new information is stored in the cell state. There are two parts here. First, the sigmoid layer called ‘input layer’ determines what value we will update. Then a *tanh* layer creates a new candidate vector and C_t is added to the state. In the next step, these two messages produce updates to the status. Multiply the current state with f_t and discard the information that is needed. Then add i_t and C_t . This is the latest candidate that changes based on the degree to which we decide to update each state. In the end, determining what value to output. This output will be built on the current

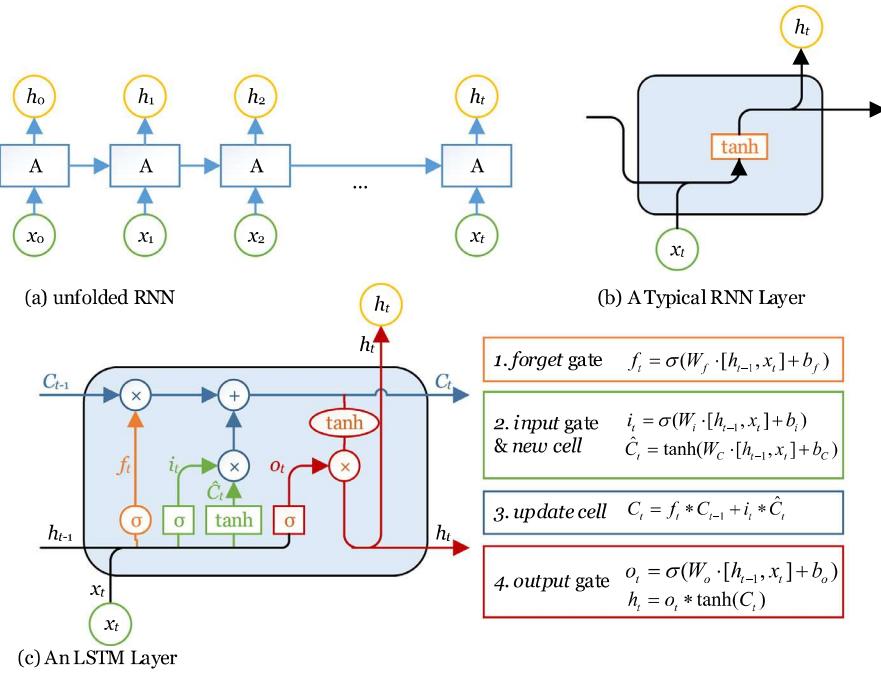


Fig. 1. LSTM layer schematic and comparison with typical RNN layer.

cell state, but it is also a filtered version. First, running a *sigmoid* layer to identify which part of the cell's state will be output. Next, the state of the cell is processed through *tanh* (a value between -1 and 1) and multiplied by the output of the *sigmoid* gate, and only the part that determines the output is finally output. The brief schematic of LSTM layer is presented in Fig. 1.

The proposed LSTM network uses the SGDM (Stochastic Gradient Descent with Momentum) optimizer. The traditional Stochastic Gradient Descent (SGD) algorithm updates the network parameters (weights and biases) to minimize the loss function by taking small steps in the direction of the negative gradient of the loss,

$$\theta_{l+1} = \theta_l - \alpha \nabla E(\theta_l) \quad (1)$$

where l represents the iteration number, $\alpha > 0$ is the learning rate, θ is the parameter vector, and $E(\theta)$ is the loss function. The gradient of the loss function, $\nabla E(\theta)$, is evaluated using the entire training set. The standard gradient descent algorithm uses all of the data in one pass.

The stochastic gradient descent algorithm may oscillate along the path of steepest descent as it moves towards the optimum. Adding a momentum term to the parameter update is one way to reduce this oscillation. The SGDM update is

$$\theta_{l+1} = \theta_l - \alpha \nabla E(\theta_l) + \gamma (\theta_l - \theta_{l-1}) \quad (2)$$

where γ determines the contribution of the previous gradient step to the current iteration.

Adding a regularization term to the weights for the loss function, $E(\theta)$, is one way to reduce overfitting. The regularization term is also called the weight decay. The loss function with the regularization term takes the form

$$E_R(\theta) = E(\theta) + \lambda \Omega(w) \quad (3)$$

where w is the weight vector, λ is the regularization factor, and the regularization function $\Omega(w)$ is

$$\Omega(w) = \frac{1}{2} w^T w \quad (4)$$

2.2. LSTM network based time series modeling

Time series modeling uses existing data to build a model that predicts future values. The premise of modeling is that the time series is a stable, predictable sequence that is generally representative of future values. The classic time series modeling method is known as the Autoregressive (AR) model algorithm [36–38], the equation is as follows:

$$x_t = c + \sum_{i=1}^p \varphi_i x_{t-i} + \varepsilon_t \quad (5)$$

where $\varphi_1, \dots, \varphi_p$ are the parameters of the mode, c is a constant, and ε_t is white noise.

Based on the AR model, the time series algorithm can be extended to adaptive learning algorithms such as neural networks. The predictive accuracy of this enhanced algorithm is significantly higher than the traditional version. The equation is as follows:

$$x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-n}) \quad t \in (n+1, k) \quad (6)$$

According to the equation, a network f can be obtained through the training of the $k-n$ sets of data. In this paper, the LSTM algorithm is used to establish the time series model. Unlike the traditional neural network, the training phase of the LSTM neural network is not independent at each step of the process. The network is not trained directly from the current moment t , but rather, it continues from the previous training results from 0 to $t-1$, making full use of the temporal continuity of the time series. When the LSTM network trains the model, x_{t-1} is used as the input and x_t is the output of the regression data. The training flow is expressed as under:

$$x_1 = \dot{\ell}(x_0), \dots, x_2 = \ddot{\ell}(x_1), \dots, x_t = \overbrace{\ell}^{\dots}(x_{t-1}), \dots, x_k = \ell(x_{k-1}) \quad (7)$$

where, $\ell = \hbar(\hat{\ell})$, $\ell = \hbar(\dots(\hbar(\hat{\ell})))$. $\hat{\ell}, \ell, \dots, \ell$ represents the trained LSTM and its intermediate network, and \hbar represents the modified rule for each step.

The final stage of the process is training, after which the LSTM neural network model is built. The neural network containing the LSTM layer is seen in [Table 1](#). It contains an input layer, an LSTM layer, a fully connected layer, and a final regression layer.

Table 1
Time series modeling LSTM network.

i.	InputLayer
ii.	LstmLayer
iii.	FullyConnectedLayer
iv.	RegressionLayer

Table 2
Classification LSTM network.

v.	InputLayer
vi.	LstmLayer
vii.	FullyConnectedLayer
viii.	SoftmaxLayer
ix.	ClassificationLayer

2.3. LSTM network based classification method

The LSTM neural network can also be used as a method to train a classification network using known input and fault types. This method has been used in many industries for a variety of tasks [24,39–42].

It is understood that within a set of data X , X contains many subsets $\{X_1, X_2, \dots, X_n\}$, where each X_i corresponds to a type, or class, p_i , $p_i \in \{p_1, p_2, \dots, p_n\}$. Neural networks can be used as a predictive model for classification, where they seek to discover and represent a connection between X_i and p_i . The formula is as follows:

$$p_i = \mathbb{R}(X_i) \quad (8)$$

where, \mathbb{R} is the representation method. X_i is expanded, $X_i = [x_i^1 \ x_i^2 \ \dots \ x_i^l]$, x_i^j is a column vector, l is the length of the data (also the length of time). When using the LSTM network for training this type of model, the length of the data needs to be considered. In order to utilize the LSTM temporal characteristics, p_i must be extended to the same length as X_i , $P_i = [p_i \ p_i \ \dots \ p_i]$, where the length of the row vector is l . The final combined training data is

$$X = \{X_1, X_2, \dots, X_n\} = \{x_1^1, x_1^2, \dots, x_1^l, x_2^1, x_2^2, \dots, x_2^l, \dots, x_n^1, x_n^2, \dots, x_n^l\} \quad (9)$$

Target type is

$$P = \{P_1, P_2, \dots, P_n\} = \{p_1, p_1, \dots, p_1, p_2, p_2, \dots, p_2, p_n, p_n, \dots, p_n\} \quad (10)$$

The equation for training the classifier using the LSTM algorithm is as follows:

$$p_i = \hat{\lambda}(x_1^1), p_i = \hat{\lambda}(x_1^2), \dots, p_i = \hat{\lambda}(x_n^l) \quad (11)$$

$$\ddot{\lambda} = \tilde{\lambda}(\dot{\lambda}), \dot{\lambda} = \tilde{\lambda}(\dots(\tilde{\lambda}(\dot{\lambda}))) \quad (12)$$

where, $\dot{\lambda}, \ddot{\lambda}, \dots, \tilde{\lambda}$ is the trained LSTM and its intermediate process network and $\tilde{\lambda}$ is the modified rule for each step of $\dot{\lambda}$.

