Article - Measurement

TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Transactions of the Institute of Measurement and Control
2021, Vol. 43(11) 2551–2566
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/01423312211005612
journals.sagepub.com/home/tim
SAGE

Novel topology convolutional neural network fault diagnosis for aircraft actuators and their sensors

Ruonan Wei[1], Ju Jiang[1], Haiyan Xu[2] and Danmeng Zhang[1]

Abstract

The variable working conditions and frequent turns make the aircraft actuator system prone to failure, seriously threatening flight safety. The identification of the airplane actuator system is critical for flight decisions and safety. Most fault diagnosis methods of actuators only focus on the actuators themselves, ignoring the disturbance caused by the fault of the actuator position sensor, which may easily lead to wrong decisions. In order to distinguish the actuator fault from its position sensor fault and identify the fault type accurately, an offline diagnosis method of convolutional neural network (CNN) with novel topology for processing time series is proposed. A new shift layer is added after the input layer, which avoids the loss of a large number of features due to the direct connection between the time series and the convolutional layer. A local topological network learning complex pattern with inception module is designed to improve the diagnostic accuracy in different working conditions. The wide residual structure is introduced to expand the convolutional channel, which allows the network features at the bottom level to propagate directly to the top level to prevent network degradation. Simulation results show that this method can accurately diagnose the actuator fault and its position sensor, with an average accuracy of 96.8%. Compared with the current mainstream data-driven methods, the precision and recall are increased by 6.3% and 6.7% respectively.

Keywords

Fault diagnosis, inception, wide residual, convolutional neural network, actuator, position sensor

Introduction

The aircraft actuator system is an important part of maintaining the normal flight of aircraft. However, the variable working environment and frequent control tasks make the actuator system prone to multiple failures and cause serious security risks for aircraft. Faults include being stuck in the fixed position (stuck fault), random drift in the normal position (floating fault), reduced efficiency of acting on the aircraft (damage fault), etc. Different types of fault have different effects on flight safety, so fault identification is of great significance to pilot or controller decision making.

Distinguishing actuators' faults from their position sensors' fault is one of the challenges of this study. In the fault diagnosis of the actuator, the angle signal of the actuator is the key diagnostic variable (Avram et al., 2017; Zhang, 2018). The acquisition of actuator angle depends on the position sensor, but most studies ignore the fault of the position sensor and take the position signal output by the sensor as the true position of the actuator or its unbiased estimation (Hassanabadi et al., 2017; Li et al., 2017; Liu et al., 2018). Some sensor faults, such as complete fault, fixed deviation and gain loss, are similar to the fault modes of the actuator (stuck, floating, damage), which are difficult to recognize. The fault is regulated by a closed-loop flight control system and loses the inherent performance mode under open-loop response. Considering only the actuator as the source of fault is likely to produce erroneous results. Sensor fault and actuator fault have different effects on flight. It is vital to distinguish between sensor fault and actuator fault and identify the type of fault for pilot or driving system decisions.

The data-driven fault diagnosis method has strong adaptability (Tidriri et al., 2016). This has been a hot research topic in recent years and has achieved many results (Mao et al., 2017; Youssef et al., 2016; Zhang et al., 2018). Parity vectors are derived in the literature (Jiang et al., 2021) to construct an optimal residual generator for both linear and nonlinear systems fault detection. The common data-driven fault diagnosis method is first to extract specified features based on experience, then select a classifier to perform pattern recognition based on the features. Support vector machine (SVM) is a classic machine learning algorithm, which effectively solves the problem of fault classification under small samples (Ji et al., 2018; Qin et al., 2016). Jiang et al. (2019) design a prognosis and diagnosis system oriented to the data-driven key-performance-indicators approaches. Cerrada et al. (2018)

[1]College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, China
[2]College of Economic and Management, Nanjing University of Aeronautics and Astronautics, China

Corresponding author:
Ju Jiang, College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 211106, China.
Email: jiangju@nuaa.edu.cn

utilize the wavelet packet decomposition method to extract features from vibration signals, using K nearest neighbour (KNN) and K-means to process marked and unmarked samples to complete gearbox fault diagnosis. However, the shallow model of traditional machine learning is difficult to characterize the complex mapping relationship between the input sample signal and the object's state. At the same time, there are problems such as dimensional disaster and overfitting.

In the field of deep learning, neural networks can automatically construct features and learn the mapping of features to the target space. Li et al. (2017) use variational mode decomposition to decompose the vibration signal into intrinsic mode functions (IMF), then extract the multifractal features in IMFs, perform principal component analysis for feature selection, and finally input a probabilistic neural network to classify the fault modes. Razavi-Far et al. (2019) convert the signal into a high-dimensional space by singular value decomposition and develop a semi-supervised deep learning scheme to diagnose various gearbox defects. These fault diagnoses based on artificial feature extraction achieve a high accuracy, but feature extraction depends on the experience of the designer. Limited by the subjective consciousness, some information will be lost, and complex feature extraction will also reduce the efficiency of the algorithm.

Establishing a deep learning fault diagnosis model with strong generalization ability without relying on prior knowledge is another challenge of this paper. Recurrent neural networks (RNN) such as long short-term memory (LSTM) neural network are often used to deal with timing problems (Song et al., 2018; Zhao et al., 2017). Because these time series have low sampling frequency, the number of extrapolated data points does not need to be particularly large, so RNNs have good performance. However, in this study, the sampling frequency of the position sensor signal is generally a few hundred Hz or even higher. Thousands of points need to be processed for prediction and data extrapolation for a few seconds, and the recurrent neural network cannot effectively estimate such a high-dimensional time series. The convolutional neural network (CNN) can extract features of high-dimensional data adaptively through hidden convolutional layers, which have made great achievements in image processing and have also been applied in fault diagnosis. Guo et al. (2018) and Wang et al. (2017) use short-time Fourier transform to convert vibration signals into three-dimensional time-frequency maps, extract several two-dimensional views from them, and construct traditional CNN to extract image features to classify signals. Yang et al. (2019a) extract time-frequency features by Hilbert-Huang transform and wave, then evaluate the state of the damper. The process of converting the signa into a time-frequency graph is limited by the window function, which may affect the features' expression in certain aspects. Various transformations have different characteristics, but at present, there is no clear applicable criterion for different objects. Yang et al. (2019b) propose a remaining useful life prediction method based on a double-CNN model architecture with no need to resort to any feature extractor. Levent et al. (2019) adopt a one-dimensional convolutional neural network for rapid fault diagnosis of induction bearings, get rid of the algorithm's dependence on prior knowledge, and achieve relatively effective fault diagnosis in the case of a small data set. However, the one-dimensional convolution easily loses the correlation between row data when processing complex signals. Although the double CNN can directly process the time series, it adds much calculation. The study of Zhang et al. (2020a) shows that replacing the traditional convolutional structure with stacked expansion convolution can obtain larger-scale related information at fewer levels, and the stacked CNN model can adapt to changes in the working load in the mechanical environment.

For neural networks, the processing of complex patterns requires the construction of more or wider hidden layers to increase the generalization ability of the network. Appropriately increasing the neural network size can improve the accuracy rate, but the number of layers of the network almost doubles for every percentage point increase in accuracy (Girshick et al., 2014). The gradient disappearance caused by the increase in depth or width will make training difficult to converge. When the depth reaches a certain level, it may also cause network degradation.

Faced with the above challenges, in view of the complexity of the aircraft actuator system fault and its own uncertainty, this paper proposes a new convolutional neural network fault diagnosis algorithm, whose contributions are mainly as follows:

(1) The fault of the actuator position sensor, which is often overlooked in aircraft actuator fault diagnosis, is considered. The diagnosis efficiency is improved, and the fault type can be diagnosed while distinguishing the fault parts.
(2) A shift layer is proposed to add after the input layer of CNN to avoid direct convolution of the measurement signal with the kernel. Continuously shifting the time signal through the misalignment layer extracts periodic fault information between non-adjacent signals, overcomes the weakness of CNN in time correlation and enables CNN to process time series better.
(3) By designing two inception modules in the convolutional layer (Zhang et al., 2020b), the sparse matrix is clustered into relatively dense sub-matrices, and an excellent local topology network is established. In the case of increasing the number of equal parameters, the depth and width are fully increased, thereby increasing the network's generalization ability and improving the computing performance.
(4) Inspired by the ILSVRC competition and literature (Zagoruyko and Komodakis, 2017), this paper introduces a wide residual structure. Through the shortcut connection between the shallow layer and the high layer, the entire network can still be trained through end-to-end back propagation, thereby solving the problem of the disappearance of the gradient caused by the increase in the depth and width of the neural network.