The description of the LSTM neural network for classification is described in 0. It includes an input layer, an LSTM layer, a fully connected layer, a softmax layer, and a classification layer (Table 2).

3. Modified LSTM network based fault detection and isolation

3.1. Cross-LSTM based fault detection

The time series based fault detection approach focuses on the behavior of individual sensor values over blocks of time. Signal

structural changes are the indicators of an off-nominal engine condition. The basic approach uses a set of training data to develop the model for predicting the present value of a sensor, using previous sensor values. A set of data is represented as $(\alpha, \beta, \gamma, \dots) \in X$, $\alpha, \beta, \gamma, \dots$ is the time series signals, and $t = [0, 1, 2, \dots]$. The fault diagnosis algorithm is using the past $\alpha(t)$ ($t = 0, 1, \dots, k-1$) to estimate the current $\alpha(k)$, and comparing the predicted value $\hat{\alpha}(t)$ with the measured value $\alpha(t)$. If there is a deviation, thinking that there is an abnormal situation.

$$\hat{\alpha}(t) = \Lambda(\alpha(t-1), \alpha(t-2), \dots, \alpha(0)) \quad \text{compare}(\hat{\alpha}(t)),$$

$$\alpha(t) \begin{cases} \text{if}(match) & \text{normal} \\ \text{if}(mismatch) & \text{abnormal} \end{cases} \quad (13)$$

$$\hat{\beta}(t) = \Gamma(\beta(t-1), \beta(t-2), \dots, \beta(0)) \quad \text{compare}(\hat{\beta}(t)),$$

$$\beta(t) \begin{cases} \text{if}(match) & \text{normal} \\ \text{if}(mismatch) & \text{abnormal} \end{cases} \quad (14)$$

Among Eqs. (13) and (14), Λ, Γ present the time series models. Training method of the time series model is performed according to the method given in Section 2.2.

During the detection progress, it is usually performed for a single time series when training the network. Look for features through the context of the time series. However, the object of our study (actuator) consists of a set of sensor parameters. In addition to having their own time relationship, they are related to each other, that is, the changes between them have synchronous rules. This means that if only the time continuity of the signal is considered, other characteristics may be lost, resulting in unsatisfactory performance of fault detection and separation. In order to prove the relevance of the data, the Pearson correlation value is calculated between the sensor parameters. Pearson correlation value $\rho_{x,y}$ is calculated as shown below.

$$\rho_{x,y} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}} \quad (15)$$

The strength of the correlation can be determined according to the Pearson correlation value. 0.8–1.0 means extremely strong correlation, 0.6–0.8 means strong correlation, 0.4–0.6 means moderate correlation and 0.2–0.4 means weak correlation. Using the Pearson correlation value to calculate the parameters of the study object, it is found that there are correlations between different sensors, and the correlation between specific sensors is very strong. The Fig. 2 shows the calculation of Pearson correlation between measured load sensor and the other four sensors. There are correlations between them, especially between the measured load sensor and motor Y current or motor Z current.

Based on the correlation analysis results, a modification is made to the fault detection method using LSTM network. The traditional time series analysis is performed on the signal of a single sensor. Considering the correlation between sensors, all sensor signals are used as input parameters of the LSTM network for time series analysis. In order to improve the utilization of data features, to get better fault detection results. The proposed detection method is named as cross-LSTM. The training equation for the sensor α is as follows:

$$\begin{aligned} \dot{\alpha}_1 &= \ell(\alpha_0, \beta_0, \gamma_0, \dots), \dot{\alpha}_2 = \ell(\alpha_1, \beta_1, \gamma_1, \dots), \dots, \dot{\alpha}_t \\ &= \ell(\alpha_{t-1}, \beta_{t-1}, \gamma_{t-2}, \dots) \end{aligned} \quad (16)$$

Taking the Cross-LSTM of two sensors as an example, the training diagram is shown in Fig. 3. The sensors are $[\alpha, \beta]$. When α sensor sequence is modeled, both α and β are used as input, and the same is for β sensor.

After completing the LSTM neural network model, the prediction data at time t is calculated through the model and data from

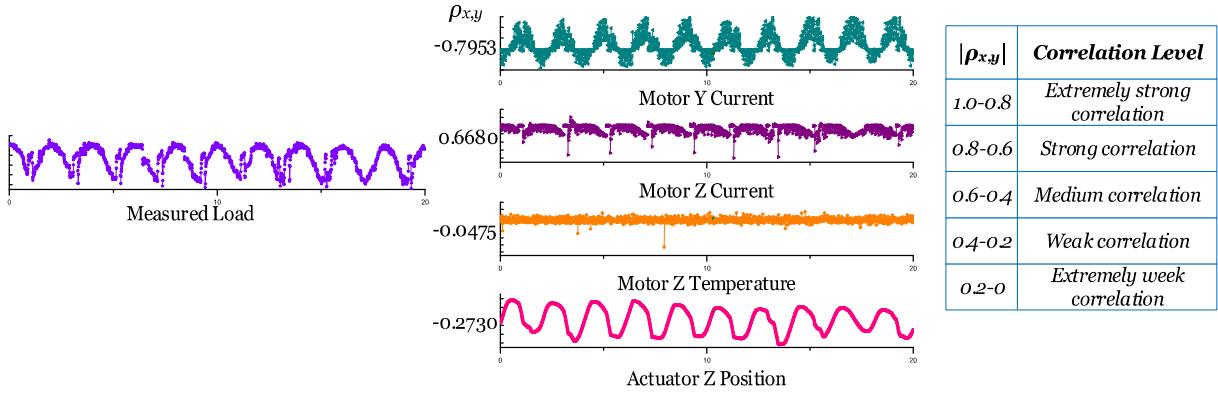


Fig. 2. Correlation of sensors.

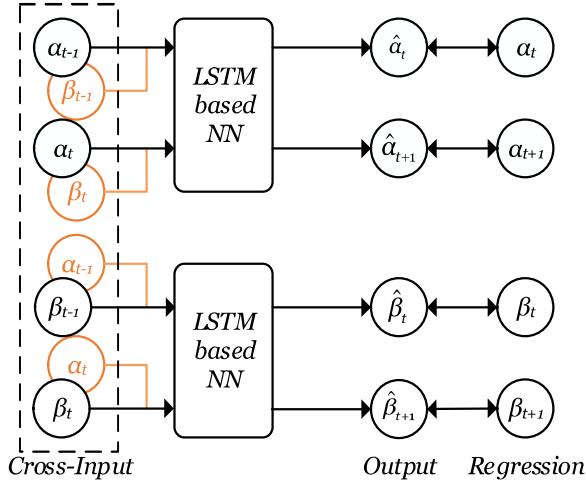


Fig. 3. Cross-LSTM schematic of two sensors.

time 1 to $t-1$. The residuals R of the predicted data and the measured data are obtained. For the sensor group ($\alpha, \beta, \gamma, \dots$), the formulas are as follows:

$$(\hat{\alpha}_2, \dots, \hat{\alpha}_t) = \ell_\alpha(\alpha_1, \dots, \alpha_{t-1}), r_\alpha(k) = \hat{\alpha}_k - \alpha_k, k = 2, 3, \dots, t \quad (17)$$

$$(\hat{\beta}_2, \dots, \hat{\beta}_t) = \ell_\beta(\beta_1, \dots, \beta_{t-1}), r_\beta(k) = \hat{\beta}_k - \beta_k, k = 2, 3, \dots, t \quad (18)$$

$$(\hat{\gamma}_2, \dots, \hat{\gamma}_t) = \ell_\gamma(\gamma_1, \dots, \gamma_{t-1}), r_\gamma(k) = \hat{\gamma}_k - \gamma_k, k = 2, 3, \dots, t \quad (19)$$

The constructed residual signals are then utilized as indicators of the health status, given that the residuals change before and after the occurrence of a fault. Consequently, by selecting a proper threshold band the engine faults can be detected by monitoring the variations in the residuals. To generate the threshold bands, the mean (μ) and the standard deviation (σ) of the residuals are obtained when the engine is operating under the healthy condition. The threshold bands are then specified according to $t.h.upper = \mu + z\sigma$ and $t.h.lower = \mu - z\sigma$, corresponding to the upper and lower bands, respectively. By assuming a normal distribution associated with residuals, a 99% confidence interval can be determined by selecting z . A fault in the actuator would then be detected if any of the five residual signals passes its corresponding thresholds that are defined by the band $[t.h.lower, t.h.upper]$.