The remaining sections of this paper are summarized as follows. In the next section, the new CNN fault diagnosis algorithm proposed in this paper is designed. In the following section, the parameters of the neural network are optimized. Next, the simulation model is established and the data set is obtained. In the penultimate section, the fault diagnosis experiment and result analysis are carried out and compared fully with other algorithms. Finally, the conclusion of this paper is given.

The structure of the novel fault diagnosis CNN

Shift layer

CNN mainly focuses on spatial local feature extraction and pattern recognition. However, the mechanical signal is different from the image signal, and its characteristic is reflected in the temporal correlation. If the mechanical signal is directly input into the CNN, part of the time correlation characteristics may be lost. In this paper, a shift layer is proposed to help increase CNN's ability to process time series. A shift layer is added after the input layer, and the time signal is converted into a two-dimensional space signal according to the shift method to compensate for the limitations of traditional CNN. For each one-dimensional time series, the adjustment of the shift layer can be expressed as

$$D(r,:) = x(t_{r1} : t_{r2}) \tag{1}$$

where

$$t_{r1} = s(r-1)$$

$$t_{r2} = t_{r1} + w_x$$

$$r = 1, 2, \dots, \left\lfloor \frac{len_x - w_x}{w_x - s} \right\rfloor + 1$$

where D is the output matrix of the shift layer, r is the number of times the signal is intercepted, len_x is the length of the sequence x, w_x is the length of the intercept window, s is the length of the window movement. After the signal is intercepted by the window, it becomes a row of D. Each time the signal is intercepted, the window is shifted backward with time points and then intercepted. The graphical demonstration is shown in Figure 1.

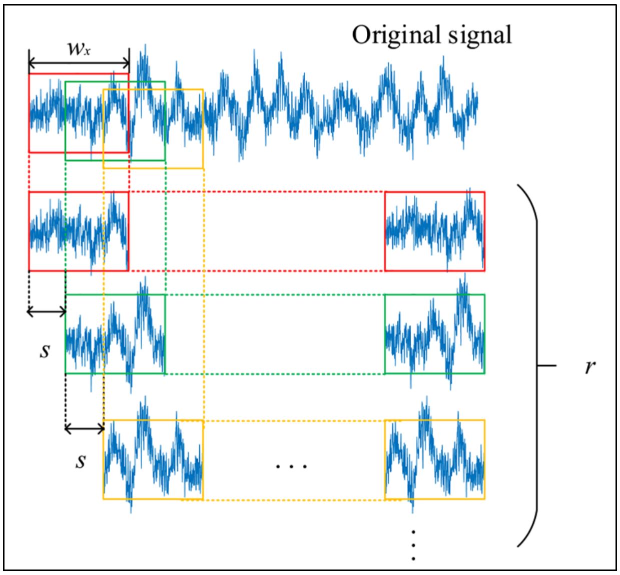


Figure 1. Processing of time series by shift layer.

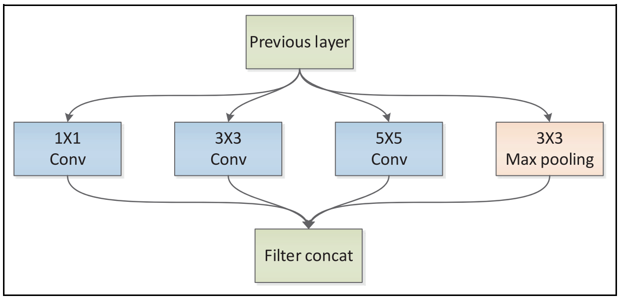


Figure 2. Classical structure of inception block.

Inception module

The conception of local topology CNN can be defined as a convolutional neural network with graph structure between partially hidden layers (convolutional layers and pooling layers). The inception module is exactly a graph structure of multiple convolutional parallel calculations. The typical structure is shown in Figure 2. In the inception module, convolution kernels of different sizes are used to obtain different receptive fields and are then integrated to enrich the information of each layer. There are various scales of features in the time series. Features with more global information distribution prefer larger convolution kernels, and features with more local information distribution prefer smaller convolution kernels. The inception module performs multiple scale convolution operations or pooling operations on the input signal in parallel and stitches all output results into a deep feature map, which helps to achieve better fault characterization.

The introduction of large-scale convolution kernels, for example 5 × 5 and 7 × 7, will generate a large number of calculations. A 5 × 5 convolution filter is 2.78 times more calculation load than a 3 × 3 convolution filter. This paper designs the inception modules A and B shown in Figure 3. Using two cascaded 3 × 3 filters instead of a 5 × 5 filter saves 28% parameter space and reduces computational overhead. Under the condition of the same receptive field, the number of parameters is reduced, and the depth of the neural network is increased. The structure of module A is relatively simple, suitable for extracting shallow features in the early stage. Module B adjusts the order of the pooling layer and the convolutional layer to help diversify learning modes, and adds a one-dimensional convolution process. In the later period, the number of feature maps increases, and module B can extract deeper features based on a large number of shallow features.

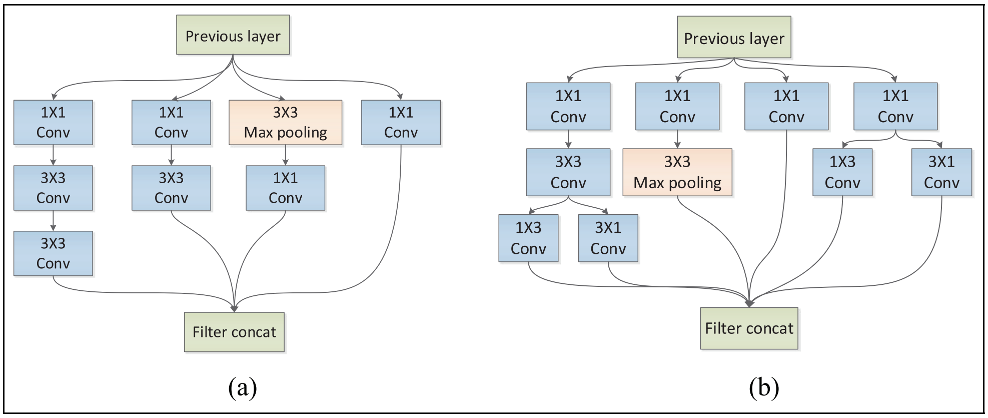


Figure 3. The inception structure proposed in this paper.
(a) Inception module A.
(b) Inception module B.

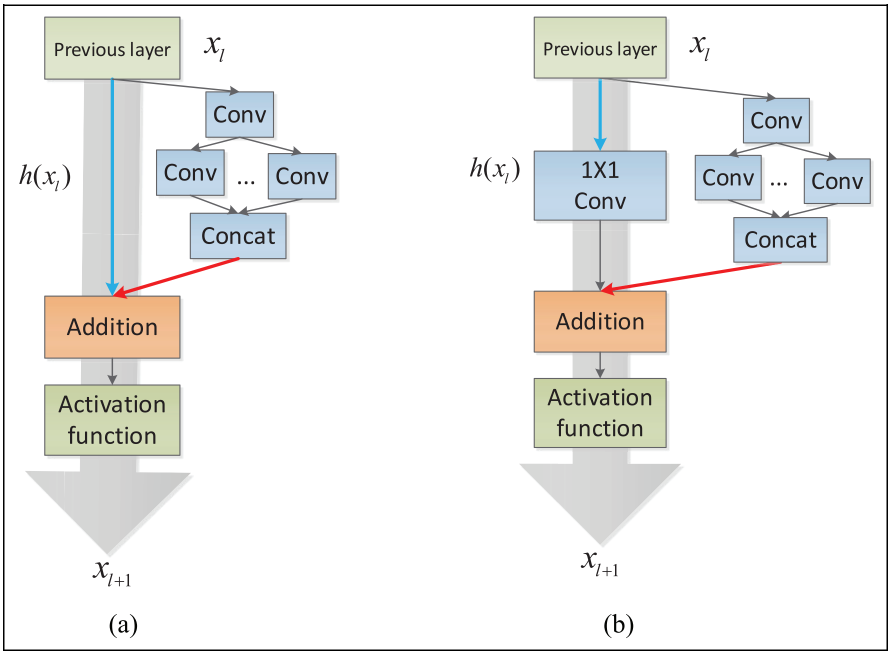


Figure 4. Structure of wide residual.
(a) The feature maps of the two channels (the blue arrow and the red arrow) are consistent.
(b) The feature maps of the two channels (the blue arrow and the red arrow) are inconsistent.