3.2. Sliding-window LSTM based fault isolation

Isolation uses the residuals obtained from a set of fault data, which has known fault types. Then, the classification network is trained with the fault type. In this work, the LSTM neural network algorithm is used for fault isolation, as described in Section 2.3. Given the types of fault, the residual array R and the known fault type $T(k) \in \{fault(1), fault(2), \dots, fault(n)\}$ is applied to train isolation's LSTM neural network. R is the residual matrix used for training, which contains the residual corresponding to all the fault type data to be trained, and the size is $t \times n$ (Eq. (20)). The residual vector at time k is $R(k)$, which contains the residual data of n sensors at time k , and the size is $1 \times n$ (Eq. (21)). The training formula of isolation is shown in Eq. (22), where the trained LSTM neural network corresponds to each $R(k)$ and $T(k)$.

$$R = \begin{bmatrix} r_\alpha(2) & r_\alpha(3) & \dots & r_\alpha(t) \\ r_\beta(2) & r_\beta(3) & \dots & r_\beta(t) \\ r_\gamma(2) & r_\gamma(3) & \dots & r_\gamma(t) \\ \dots & \dots & \dots & \dots \end{bmatrix} \quad (20)$$

$$R(k) = (r_\alpha(k), r_\beta(k), r_\gamma(k), \dots), k = 2, 3, \dots, t \quad (21)$$

$$\{T(2), T(3), \dots, T(k)\} \approx \lambda \{R(2), R(3), \dots, R(t)\} \quad (22)$$

The training method described above is between single sensor parameters and types, which has relatively poor robustness. Therefore, some improvements were made in this method. In the traditional isolation network, the residual data of n sensor is usually used for fault classification. It means $1 \times n$ data is used as *InputLayer* to extract fault data characteristics. However, in the proposed algorithm, the value of a certain point used in training data is expanded to a section containing several points in order to improve the accuracy of classification. This several points obtained by a sliding window. The sliding window implementation will be smoother and will automatically eliminate glitches. Therefore, we use sliding window-LSTM (SW-LSTM) to improve this part of the algorithm.

The SW-LSTM fault isolation network training method is shown in Fig. 4. Data of *InputLayer* are a window of data, and the relevant output is the corresponding fault type. Training formula is shown in Eqs. (23) and (24), and the window length is $l+1$ at this time.

$$\left\{ \begin{array}{c} T(2+l) \\ T(3+l) \\ \dots \\ T(t) \end{array} \right\} \approx \lambda \left\{ \begin{array}{c} [R(2), \dots, R(2+l)] \\ [R(3), \dots, R(3+l)] \\ \dots \\ [R(t-l), \dots, R(t)] \end{array} \right\} \quad (23)$$

$$\hat{T} \approx \lambda(R_{test}), \hat{T} \in \{fault(1), fault(2), \dots, fault(n)\} \quad (24)$$

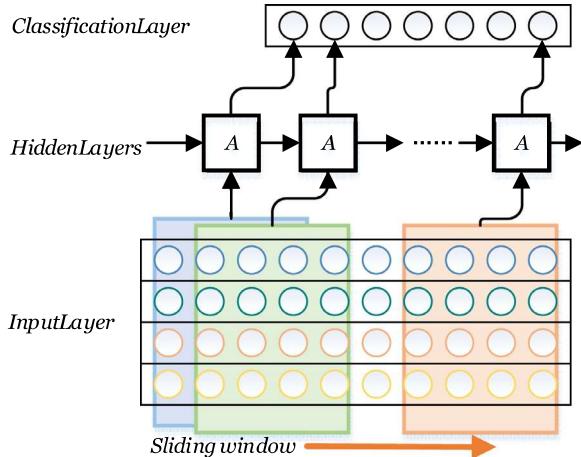


Fig. 4. SW-LSTM fault isolation algorithm.

3.3. Fault detection and isolation program

The algorithm is divided into two parts: model training, fault detection and isolation (FDI). The proposed process is shown in [Algorithm 1](#). Before the FDI, two models for detection and isolation need to be trained. The training data set includes a nominal dataset and a fault dataset. A cross-LSTM model is obtained by training the nominal dataset. The model precisely predicts the parameters at the next moment when it encounters normal working data. However, when an abnormal state occurs, the predicted result will deviate from the actual value, and the occurrence of the fault will be judged. The fault isolation step is performed after fault detected. Training this network requires the residuals of fault dataset that is biased by the trained cross-LSTM network. The residual data is used as input to train an SW-LSTM neural network, which can be the type of failure when encountering fault data.

4. Fault detection and isolation case studies

4.1. Introduction to fault diagnosis datasets

The algorithm used to verify the data from the NASA website is a dedicated actuator failure test system. The advantage of this test platform is that it can simulate the occurrence of faults just like a normal ground simulation test platform and it can also follow the aircraft to simulate the real working environment [34]. Datasets include nominal dataset and fault datasets. EMA faults can be divided into 4 types: mechanical/structural faults, motor

Table 3
Fault datasets for FDI algorithm verification.

Type	Dataset name
Mechanical/structure fault	Return channel jam (Jam)
Motor fault	Spall
Sensor fault	Motor failure (Motor)
	Position sensor failure (PosiDead)

Table 4
Sensors for training and testing.

Actuator Z position	Motor Y voltage
Measured Load	Motor X temperature
Motor X current	Motor Y temperature
Motor Y current	Motor Z temperature
Motor Z current	Nut X temperature
Motor X voltage	Nut Y temperature

faults, electrical/electronic faults, and sensor faults. The characteristics of electrical and electronic faults in the power and control systems of EMAs do not differ significantly from the same type of faults in other aerospace systems [1]. Therefore, four faults of other three fault-types are selected for FDI algorithm verification, as [Table 3](#) shown. Mechanical and structural faults are likely to be the main source of concern for EMA deployed in the demanding conditions of aerospace applications. Their main causes are excessive loads, environmental factors, lubrication issues, and manufacturing defects. In the experiment, two types of faults, return channel jam and spalling, are used to represent mechanical and structural faults. A jam in the return channel, caused, for example, by a piece of debris or a deformed ball, would stop that circulation and could lead to catastrophic consequences. And spalling refers to the development of indentations in metal surfaces at high-stress contact points. A severe case of a spall may result in metal flakes separating from the surface, creating potentially dangerous debris. Motor faults are the next most important category of EMA faults. Motors are often operated at high rotational rates, leading to increased temperatures within their housing and significant mechanical stresses, thus making them prone to developing winding shorts, rotor shaft eccentricities, and other problems. The sensor failure in this test is represented by the position sensor failure.

The sensors used for fault diagnosis are shown in [Table 4](#), including position, temperature, current and voltage.

Algorithm 1 Fault detection and isolation program using modified LSTM.

```

Training for LSTM Nets
Step1. Training data import
Step2. Normalization
Step3. for i = 1: sensors_num
    net_Detection = RegressionLSTM(nominal_data)% Training cross-LSTM net for every single sensor
Step4. for i = 1: sensors_num
    Residual = net_Detection (i)(fault_data)% Calculate residual to train fault isolation net
Step5. net_Isolation = ClassifyLSTM(Residual, Fault_type);% Training SW-LSTM net for isolation
Detection and Isolation
Step1. Testing Data import
Step2. Normalization
Step3. for j = 1:num
    Res = net_Detection (j) (test_data)- test_data% Calculating residual for fault detection
Step4. if Res > thr
    fault detected = true% Fault detection
Step5. if fault detected == true
    fault type = net_Isolation(Res);% Fault isolation

```

Table 5
Comparison of training accuracy.