The 1x1 convolution kernel can fuse and reduce the dimension of the feature map, further reduce the training parameters and calculation amount, and at the same time introduce more nonlinearity and improve the generalization ability.

Wide residual structure

The depth of the residual network can prevent network degradation to some extent, but the CNN with deep residual has the problem of low feature utilization, which makes the network training very slow. In this paper, a fault diagnosis CNN with a wide residual structure is proposed. As shown in Figure 4, a shortcut connection is added between two layers of the network. The residual structure is also one of the embodiments of local topology.

Each arrow between the boxes in Figure 4 represents a channel. Feature maps can be transferred directionally on these channels. By widening the convolution channel, the convolution operation increasing horizontally, the width of the residual block is increased, and the depth of the residual block is reduced to speed up the training.

Shortcuts can be viewed as simple equivalent maps. When the network is degraded, the features of the lower layer are directly passed to the upper layer to prevent the network from degrading. Shortcuts do not generate additional parameters, nor increase the computational complexity. This can be described by the following formula:

$$x_{l+1} = \sigma(h(x_l) + F(x_l, W_l)) \tag{2}$$

where x_l and x_{l+1} are the input and output of the lth layer. σ is the activation function. $F(x_l, W_l)$ is the wide residual part, generally composed of multiple convolution operations. $h(x_l)$ is a direct mapping, such as Figure 4(a). $h(x_l) = x_l$ when the number of feature maps of the left and right channels are the same, and when the number of feature maps of the two channels are inconsistent, use 1x1 convolution to increase or decrease the dimension to be consistent, as shown in Figure 4(b).

The overall structure of the network

The overall structure of the new CNN network in this paper is shown in Figure 5, including the shift layer, convolution layer, pooling layer, inception modules, wide residual module, dropout layer, flatten layer, fully connected layer and softmax layer. First, the input layer accepts input samples. Then the shift layer reconstructs the samples according to the method described. Next, the features are extracted by parallel convolution layer and pooling layer, in which the inception modules are used to strengthen the features and the wide residual module is used to prevent the network from degenerating in the training process. Filter concat indicates that the feature map is connected in the depth direction. Following the pooling layers and the convolutional layers is the dropout layer. The dropout layer randomly disconnects neurons and prevents overfitting. Then, the flatten layer converts multidimensional feature maps into one dimension, preparing for the classification. Finally, the fully connected layer integrates features and the softmax layer outputs the probability of various labels.

The key parameter settings of each layer are shown in Table 1. The layer marked with 'V' indicates that the 'valid' filling principle is used, and the extra rows or columns are discarded, and the size of the output feature map will be reduced accordingly. The unmarked 'V' indicates that the 'same' filling principle is used, and the output is equal to the input size when the moving step of kernel is 1. The inside of '(·)' indicates the size, number, moving step size, and filling principle of the convolution kernel. Inside the '[·]' is the pooling core size, the moving step size, and the filling principle. The '–' connection represents the same level.

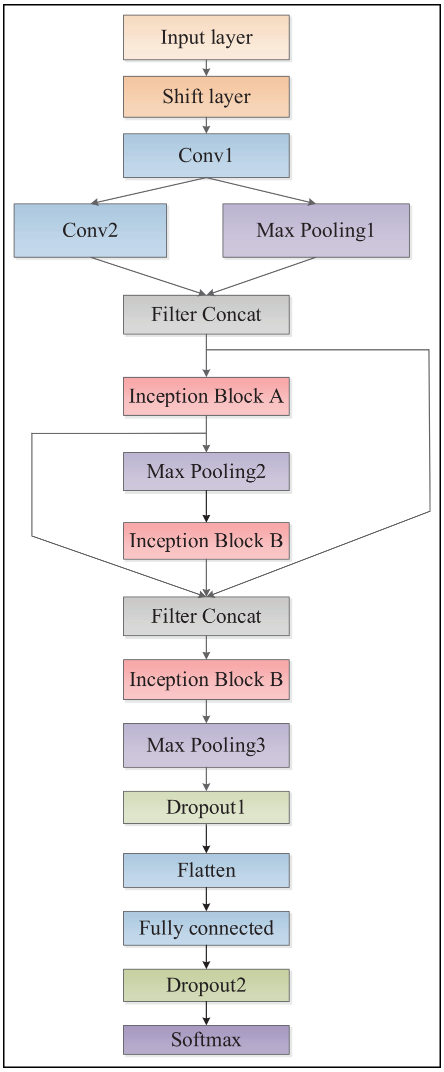


Figure 5. The overall structure of CNN proposed in this paper.

Model parameter optimization

Loss function

The fault diagnosis studied in this paper is a multi-classification problem. The labels output by the neural network are defined as one-hot vectors, which eliminates the quantitative association of scalar labels. The training set like $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$, $(x^{(i)}, y^{(i)})$ is the value and label of the ith sample, where N represents the number of samples in the training set. The vector form of the label is

$$y^{(i)} = e_j = [0, \ldots, 1 \ldots, 0]^T, y^{(i)} \in R^{k \times 1} \tag{3}$$

where e_j is a natural vector, that is, the sample is class j, then the jth term of e_j is 1, and the remaining terms are 0. k is the number of label types. The neural network finally outputs the probability that the sample $x^{(i)}$ belongs to various types through the softmax layer. Cross entropy is used as a loss function for a single sample, and the calculation is as follows:

$$Loss_i(W, b; x^{(i)}, y^{(i)}) = -\sum_{i=1}^{k} (y^{(i)})^T \ln \begin{bmatrix} p(y^{(i)} = e_1 | x^{(i)}; W, b) \\ p(y^{(i)} = e_2 | x^{(i)}; W, b) \\ \vdots \\ p(y^{(i)} = e_k | x^{(i)}; W, b) \end{bmatrix} \tag{4}$$

Table 1. Hidden layer parameter settings.

Hidden layers	Parameter	Activation function
Conv1	(3×3, 16, 2)	ReLu
Conv2	(3×3, 32, 2, V)	ReLu
Max pooling1	[3×3, 2, V]	None
Max pooling2		
Inception block A	(1×1, 16, 1) – (1×1, 16, 1) – [3×3, 1] – (1×1, 32, 1) (3×3, 16, 1) – (3×3, 16, 1) – (1×1, 16, 1) (3×3, 32, 1)	ReLu
Inception block B	(1×1, 16, 1) – (1×1, 16, 1) – (1×1, 32, 1) – (1×1, 32, 1) (3×3, 16, 1) – [3×3, 1] – (1×3, 32, 1) – (3×1, 32, 1) (1×3, 64, 1) – (3×1, 64, 1)	ReLu
Max pooling3	[3×3, 3, V]	None
Dropout1	0.4	None
Dropout2	0.5	None
Fully connected	8	Sigmoid
Filter concat	None	None
Softmax	None	Softmax

where W, b are the weight and bias parameters of the neural network.

Parameter update

The batch gradient back propagation algorithm is used to optimize the parameters of the neural network, and the gradient of the output layer is calculated as

$$\delta^{nl} = \frac{\partial J}{\partial z^{nl}} = \frac{\partial J}{\partial a^{nl}} \odot \sigma'(z^{nl}) \tag{5}$$

where a^l is the output of the lth layer, $\sigma(z)$ is the activation function, $\odot$ represents the Hadamard product. From this, the gradient of the fully connected layer can be calculated as

$$\delta^l = -((W^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{6}$$

Let the output of the pooling layer be a^{l+1}, and the output of the previous layer of the pooling layer be a^l. Since the pooling layer has no activation function, the forward propagation of the pooling layer is

$$z^{l+1} = pooling(a^l) \tag{7}$$

Then the gradient of the previous layer is

$$\delta^l = \frac{\partial J}{\partial z^l} = \frac{\partial J}{\partial z^{l+1}} \frac{\partial z^{l+1}}{\partial a^l} \frac{\partial a^l}{\partial z^l} = upsample(\delta^{l+1}) \odot \sigma'(z^l) \tag{8}$$

where *upsample* refers to the reduction of the size of the matrix δ^{l+1} to the size before pooling.