	MSE_{train}	μ_{train}	σ_{train}	MSE_{test}	MSE_{train}	μ_{train}	σ_{train}	MSE_{test}
Actuator Z position				Measured load				
NARNN	0.0064	0.0048	0.0000	0.0114	0.0314	0.0164	0.0006	0.0570
RNN	0.0085	0.0065	0.0000	0.0104	0.0348	0.0193	0.0008	0.0349
LSTM network	0.0058	0.0045	0.0000	0.0060	0.0293	0.0149	0.0006	0.0275
Cross-LSTM network	0.0051	0.0039	0.0000	0.0092	0.0233	0.0163	0.0003	0.0310
Motor X current				Motor Y current				
NARNN	0.1347	0.1062	0.0069	0.1356	0.0812	0.0599	0.0030	0.0946
RNN	0.1357	0.1072	0.0069	0.1355	0.0465	0.0337	0.0010	0.0712
LSTM network	0.1341	0.1056	0.0068	0.1358	0.0625	0.0442	0.0020	0.0701
Cross-LSTM network	0.1286	0.1009	0.0063	0.1475	0.0491	0.0328	0.0013	0.0531
Motor Z current				Motor X voltage				
NARNN	0.0700	0.0429	0.0032	0.0710	0.1195	0.0833	0.0073	0.1461
RNN	0.0381	0.0240	0.0009	0.0655	0.0737	0.0500	0.0029	0.0915
LSTM network	0.0366	0.0222	0.0008	0.0313	0.0976	0.0678	0.0049	0.1226
Cross-LSTM network	0.0455	0.0281	0.0013	0.0469	0.0781	0.0571	0.0028	0.1350
Motor Y voltage				Motor X temperature				
NARNN	0.1813	0.1203	0.0187	0.1782	0.0853	0.0612	0.0035	0.0737
RNN	0.0831	0.0627	0.0030	0.0852	0.0857	0.0611	0.0036	0.0732
LSTM network	0.1185	0.0842	0.0069	0.1264	0.0857	0.0615	0.0036	0.0732
Cross-LSTM network	0.0796	0.0553	0.0033	0.1070	0.0823	0.0599	0.0032	0.0765
Motor Y temperature				Motor Z temperature				
NARNN	0.1060	0.0667	0.0068	0.2001	0.0618	0.0432	0.0019	0.0595
RNN	0.1072	0.0675	0.0069	0.1999	0.0644	0.0451	0.0021	0.0612
LSTM network	0.1059	0.0669	0.0067	0.2130	0.0620	0.0435	0.0020	0.0595
Cross-LSTM network	0.0999	0.0648	0.0058	0.2156	0.0610	0.0432	0.0019	0.0622
Nut X temperature				Nut Y temperature				
NARNN	0.1280	0.0989	0.0066	0.1324	0.1363	0.1013	0.0083	0.1425
RNN	0.1341	0.1038	0.0072	0.1388	0.1086	0.0687	0.0071	0.1168
LSTM network	0.1279	0.0985	0.0066	0.1329	0.1212	0.0891	0.0068	0.1323
Cross-LSTM network	0.1230	0.0946	0.0062	0.1387	0.1001	0.0726	0.0047	0.1399

4.2. Time series modeling accuracy

In order to certify the effectiveness of the proposed cross-LSTM algorithm, we chose the classical NARNN [43,44], RNN and the standard LSTM neural network as the comparison algorithm. LSTM and cross-LSTM network are both contain one hidden layer with 50 hidden nodes. Training data's length is 1500 points. Training accuracy is measured with training Mean Square Error (MSE), mean (μ), variance (σ), and MSE for test data.

The comparison results of the four methods are shown in Table 5. The average value of training MSEs of the ten detection

sensors was calculated. They are 0.1 for RNN, 0.0767 for NARNN, 0.0823 for LSTM network, and 0.0720 for cross-LSTM network separately. The NARNN, RNN, and LSTM of our choice are all excellent in data fitting ability, but their overfitting effect is desperate. Overfitting affects the ability to predict in other data outside of the training data set. Therefore, in order to verify the overfitting phenomenon, 18,000 test points were used to test the effect of neural network fitting, results of measured load data are shown in Fig. 5. By comparing to the NARNN and RNN algorithms, the LSTM and cross-LSTM algorithms have a better ability to fit the test dataset.

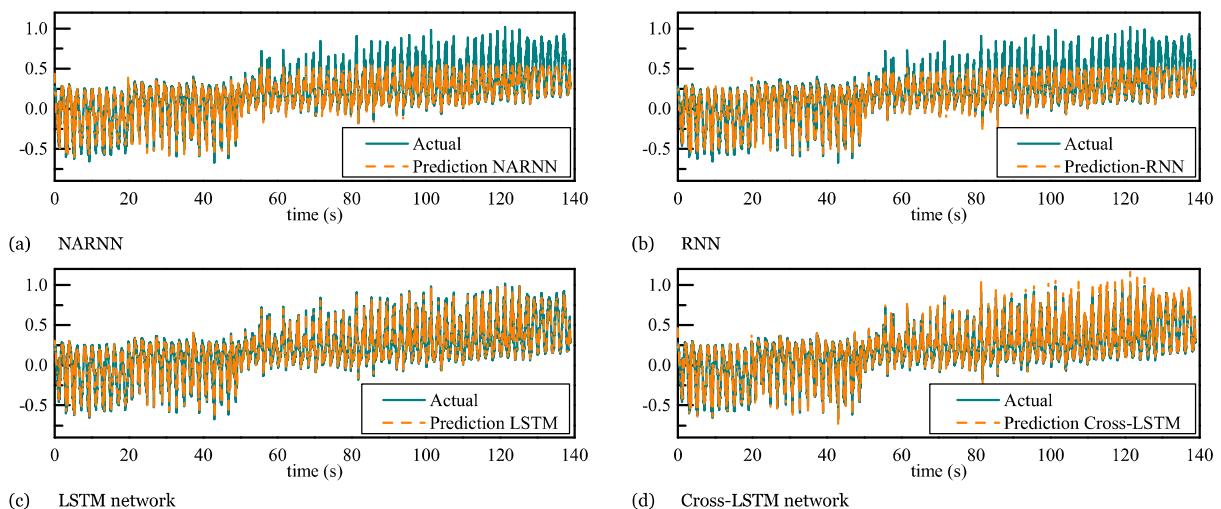


Fig. 5. Fitting effect of measured load data.