Let the output of the convolutional layer be a^{l+1} and the forward propagation of the convolutional layer be

$$a^{l+1} = \sigma(z^{l+1}) = \sigma(a^l * W^{l+1} + b^{l+1}) \tag{9}$$

where $*$ represents the convolution operation, and the gradient of the previous hidden layer is

$$\begin{aligned}\delta^l &= \frac{\partial J}{\partial z^l} = \frac{\partial J}{\partial z^{l+1}} \frac{\partial z^{l+1}}{\partial a^l} \frac{\partial a^l}{\partial z^l} \\ &= \delta^{l+1} * Rotation180(W^{l+1}) \odot \sigma'(z^l)\end{aligned} \tag{10}$$

where $Rotation180$ indicates that the convolution kernel is rotated 180 degrees, that is, flipped up and down once, and then flipped left and right again.

The mini-batch gradient descent method uses one batch at a time to greatly reduce the number of iterations required for convergence, and at the same time can make the convergence result closer to the global optimal. The parameters are updated according to the mini-batch gradient descent method, and the iteration of the fully connected layer parameters is

$$W^l_{(i+1)} = W^l_{(i)} - \alpha \sum_{j=1}^{m} \delta^l_j (a^l_j)^T \tag{11}$$

$$b^l_{(i+1)} = b^l_{(i)} - \alpha \sum_{j=1}^{m} \delta^l_j \tag{12}$$

where $W^l_{(i)}$ represents the value of W^l at the ith update, δ^l_j represents the δ^l of the jth sample, and m is the number of samples. The iteration of the convolutional layer parameters is

$$W^l_{(i+1)} = W^l_{(i)} - \alpha \sum_{j=1}^{m} \delta^l_j * a^{l-1}_j \tag{13}$$

$$b^l_{(i+1)} = b^l_{(i)} - \alpha \sum_{j=1}^{m} \sum_{u,v} (\delta^l_j)_{u,v} \tag{14}$$

where $\sum_{u,v} (\delta^l_j)_{u,v}$ represents that all channels of δ^l_j are accumulated, and finally becomes one channel.

Data set production

The simulation in this paper is based on a six-degree-of-freedom nonlinear dynamic model of Boeing 747-200 aircraft.

Table 2. Actuator system parameter setting.

	Elevator	Rudder	Ailerons
Natural frequency /Hz	40	40	35
Damping ratio	0.8	0.8	0.85
Maximum amplitude /°	25	30	30
Maximum speed $/^{\circ} \cdot s^{-1}$	40	40	40

The relevant parameters of the aircraft are from the literature (Roskam, 2001). Flight control uses conventional PID control to keep the aircraft at a stable attitude, speed and altitude within a certain range. At the same time, the actuator servo model, sensor model, wind interference model and various fault models are established to obtain data samples.

In practice, the mechanism model is usually established in the form of a state-space model or transfer function (Liu et al., 2020). The output capacity of large servo actuator is larger than the actual load, and it pays special attention to the dynamic process in a small range. In this case, the transfer function can be used to establish a credible model. The actuator servo model in this paper is described by the following transfer function:

$$G(s) = \frac{k\omega_n^2}{s^2 + 2\xi\omega_n + \omega_n^2} \tag{15}$$

where ξ is the damping ratio of the system and ω_n is the natural frequency of the system. The amplitude and speed of the actual actuator system are limited, and the amplitude limit is given in this paper. The settings of the actuator system parameters are shown in Table 2.

The noise and delay of the sensor cannot be ignored in the actual signal acquisition process. In this paper, a white noise signal with a power of 0.0001 and a delay of 0.03 seconds are added to the position sensor of the actuator. The proposed method is not affected by the sensor delay. First, time delay can be reflected in the phase difference between the control command and the position sensor signal. Then, the shift layer reconstructs a one-dimensional signal into a two-dimensional signal, which translates the signal in different scales. Finally, through convolution and pooling operations of different sizes and strides, the phase difference caused by the time delay can be eliminated. There are various disturbances in the atmosphere, and aircraft flying in the air will be affected by atmospheric turbulence. In order to balance this unpredictable wind disturbance, the flight control system will slightly deflect the rudder surface of the aircraft. In order to further improve the authenticity of the simulation data, the Dryden atmospheric turbulence model was established to simulate the real atmospheric disturbance environment. When the disturbance is applied to the aircraft, the control surface of the aircraft will fluctuate in a small range, which is difficult to predict, resulting in noise and outliers in the data set. Training the neural network using data with noise and outliers can improve the robustness of the fault diagnosis model (You et al., 2019).

Fault model

Consider the eight common fault modes of actuators and sensors. The mathematical model of the fault and the marking of each state are shown in Table 3. Considering the high reliability of aviation components, it is assumed that actuator and sensor faults will not occur simultaneously. The data is generated under the conditions of a flight altitude of 6km and a flight speed of 250m/s. Assuming that the simulation duration is 10 seconds, actuator fault and sensor fault occur from the 5th second. The elevator signals for each state corresponding to Table 3 are shown in Figure 6.

We applied different strengths for actuator damage (C3), sensor gain loss (C7) and sensor fixed deviation (C6). The values of damage coefficient and gain loss coefficient are 20%, 40%, 60% and 80%, which are uniformly distributed in the data set. The value of the fixed deviation of the sensor was set in the range of ±0.1 rad. As for actuator floating (C2), the disturbance $\Delta(t)$ is composed of two sine curves with different amplitudes as follows:

$$\Delta(t) = [k\sin(2\pi t) + 0.1k\sin(20\pi t)]/57.3\ (rad) \tag{16}$$

where k takes a random number between 0 and 2 in each simulation.

Data set partition

Every sample includes two time series: actuator position signal and corresponding control command. The sampling

Table 3. Description of each state.

States	Mark	Mathematical model	Explanation	Label
Actuator stuck	C1	$u_i(t) = \begin{cases} u_i(t), t<t_f \\ u_i(t_f), t \geqslant t_f \end{cases}$	$u_i(t)$ is the position of the ith actuator at the time t. t_f is the moment when the stuck fault occurs.	00000001
Actuator floating	C2	$u_i(t) = v_i(t) + \Delta(t)$	$\Delta(t)$ is an irregular disturbance signal.	00000010
Actuator damage	C3	$u_i(t) = \lambda_i v_i(t), \lambda_i \in (0, 1)$	$\lambda_i \in (0, 1)$ is the failure factor of the ith actuator.	00000100
Actuator reverse	C4	$u_i(t) = -v_i(t)$		00001000
Sensor complete failure	C5	$y_i(t) = \begin{cases} y_i(t), t<t_f \\ y_i(t_f), t \geqslant t_f \end{cases}$	$y_i(t)$ is the measurement of the ith sensor at time t.	00010000
Sensor fixed deviation	C6	$\begin{cases} y_i(t) = y_i(t) + \delta_i(t), t \geqslant t_f \\ \dot{\delta}_i(t) \equiv 0 \end{cases}$	None	00100000
Sensor gain loss	C7	$y_i(t) = \lambda_i y_i(t), \lambda \in (0, 1)$	λ_i is the gain loss coefficient of the ith sensor.	01000000
No fault	C8	None	None	10000000

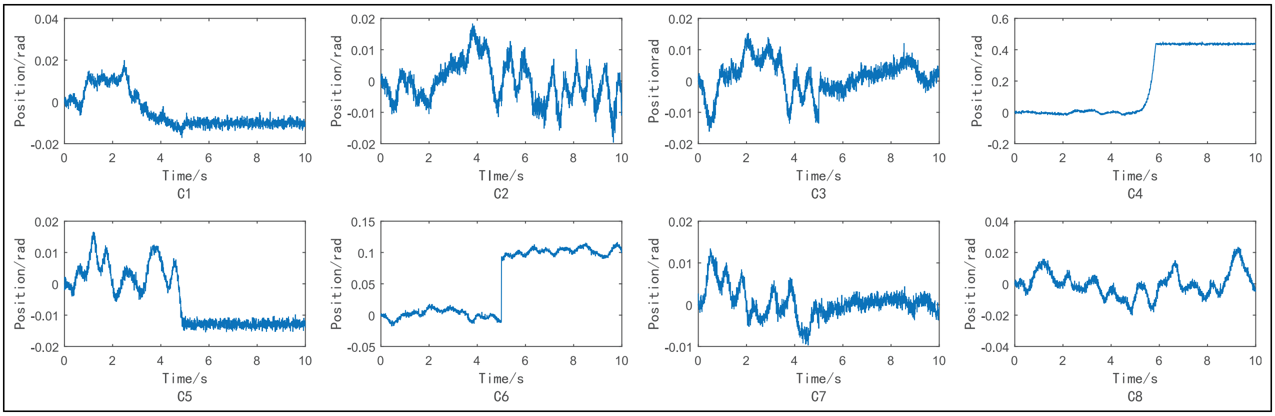


Figure 6. Actuator signal of each state. The fault occurred in the fifth second.

frequency is 200Hz and the sampling time is 5s. State C1 to C8 produce 1,450 samples respectively, for a total of 11,600 samples. For training set selection, we randomly select 65% of the data set samples in each class. For verification set selection, we randomly select 20% of the data set samples in each class, and these samples are not repeated with the previous training set. The remaining 15% of the samples are used as a test set.

Data standardization

The main purpose of standardizing the original data is to reduce the impact of unexpected noise and preserve the signal distribution, eliminate the influence of the difference between the dimensions and the range of the indicators, and improve the comparability of data. The calculation is as follows:

$$x^* = [x - mean(x)]/std(x) \tag{17}$$

where x^* represents the standardized signal, x is the original signal, $mean(x)$ is the average of x, $std(x)$ is the standard deviation of x.

Results and discussion

Performance on train and validation set

Set model parameters epoch is 80, the batch size is 150, the learning rate is 0.0001. The model is verified every 10 weight updates during the training process, calculating the accuracy and loss of the model in the verification set. In the training process, every time an epoch is trained, the accuracy of the training set is calculated, and the learning effect of the neural network can be seen from it.

The output of the neural network is a probability vector, each element is a continuous variable between 0–1, and the one-hot label is an integer vector with an element value of 0 or 1. The continuous vector output from the neural network is converted into an integer vector by the function *fix* for logical judgment. The accuracy rate is calculated as follows:

$$acc = \frac{sum(fix(\hat{y}) = = y)}{M} \tag{18}$$

The function *fix* sets the largest element in the vector $\hat{y}$ to 1 and the remaining elements to 0. The state with the highest probability value is selected as the state diagnosed by the algorithm. M is the total number of samples.

The loss value curves of test set and verification set are shown in Figure 7(a). With the iteration of neural network training, the loss value of the verification set gradually decreases. The loss value of the training set fluctuates slightly but the overall trend is decreasing. Both the loss values of the training set and the verification set finally converge to around 0.001. The loss value of the test set is generally higher than that of the verification set, indicating that there is no overfitting. The overall accuracy rate curve is shown in Figure 7(b). With the increase of the training times, the accuracy rate gradually increases, and the accuracy of the verification set finally stabilizes at about 95%. The accuracy of the test set fluctuates significantly, and most of the time during the iteration process the accuracy of the test set is lower than that of the verification set. This is because in order to avoid overfitting during training, the dropout layer will randomly disconnect some neurons. While the dropout operation is turned off when predicting the validation set samples, all neurons have weights between each other, so the model is complete and the accuracy rate is likely to be higher than the training set during the same period.

From the comparison of the two figures, it can be seen that the accuracy rate tends to converge after about the 2000th generation, but the loss value has been declining throughout the training process. This shows that the model is strengthening its own stability and improving the credibility of the diagnosis results.

The prediction results of each iteration are counted. The changes in the accuracy of actuator fault and sensor fault during training are shown in Figure 7(c) and Figure 7(d) respectively. Whether it is a training set or a verification set, the model's diagnostic performance for actuator and sensor faults is improved at the same time. The accuracy rate of these two types of fault has been raised to a higher level with the training, which is equivalent to the overall accuracy rate. The

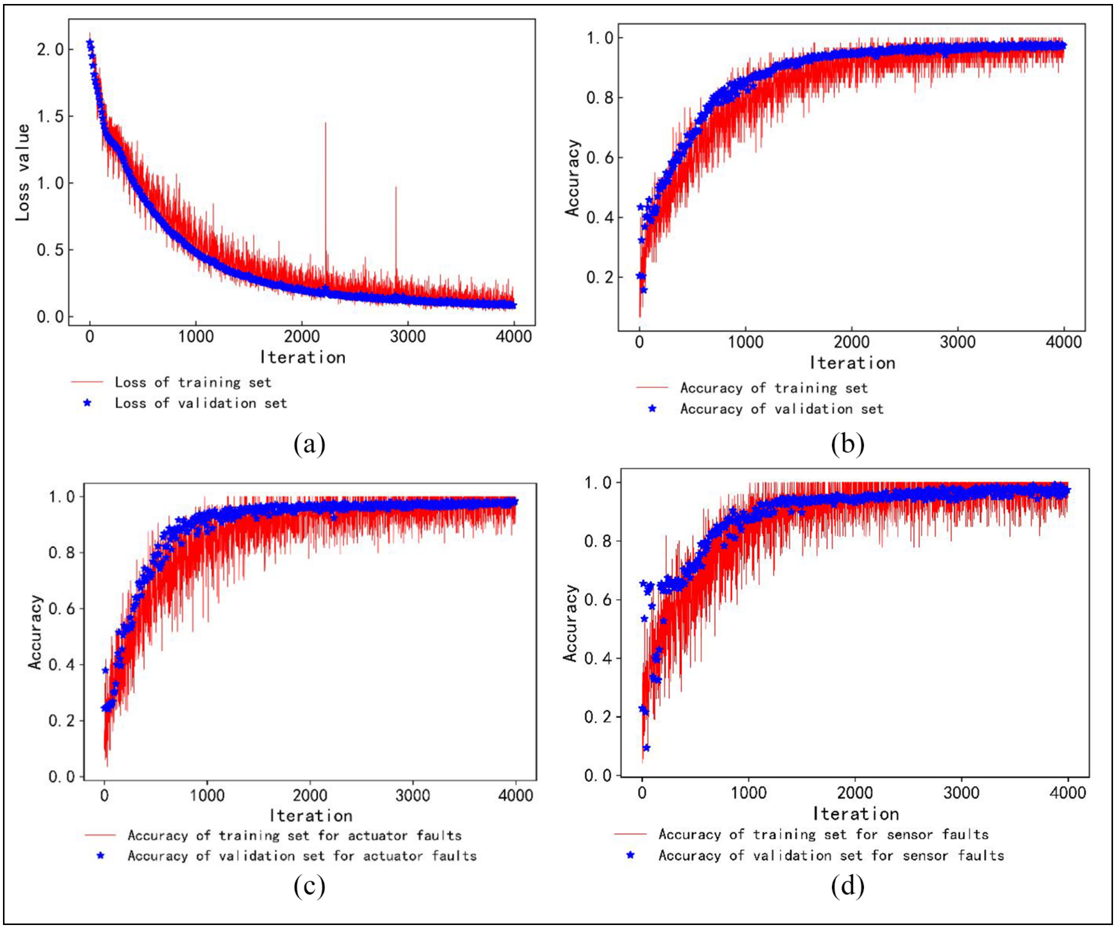


Figure 7. Loss and accuracy of iterative process.
(a) Loss of training set and validation set.
(b) Accuracy of training set and validation set.
(c) Accuracy of training set and validation set for actuator fault.
(d) Accuracy of training set and validation set for sensor fault.

model's ability to identify these two types of fault is balanced and accurate. First, the principal component analysis method is used to reduce the data to 50 dimensions and to suppress signal noise. Next, 't-SNE' is adopted to reduce the data to two dimensions to complete the data visualization. Each sample is drawn in the form of a number in the figure, and the number represents the true label of the sample.