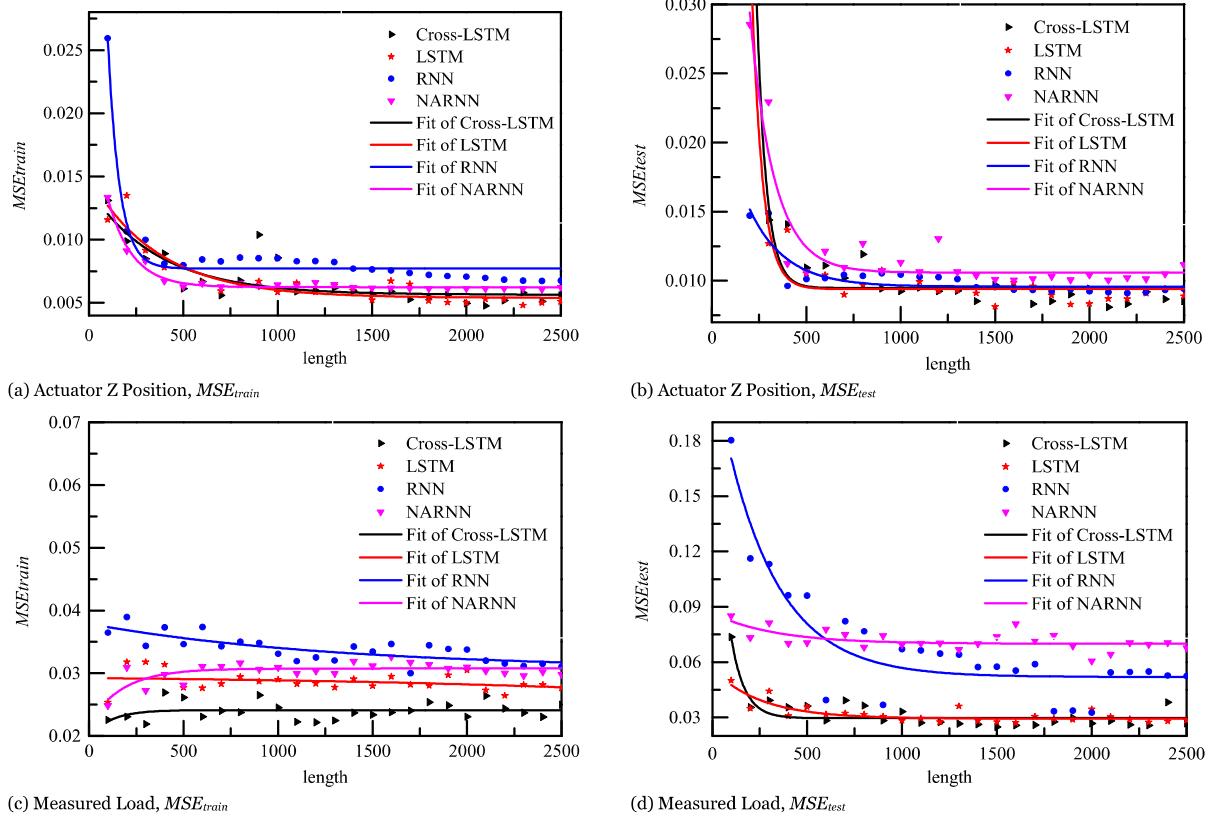


Fig. 6. Comparison of modeling effects under different training data lengths.

Table 6
Comparison of $MAE_{fault} / MAE_{train}$ value.

	Jam		Spall		PosiDead		Motor	
	LSTM	Cross- LSTM	LSTM	Cross- LSTM	LSTM	Cross- LSTM	LSTM	Cross- LSTM
Actuator Z position	1.00	6.12	1.24	6.61	42.38	51.17	3.77	9.08
Measured load	0.96	3.50	1.23	3.57	0.89	10.68	0.19	2.74
Motor X current	30.10	32.22	29.45	31.74	1.27	2.37	8.67	9.59
Motor Y current	0.20	7.60	0.19	7.92	3.29	7.14	0.19	1.53
Motor Z current	1.01	4.66	1.29	5.40	2.46	13.61	0.40	2.95
Motor X voltage	18.63	18.15	24.10	21.94	0.18	7.76	4.62	4.98
Motor Y voltage	0.77	3.85	1.15	3.43	3.33	5.05	2.83	5.90
Motor X temperature	4.27	4.89	4.58	4.68	79.07	83.11	1.52	1.65
Motor Y temperature	5.76	6.76	1.20	2.76	1.12	2.27	3.28	5.14
Motor Z temperature	4.03	4.51	4.59	4.88	4.09	10.88	8.21	8.32
Nut X temperature	5.14	6.19	5.08	6.57	27.37	31.10	1.56	1.69
Nut Y temperature	3.76	4.49	0.54	2.70	1.91	18.83	2.23	3.82

Training accuracy and testing accuracy of the time series model at different training lengths is shown in Fig. 6, and the lines in this figure are the approximates lines of actuator z position and measured load sensor data. According to the results of actuator Z position, NARNN has better training performance than LSTM and cross-LSTM algorithms when the number of training data is less than 1000. Yet, the training accuracy of LSTM and cross-LSTM algorithms becomes better and reaching stability when training data increases. However, the fitting accuracy of the test data of the NARNN algorithm is not as good as the other three algorithms. The fitting accuracy of the measured load is different from that of the actuator Z position. Cross-LSTM method is superior to NARNN and RNN in both training accuracy and testing accuracy. Meanwhile, its test accuracy is similar to the LSTM algorithm, and better when the number of training samples is low.

4.3. Fault detection

The mechanism of fault detection is possible that the gap between the data from the detection model and the actual data becomes larger. This change is calculated by the Mean Absolute Error (MAE) of the residual data in training stage. From the comparison in Section 4.2, we believe that time series modeling of LSTM and cross-LSTM methods is better. Therefore, this section only compares the fault detection effect of these two methods.

In Table 6, the ratio of the MAE of the fault residuals (MAE_{fault}) and the MSE of the training residuals (MAE_{train}) is given, including residuals and four faults of all ten sensors. For the four failures, the average growth ratio of the 10-sensor data is 2.28, 2.30, 6.39 and 1.66. Especially for some sensor parameters whose residual error is not obvious under the LSTM algorithm, the cross-LSTM algorithm

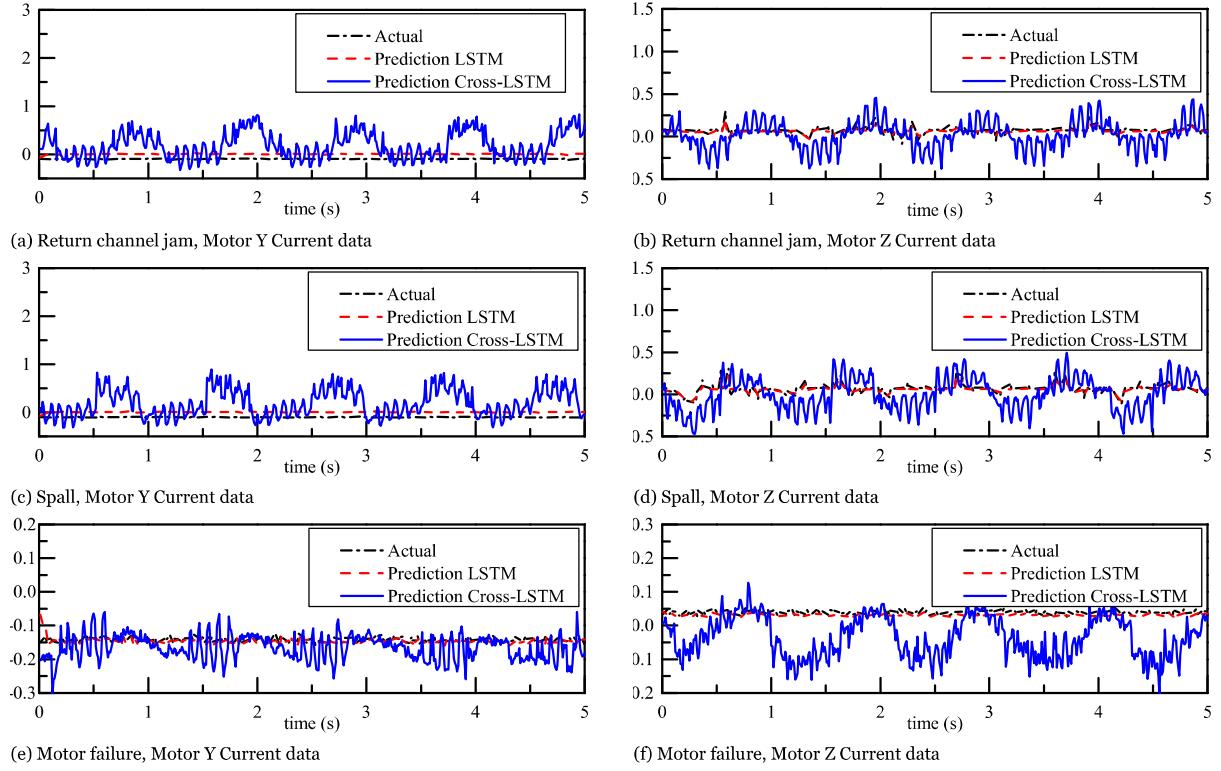


Fig. 7. Comparison of model prediction results under fault status.

Table 7
Fault detection thresholds.

Sensor	<i>t.h.lower</i>	<i>t.h.upper</i>	Sensor	<i>t.h.lower</i>	<i>t.h.upper</i>
Actuator Z position	0.0152	-0.0152	Motor Y voltage	0.2746	-0.2753
Measured load	0.0854	-0.0854	Motor X temperature	0.2369	-0.2401
Motor X current	0.3662	-0.3664	Motor Y temperature	0.3807	-0.3813
Motor Y current	0.1795	-0.1799	Motor Z temperature	0.1540	-0.1519
Motor Z current	0.1809	-0.1778	Nut X temperature	0.3702	-0.3691
Motor X voltage	0.2445	-0.2449	Nut Y temperature	0.3435	-0.3444

can significantly increase the variation of the residual error, for example, actuator Z position data and motor Y current data of return channel jam and spall, measured load, motor Z Current, motor X voltage data of position sensor failure, actuator Z position and motor Y voltage of motor failure.