To explore the influence of test sample size on accuracy, we conducted experiments with test sets of different sample proportions. The training set accounts for 65%, 60%, 55%, 50%, 45%, 40%, 35%, 30% and 20% of the data set respectively. Because the model hyperparameters are not adjusted, the remaining samples are all used as test sets. We performed 10 times for each training set proportion; the mean accuracy and standard deviation are shown in Figure 8.

What stands out in this figure is the rapid decrease in accuracy after the training set proportion less than 50%. In general, the accuracy decreases with the decrease of training set proportion, whereas the variance increases with it. The larger the variance, the worse the stability of the model. Therefore, 50% to 60% of the training samples (725 to 942 samples for each class) can ensure the accuracy and stability of the model.

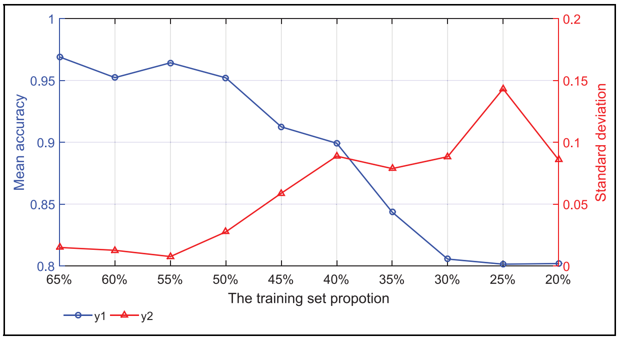
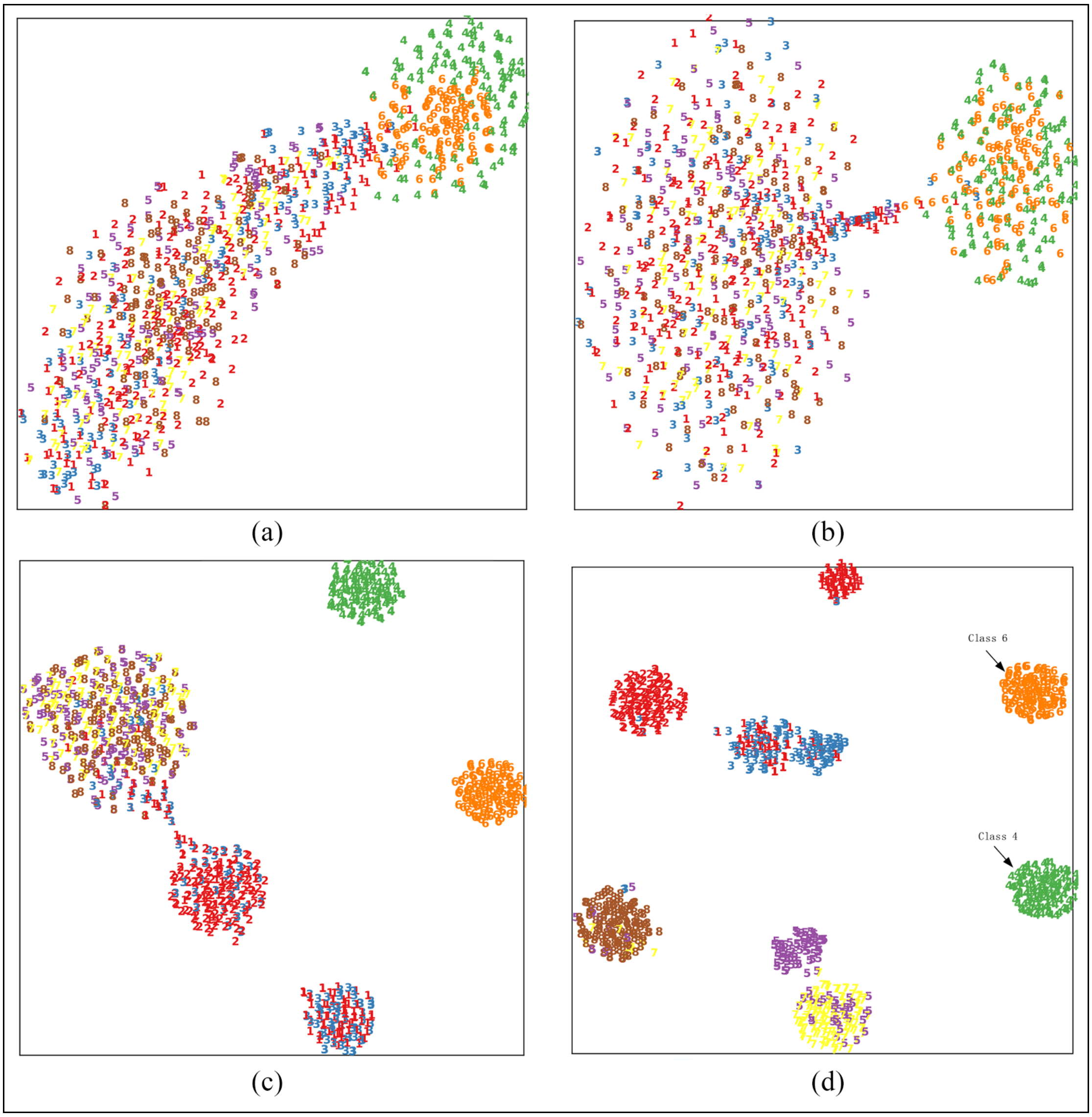


Figure 8. The accuracy and standard deviation of different training set proportion.



(a)

(b)

(c)

(d)

Figure 9. Visualization of the feature space in the last convolution layer.
(a) Iteration 1.
(b) Iteration 50.
(c) Iteration 1000.
(d) Iteration 4000.

Process visualization

The t-SNE technology visualizes the data by nonlinearly reducing the high-dimensional data of the hidden layer to two dimensions (Laurens and Hinton, 2008). Data visualization can analyse the changes of the feature distribution inside the neural network during the training process. The further back the hidden layer can reflect the global features, the output of the last convolutional layer is selected to analyse its feature space.

Figure 9 is the visualization results of the convolutional layer at the 1st, 50th, 1000th and 4000th iterations of the network. It can be seen that the distribution of the feature space is chaotic at the beginning of the training. With the iteration, the samples are gradually separated and clustered, and finally clustered into eight clusters, which is consistent with the actual number of labels. The interval between clusters is obvious, and the clusters labelled 4 and 6 neither contain samples of other classes nor are mixed into other clusters. It shows that the separation effect of C4 and C6 faults is the best.

The influence of shift layer parameters

The shift layer affects the adaptability of the CNN network to the processing time series, which in turn affects the diagnostic

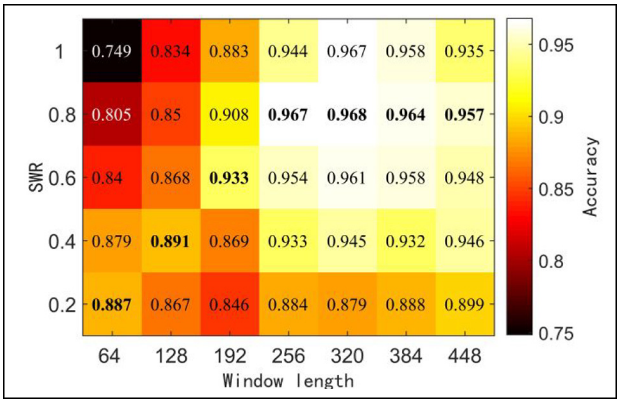


Figure 10. The influence of shift layer parameters on accuracy.

accuracy. The window length and step window ratio (SWR) are selected to analyse the effect of the shift layer on the accuracy, where the SWR is defined as

$$SWR = s/w_x \tag{19}$$

The step size is calculated by multiplying the window length by the SWR, and the decimal part is rounded off. Figure 10 shows the average accuracy of 20 runs for each pair of parameter values. The step size is calculated by multiplying the window length by the SWR and then rounding down the decimal part.

It can be seen that when the window length is fixed, as the SWR increases, the accuracy rate basically increases first and then decreases. When the SWR is constant, as the window length increases, the change trend of accuracy also increases first and then decreases. SWR too large will miss the time correlation of the sequence, too small will pay too much attention to the local features and ignore the global features. If the window length is too small, long-term changes of the time series cannot be learned, and if it is too large, the sample size is increased and the calculation load is increased. The shift layer achieves better results when the window length is between 256 and 448, the SWR is between 0.2 and 0.8, and its accuracy rate is above 90%. The performance of CNN processing time series can be improved by selecting the appropriate parameters of the dislocation layer.

Testing of different fault channels

The trained model is applied to three test sets with different state conditions to verify the performance and generalization ability of the algorithm proposed in this paper. The state conditions of the three test sets are shown in Table 4. The fault channel represents the control channel of the fault location. For example, if the elevator or elevator position sensor fails, the fault channel is called the elevator channel. Parameters of the shift layer are the window length 320 and the step length 192.

Table 4. Test set settings for different fault channels.

Test set	Number of samples	Fault channel	Flying height	Flight speed
1	1740	Elevator	6km	250m/s
2	1740	Rudder	8km	200m/s
3	1740	Ailerons	9km	220m/s

Table 5. The confusion matrix representation of the classification results.

		True label	
		Positive	Negative
Predicate label	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

The flight height and speed of test set 1 are the same as the training set. The confusion matrix is adopted to visually present the diagnosis results and evaluate the performance of the model. Table 5 is the confusion matrix representation of the classification results. With real tags and diagnostic categories, we can calculate the confusion matrix shown in Table 5. When we calculate the index of class C_i, C_i is regarded as positive and the rest class $C_j(j \neq i)$ are regarded as negative.

Here are definitions of precision, recall, missing alarm rate (MAR) and fault alarm rate (FAR).

$$precision = \frac{TP}{TP + FP} \tag{20}$$

$$recall = \frac{TP}{TP + FN} \tag{21}$$

$$MAR = 1 - recall = \frac{FN}{TP + FN} \tag{22}$$

$$FAR = \frac{FP}{TN + FP} \tag{23}$$

Evaluate the model from different aspects through precision, recall and fault alarm rate. The precision describes the generalization ability of the model. The recall, also called sensitivity and fault detection rate (FDR), is the ability to detect faults. The confusion matrices of test set 1–test set 3 are shown in Tables 6–8. It can be seen that the model also performs well on test set 2 and test set 3 that are different from the training set state. In terms of fault source diagnosis, the model can correctly distinguish between actuator faults and position sensor faults. In terms of fault type diagnosis, each type of identification is accurate, and the diagnosis effect of actuator jamming, reverse fault and sensor fixed deviation fault is the best. The fault alarm rate and fault detection rate are shown in Figure 11 and Figure 12. The false alarm rate of each type of fault is between 12% and 16%, and is not related to the type of failure. The fault detection rate is generally above 90%, which is related to the fault type. The algorithm

Table 6. Confusion matrix of test set 1.

Test set 1	True label									
Predicate label		C1	C2	C3	C4	C5	C6	C7	C8	Precision (%)
	C1	194	0	0	0	9	0	0	0	95.57
	C2	0	233	0	0	0	0	0	7	97.08
	C3	0	0	199	0	0	0	4	0	98.03
	C4	0	0	0	213	0	0	0	0	100
	C5	0	0	0	0	200	0	15	0	93.02
	C6	0	0	0	0	0	231	0	0	100
	C7	0	0	3	0	0	0	220	0	98.65
	C8	0	18	0	0	1	0	0	193	91.04
	Recall (%)	100	92.83	98.51	100	95.24	100	92.05	96.50	

Table 7. Confusion matrix of test set 2.

Test set 2	True label									
Predicate label		C1	C2	C3	C4	C5	C6	C7	C8	Precision (%)
	C1	201	0	0	0	9	0	0	0	95.71
	C2	0	197	0	0	0	0	0	7	96.57
	C3	0	0	213	0	0	0	3	0	98.61
	C4	0	0	0	225	0	0	0	0	100
	C5	0	0	0	0	205	0	18	0	91.93
	C6	0	0	0	0	0	219	0	0	100
	C7	0	0	4	0	0	0	214	0	98.17
	C8	0	13	0	0	4	0	0	208	92.44
	Recall (%)	100	93.81	98.16	100	94.04	100	91.06	96.74	

Table 8. Confusion matrix of test set 3.

Test set 3	True label									
Predicate label		C1	C2	C3	C4	C5	C6	C7	C8	Precision (%)
	C1	217	0	0	0	4	0	0	0	98.19
	C2	0	207	0	0	0	0	0	6	97.18
	C3	0	0	215	0	0	0	4	0	98.17
	C4	0	0	0	219	0	0	0	0	100
	C5	0	0	0	0	194	0	18	0	91.51
	C6	0	0	0	0	0	224	0	0	100
	C7	0	0	1	0	0	0	212	0	99.53
	C8	0	14	0	0	2	0	0	203	92.69
	Recall (%)	100	93.67	99.54	100	97	100	90.60	97.13	

has the highest recognition rate for actuator stuck, damaged, reverse fault, and position sensor fixed deviation fault.

Comparing with other methods

First of all, a longitudinal comparison is conducted to verify the contribution of the shift layer, residual structure and inception module to the performance improvement of the model in this paper. Then it is compared with other mainstream data-driven fault diagnosis methods to verify the advantages of this algorithm. These methods include traditional machine learning methods such as SVM (Qin et al., 2016), KNN (Cerrada et al., 2018), deep learning methods such as BP (Liang et al., 2018) neural network, CNN (Jing et al., 2017), RNN (Dong, 2018), and several famous neural networks such as AlexNe (Krizhevsky et al., 2012), VGG19 (Liu and Deng, 2015), ResNet (He et al., 2016). The result of each algorithm is the average of 1,500 random samples on test sets 1–3 for 20 runs. The specific diagnosis results are shown in Table 9 and Table 10.

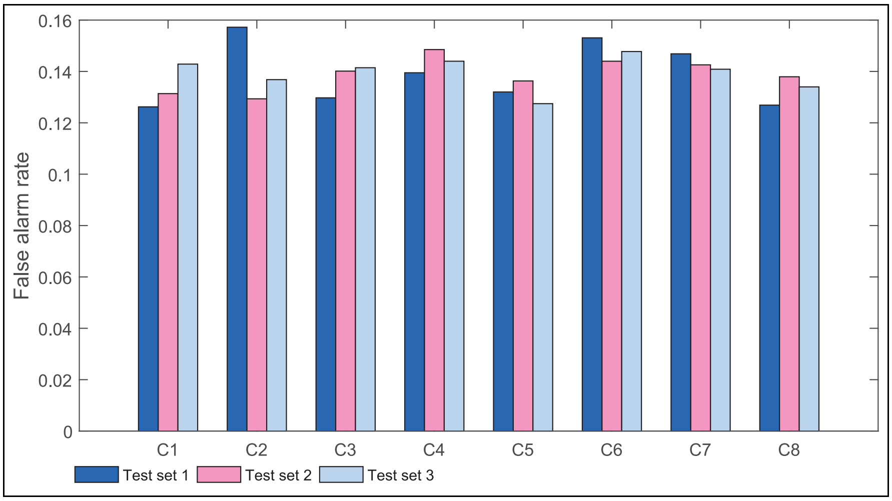


Figure 11. The false alarm rate of three test sets.

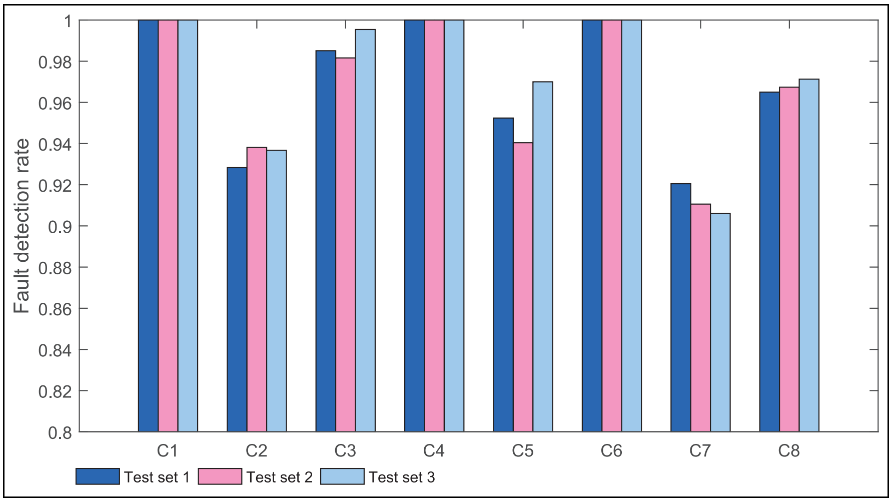


Figure 12. The fault detection alarm rate of three test sets.