In Fig. 7, the measured data and prediction data under fault conditions obtained by the two time series models of LSTM and cross-LSTM are compared. The figure displays the comparison of motor Y current data and motor Z current data. The characteristics of the fault data's residuals under the two algorithms are totally different. The Cross-LSTM method has a greater gap between the predicted and measured values when a fault occurs, making it easier to determine the fault. Therefore, compared with the LSTM algorithm, the cross-LSTM method is most effective when used in the time series modeling based fault detection methods.

According to the method given in Section 2.3, the fault detection threshold is determined. The threshold is determined by the 99% confidence interval [*t.h.lower*, *t.h.upper*] of the training residual data. Fault detection thresholds for ten sensors are shown in Table 7. The position sensor failure detection result of six sensor data can be seen in 0. The residual data of sensors actuator Z position, measured load, motor Z current, motor X voltage, and motor Z temperature exceed the threshold [*t.h.lower*, *t.h.upper*] of fault detection when a position sensor failure occurs. Therefore, the fault can be detected when it happens (Fig. 8).

Table 8
Comparison of isolation performance using the different classifier.

	Train accuracy	Test accuracy
Softmax	94.7 ± 0%	93.7 ± 0%
SVM	94.4 ± 0%	93.6 ± 0%
LSTM	94.75% ± 0.17%	94.34% ± 0.22%
SW-LSTM (length=3)	99.18% ± 0.66%	99.40% ± 0.68%
SW-LSTM (length=5)	99.82% ± 0.22%	99.89% ± 0.29%
SW-LSTM (length=10)	99.93% ± 0.27%	99.96% ± 0.16%

4.4. Fault isolation

The fault isolation process occurs after the fault is detected. The algorithm utilizes the residual data and the SW-LSTM algorithm mentioned in Section 3.3 to implement fault classification. For each fault, 500 training points were supposed to train the classification model, and training maximum iterations epoch is 30. The total length of the training data is 2000. Fig. 9 compares the training accuracy of the LSTM algorithm and the proposed SW-LSTM algorithm in iteration training process. Training with LSTM can hardly achieve 100% classification accuracy. When using a single point for classification, the amount of information for classification training is small. Therefore, the classification accuracy is difficult to achieve a satisfactory level. However, by adding a sliding window, the training accuracy can reach 100% quickly. This is can be explained by the fact that the use of a sliding window is equiv-

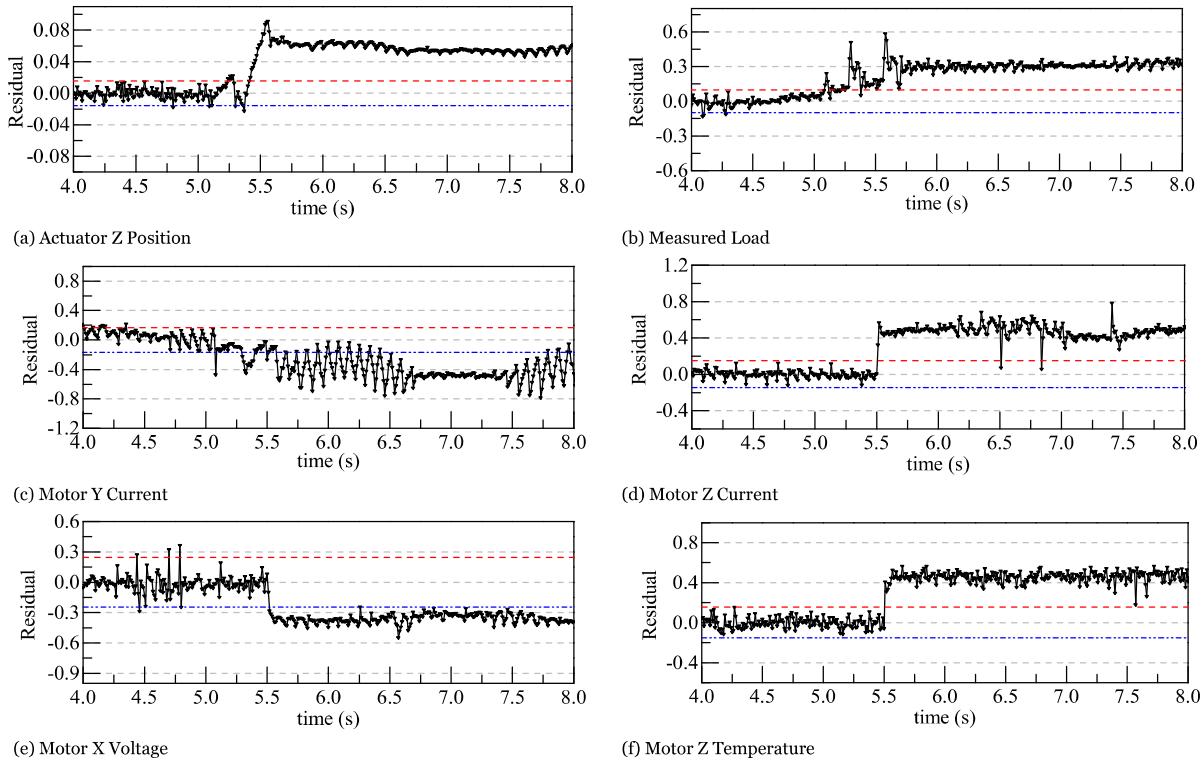


Fig. 8. The detection result of position sensor failure.

alent to considering the relationship between the preceding and succeeding sequences in the classification, so the SW-LSTM algorithm is superior to the LSTM algorithm for fault isolation. Comparing the model training accuracy under different window lengths in Fig. 9, the longer the window length is, the fewer iterations are needed in order to achieve 100% training accuracy.

In addition, we compared the SW-LSTM algorithm for fault isolation under different window lengths. Fig. 10 gives the fault isolation accuracy of the training data and test data with both 500 points. The results show that as the window length becomes longer, the fault classification ability becomes stronger. When the window length exceeds 5, the improvement of the fault classification becomes slower.

In statistical learning theory, support vector machines (SVM) are typically used for classification because it works very well [45–47]. Its central idea is to apply the principle of structural risk minimization to the field of classification. In order to identify the accuracy of the model, 2000 fault points were used for model training accuracy testing. We compare the training accuracy and test accuracy in Table 8. Each of these classification algorithms

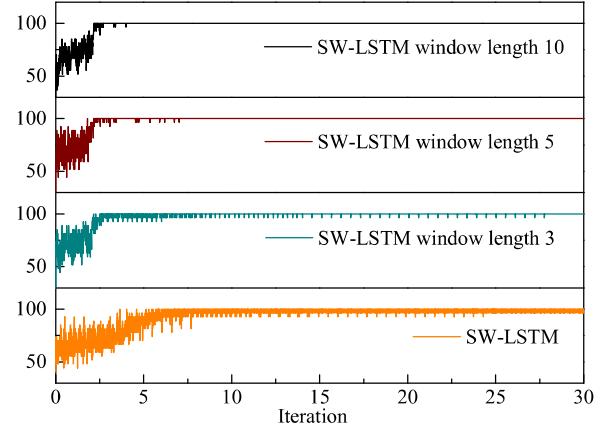


Fig. 9. The accuracy of isolation model training during the iteration process.

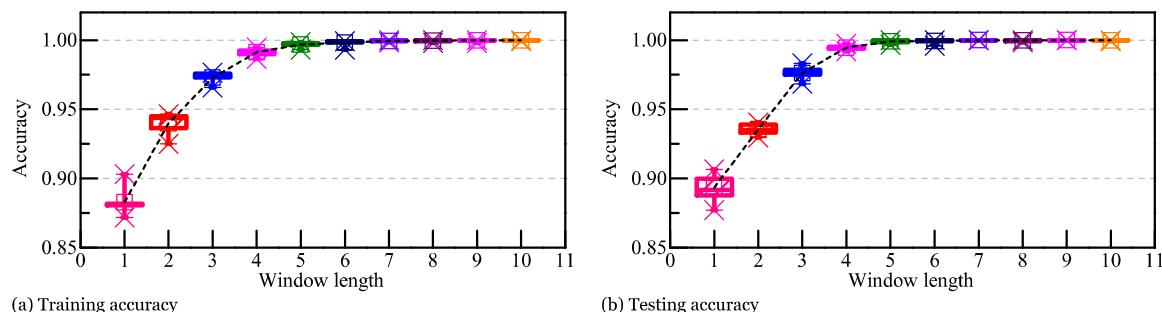


Fig. 10. Classification accuracy for the different window length of SW-LSTM.

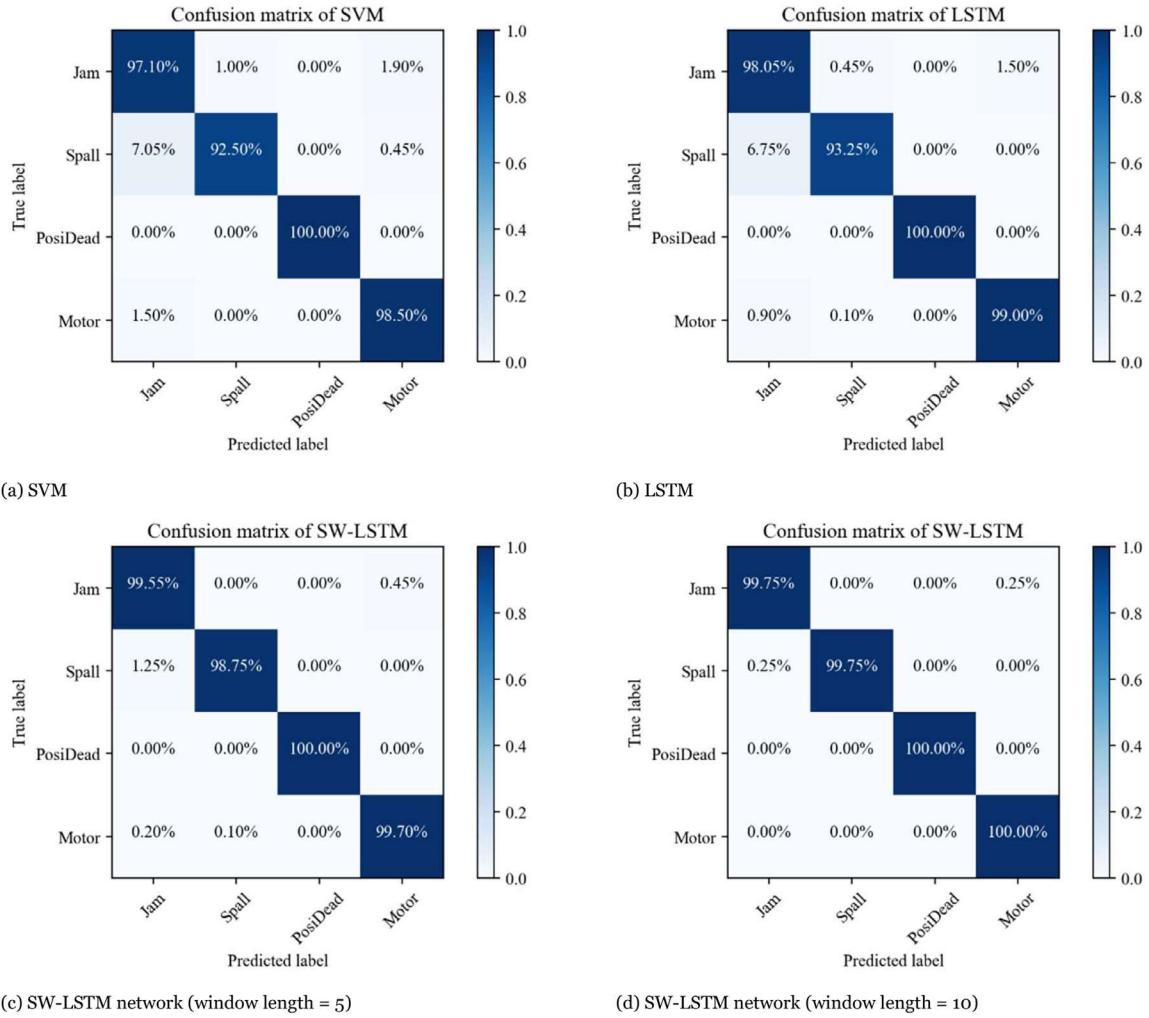


Fig. 11. Confusion matrix of different classification algorithms including SVM, LSTM, and SW-LSTM with different window width.

was trained ten times in order to obtain the distribution of its training accuracy. Maximum iteration epoch for LSTM and SW-LSTM is 30. In addition, the SVM isolation algorithm, the standard LSTM isolation method, and the SW-LSTM isolation method with window lengths of 5 and 10 are compared, shown in Fig. 11.

Compared with the SVM algorithm, LSTM method can improve the accuracy of fault isolation. Further, fault isolation of the SW-LSTM method is more effective than the former two methods, especially when the window length is 10. However, the window should not be too lengthy, because if the window length is too large, the classification speed will slow down, and the isolation accuracy will not increase much.

5. Conclusion

This paper innovatively applies the LSTM neural network to the tasks of fault detection and isolation for EMAs in the aerospace field. Our findings indicate that when using the LSTM neural network for time series modeling, the cross-LSTM neural network with cross input performs better than the standard version. Moreover, when this method predicts each of the four example EMA faults, the gap between the estimated value and the measured value reports as significantly larger than that of the standard LSTM algorithm. Specifically, with respect to the ratio of the residual MAE for the fault data to the training residual MAE, the results of the cross-LSTM method proposed in this paper is an average of

3.17 higher than the LSTM. This result suggests that there is a benefit, in terms of predictive value for fault detection, introduced by including the sensor information. By considering the relationship between sensors in the model, it undoubtedly improves its predictive accuracy. Also, applying a sliding window on the LSTM isolation network delivers significantly better classification accuracy than both the SVM and standard LSTM network approaches. This leads to fewer training iterations and superior isolation accuracy.

Conflict of interest

None.

References

- [1] E. Balaban, P. Bansal, P. Stoelting, A. Saxena, K. Goebel, S. Curran, A diagnostic approach for electro-mechanical actuators in aerospace systems, in: Proceedings of Aerospace Conference, IEEE, Big Sky, MT, USA, 2009, pp. 1–13.
- [2] D. Ossmann, F.L.J.v.d. Linden, Advanced sensor fault detection and isolation for electro-mechanical flight actuators, in: Proceedings of the 2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2015, pp. 1–8.
- [3] D. Arriola, F. Thielecke, Model-based design and experimental verification of a monitoring concept for an active-active electromechanical aileron actuation system, *Mech. Syst. Signal Process.* 94 (2017) 322–345.
- [4] Y. Zhang, L. Gao, X. Li, P. Li, A novel data-driven fault diagnosis method based on deep learning, in: Proceedings of International Conference on Data Mining and Big Data, Springer International Publishing, 2017, pp. 442–452.
- [5] L. Feiya, W. Chenglin, B. Zejing, L. Meiqin, Fault diagnosis based on deep learning, in: Proceedings of the 2016 American Control Conference (ACC), IEEE, Boston, MA, USA, 2016, pp. 6851–6856.

- [6] S. Sina Tayarani-Bathaie, Z.N. Sadough Vanini, K. Khorasani, Dynamic neural network-based fault diagnosis of gas turbine engines, *Neurocomputing* 125 (2014) 153–165.
- [7] M. Amozegar, K. Khorasani, An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines, *Neural Netw.* 76 (2016) 106–121.
- [8] M. Mazzoleni, S. Formentin, F. Previdi, S.M. Savaresi, Fault detection via modified principal direction divisive partitioning and application to aerospace electro-mechanical actuators, in: Proceedings of the 53rd IEEE Conference on Decision and Control, 2014, pp. 5770–5775.
- [9] W.S. Craig, Data Driven Approach to Non-Stationary EMA Fault Detection and Investigation into Remaining Useful Life, Mechanical Engineering, Rochester Institute of Technology, 2014.
- [10] J. Chen, L. Wang, Electromechanical actuator modeling and its application in fault diagnosis, in: Proceedings of the 2018 International Conference on Mechanical, Electronic and Information Technology, 2018, pp. 223–228.
- [11] H. Liu, J. Jing, J. Ma, Fault diagnosis of electromechanical actuator based on VMD multifractal detrended fluctuation analysis and PNN, *Complexity* 2018 (2018) 1–11.
- [12] T. Yaoyao, Z. Huijuan, Y. Zhong, L. Xiaoming, Z. Huibin, Fault diagnosis of electromechanical actuator based on wavelet packet and SOM neural network, *Appl. Sci. Technol.* 45 (2018) 1–6.
- [13] H. Zhang, Z. Liu, G.B. Huang, Z. Wang, Novel weighting-delay-based stability criteria for recurrent neural networks with time-varying delay, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (2013) 513–521.
- [14] H. Zhang, F. Yang, X. Liu, Q. Zhang, Stability analysis for neural networks with time-varying delay based on quadratic convex combination, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (2013) 513–521.
- [15] H. Zhang, Z. Wang, D. Liu, Global asymptotic stability of recurrent neural networks with multiple time-varying delays, *IEEE Trans. Neural Netw.* 19 (2008) 855–873.
- [16] Z. Wang, H. Zhang, B. Jiang, LMI-based approach for global asymptotic stability analysis of recurrent neural networks with various delays and structures, *IEEE Trans. Neural Netw.* 22 (2011) 1032–1045.
- [17] M. Mrugalski, M. Luzar, M. Pazera, M. Witczak, C. Aubrun, Neural network-based robust actuator fault diagnosis for a non-linear multi-tank system, *ISA Trans.* 61 (2016) 318–328.
- [18] H.A. Talebi, K. Khorasani, S. Tafazoli, A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite's attitude control subsystem, *IEEE Trans. Neural Netw.* 20 (2009) 45–60.
- [19] Y.S. Sun, Y-m Li, G-zhang, H-b Wu, Actuator fault diagnosis of autonomous underwater vehicle based on improved Elman neural network, *J. Central South Univ.* 23 (2016) 808–816.
- [20] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to Forget: continual prediction with LSTM, *Neural Comput.* 12 (2000) 2451–2471.
- [21] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: a search space odyssey, *IEEE Trans. Neural Netw. Learn. Syst.* 28 (2017) 2222–2232.
- [22] R. Mahdaoui, L.H. Mouss, A TSK-type recurrent neuro-fuzzy systems for fault prognosis, *J. Softw. Eng. Appl.* 05 (2012) 477–482.
- [23] M. Cai, J. Liu, Maxout neurons for deep convolutional and LSTM neural networks in speech recognition, *Speech Commun.* 77 (2016) 53–64.
- [24] V. Peddinti, Y. Wang, D. Povey, S. Khudanpur, Low latency acoustic modeling using temporal convolution and LSTMs, *IEEE Signal Process. Lett.* 25 (2018) 373–377.
- [25] A. Farzad, H. Mashayekhi, H. Hassanpour, A comparative performance analysis of different activation functions in LSTM networks for classification, *Neural Comput. Appl.* (2017) 1–15.
- [26] A. Rao, N. Spasojevic, Actionable and Political Text Classification Using Word Embeddings and LSTM, (2016).
- [27] C. Zhou, C. Sun, Z. Liu, F.C.M. Lau, A C-LSTM neural network for text classification, *Comput. Sci.* 1 (2015) 39–44.
- [28] J. Schmidhuber, Deep learning in neural networks: an overview, *Neural Netw.* 61 (2015) 85–117.
- [29] E. Chemali, P.J. Kollmeyer, M. Preindl, R. Ahmed, A. Emadi, Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries, *IEEE Trans. Ind. Electron.* 65 (2018) 6730–6739.
- [30] M. Yuan, Y. Wu, L. Lin, Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network, in: Proceedings of the 2016 IEEE International Conference on Aircraft Utility Systems (AUS), 2016, pp. 135–140.
- [31] A. Elsaid, B. Wild, J. Higgins, T. Desell, Using LSTM recurrent neural networks to predict excess vibration events in aircraft engines, in: Proceedings of the 2016 IEEE 12th International Conference on e-Science (e-Science), 2016, pp. 260–269.
- [32] Y. Wu, M. Yuan, S. Dong, L. Lin, Y. Liu, Remaining useful life estimation of engineered systems using vanilla LSTM neural networks, *Neurocomputing* 275 (2018) 167–179.
- [33] Y. Li, H. Zhang, X. Liang, B. Huang, Event-triggered based distributed cooperative energy management for multi-energy systems, *IEEE Trans. Ind. Inform.* 15 (2019) 2008–2022.
- [34] E. Balaban, A. Saxena, S. Narasimhan, I. Roychoudhury, K. Goebel, M.T Koopmans, Airborne electro-mechanical actuator test stand for development of prognostic health management systems, in: Proceedings of Annual Conference of the Prognostics and Health Management Society, 2010, pp. 1–13.
- [35] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (1997) 1735–1780.
- [36] P. Saha, S. Ghorai, B. Tudu, R. Bandyopadhyay, N. Bhattacharyya, Tea quality prediction by autoregressive modeling of electronic tongue signals, *IEEE Sens. J.* 16 (2016) 4470–4477.
- [37] S. Braun, E.A.P. Habets, Online dereverberation for dynamic scenarios using a Kalman filter with an autoregressive model, *IEEE Signal Process. Lett.* 23 (2016) 1741–1745.
- [38] K.S. Tuncel, M.G. Baydogan, Autoregressive forests for multivariate time series modeling, *Pattern Recognit.* 73 (2018) 202–215.
- [39] Y. Shi, K. Yao, L. Tian, D. Jiang, Deep LSTM based feature mapping for query classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, California, San Diego, 2016, pp. 1501–1511.
- [40] Q. Qian, M. Huang, J. Lei, X. Zhu, Linguistically regularized LSTM for sentiment classification, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1679–1689.
- [41] H. Sak, A. Senior, F. Beaufays, Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling, (2014).
- [42] M. Sundermeyer, R. Schlüter, H. Ney, LSTM neural networks for language modeling, *Interspeech* (2012) 194–197.
- [43] A. Rai, S.H. Upadhyay, The use of MD-CUMSUM and NARX neural network for anticipating the remaining useful life of bearings, *Measurement* 111 (2017) 397–410.
- [44] H. Santosa, Y. Hobara, One day prediction of nighttime VLF amplitudes using nonlinear autoregression and neural network modeling, *Radio Sci.* 52 (2017) 132–145.
- [45] L. Pellaco, P. Costamagna, A. De Giorgi, A. Greco, L. Magistri, G. Moser, A. Trucco, Fault diagnosis in fuel cell systems using quantitative models and support vector machines, *Electron. Lett.* 50 (2014) 824–826.
- [46] N. Shafabady, L.H. Lee, R. Rajkumar, V.P. Kallimani, N.A. Akram, D. Isa, Using unsupervised clustering approach to train the support vector machine for text classification, *Neurocomputing* 211 (2016) 4–10.
- [47] X.J. Shen, L. Mu, Z. Li, H.X. Wu, X. Chen, X. Chen, Large-scale support vector machine classification with redundant data reduction, *Neurocomputing* 172 (2016) 189–197.



Jing Yang received her B.S. and M.S. degrees from Xi'an Jiaotong University in 2008 and 2011, majored in signal and information processing. She is currently a Ph.D. candidate at Northwestern Polytechnical University, China. Her research interests include fault diagnosis and prediction of complex systems.



Yingqing Guo received a Ph.D. in Engineering from Northwestern Polytechnical University, China in 2000. He has been a professor of Power and Energy School in Northwestern Polytechnical University since 2003. His main research interests are advanced engine control technology, system status monitoring, and health management technology. He is Senior Member of the American Aerospace Association (AIAA), and member of the Engine Control Professional Committee of the China Aviation Society.



Wanli Zhao received his B.S. degree in Harbin Engineering University in 2017. He is currently a graduate student for M.S. degree of Power and Energy Engineering at Northwestern Polytechnical University, China. Mr. Zhao's research interests are in the area of gas turbine fault diagnosis and hardware design in loop simulation.