Table 9. Precision (%) for different methods.

No.	Method	C1	C2	C3	C4	C5	C6	C7	C8	Overall
1	SVM (Qin et al., 2016)	47.46	65.76	64.69	72.74	38.69	91.33	88.10	30.24	62.38
2	KNN (Cerrada et al., 2018)	55.13	69.36	36.87	74.94	48.13	83.58	3.32	54.99	53.29
3	BP (Liang et al., 2018)	68.91	69.23	22.09	73.75	56.61	**100**	40.48	33.82	58.11
4	CNN (Jing et al., 2017)	74.32	62.81	77.01	100	30.60	94.71	73.93	27.74	67.64
5	RNN (Dong, 2018)	62.73	89.06	42.56	**100**	25.7	**100**	92.43	90.70	72.51
6	Shift layer BP	62.84	72.87	40.61	86.87	50.44	**100**	36.72	52.29	62.83
7	Shift layer CNN	90.07	68.30	65.99	94.37	71.22	**100**	93.69	48.18	78.98
8	Shift layer RNN	61.84	**99.35**	3.76	**100**	2.92	**100**	**100**	78.28	67.94
9	Inception CNN	85.51	6783	94.78	**100**	89.22	**100**	93.32	66.80	87.18
10	Wide residual CNN	75.35	81.18	74.17	99.16	77.06	**100**	92.33	58.85	82.26
11	AlexNet (Krizhevsky et al., 2012)	96.40	95.51	90.36	**100**	40.03	**100**	95.47	90.66	88.55
12	VGG (Liu and Deng, 2015)	**97.28**	89.02	83.82	**100**	91.73	**100**	9357	68.64	90.51
13	ResNet (He et al., 2016)	91.38	81.19	75.80	**100**	61.25	**100**	94.11	73.44	84.65
14	The proposed method	96.49	96.95	**98.27**	**100**	**92.14**	**100**	98.79	**92.04**	**96.84**

Table 10. Recall (%) for different methods.

No.	Method	C1	C2	C3	C4	C5	C6	C7	C8	Overall
1	SVM (Qin et al., 2016)	86.32	54.68	60.74	55.36	44.32	83.45	53.29	45.54	60.46
2	KNN (Cerrada et al., 2018)	50.33	65.88	49.97	63.99	34.56	78.65	16.15	40.46	50.04
3	BP (Liang et al., 2018)	43.02	70.99	51.32	82.42	35.78	**100**	39.94	61.31	60.60
4	CNN (Jing et al., 2017)	61.46	59.82	57.15	**100**	37.89	**100**	67.32	55.29	67.37
5	RNN (Dong, 2018)	50.93	83.44	57.00	99.57	13.79	**100**	49.76	85.15	67.46
6	Shift layer BP	48.20	76.53	58.79	83.88	41.71	**100**	40.28	62.78	64.02
7	Shift layer CNN	91.55	69.89	51.62	**100**	40.41	**100**	55.06	38.01	68.32
8	Shift layer RNN	56.51	77.22	27.59	**100**	3.45	99.14	44.95	58.67	58.44
9	Inception CNN	95.91	74.65	92.70	99.22	83.38	**100**	89.81	65.56	87.65
10	Wide residual CNN	93.87	80.02	68.50	**100**	72.76	99.69	72.90	74.46	82.77
11	AlexNet (Krizhevsky et al., 2012)	100	91.63	96.87	**100**	70.28	**100**	64.36	96.38	89.94
12	VGG (Liu and Deng, 2015)	96.74	74.06	90.18	**100**	**97.51**	**100**	81.42	82.43	90.29
13	ResNet (He et al., 2016)	95.46	80.37	80.76	**100**	81.12	**100**	75.37	65.23	84.79
14	The proposed method	**100**	**93.43**	**98.75**	**100**	95.41	**100**	**91.25**	**96.74**	**96.95**

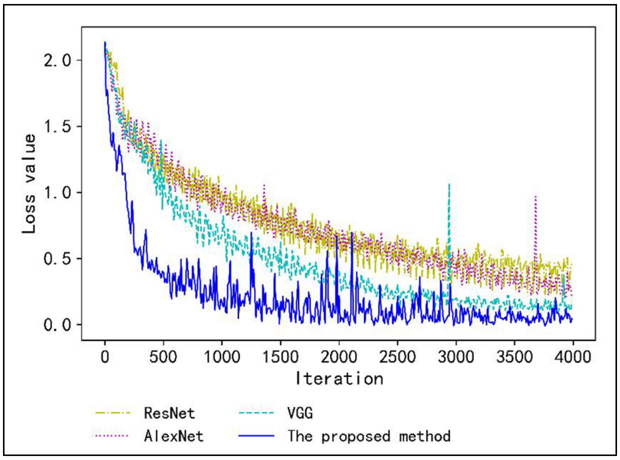


Figure 13. The convergence trend of loss values of different algorithms.

Table 9 shows that after the traditional CNN applied the proposed shift layer, the precision increased from 67.64% to 78.98%. This indicates that the shift processing in this paper is beneficial for CNN to analyse time series. The precision of BP has improved after applying the misalignment layer, but it has declined for RNN. This is because the RNN itself has a memory function for time series, and misaligned input increases the complexity of processing. The precision of BP has improved after applying the shift layer, but it has declined for RNN. This is because the RNN itself has a memory function for time series, and the shift layer increases the complexity of processing. On the basis of the CNN with shift layer, the inception module (Method 9) and the wide residual structure (Method 10) are added respectively. Then the average precision of the two methods is much higher than that of the CNN with shift layer. The precision of the method proposed in this paper combined with the inception module and the wide residual structure has reached 96.8%, which can give full play to the strengths of the two and learn more features.

Compared with traditional machine learning methods, the model based on deep learning is more generalized and more sensitive to faults. Among these deep learning-based methods, ResNet, AlexNet and VGG have also achieved relatively good results. Figure 13 shows their convergence process with the loss value of this algorithm.

It can be observed that ResNet's convergence process is the most stable, but the convergence speed is slow and the convergence error is large. VGG has a faster convergence speed and a smaller convergence error. VGG and AlexNet produce abnormally high loss values in the late training period, and the model stability is poor. Although the algorithm proposed in this paper fluctuates greatly in the accuracy rate in the middle stage, it gradually stabilizes in the later stage of training, with the fastest convergence speed and the smallest convergence error. Combined with the accuracy and sensitivity analysis of Table 9 and Table 10, the method of this paper has achieved relatively optimal fault diagnosis performance.

Conclusion

The problem solved by the proposed method is to distinguish the fault of an actuator and its position sensors accurately and identify the fault type. A new local topological convolutional neural network is proposed to diagnose the faults of actuators and their position sensors effectively. Tests and comparisons in several states show that this method has the following advantages:

(1) The proposed dislocation layer can enhance the correlation of the input signal in time, and improve the performance of CNN and BP neural network in processing time series.

(2) The inception module with local topology breaks the sequential stacking design pattern adopted by traditional CNN. It can learn more complex patterns while reducing neural network parameters. The introduction of the wide residual structure prevents network

degradation and accelerates the convergence of the loss function.

(3) Simulation results show that the method proposed in this paper has better generalization ability and sensitivity to faults. The fault recognition precision and recall reached 96.8% and 96.9% respectively. Compared with other mainstream algorithms, it has improved by 6.3% and 6.7%.

(4) This paper relaxes the assumptions for actuator fault diagnosis, which is not affected by the position sensor fault. The model does not rely on human empirical knowledge to extract fixed features in advance, does not require a strict mathematical model and has good portability, laying a foundation for other types of fault diagnosis.

The limitation of the proposed method is that one model in this paper can only be adapted to one aircraft type. Future work will focus on data imbalance and model migration. Synthetic minority oversampling technique (SMOTE) will be utilized to solve the problem of data imbalance. The transfer learning theory will be combined to establish a domain adaptive neural network to solve the model migration problem, so that the model can adapt to the fault diagnosis of different aircraft types.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the National Natural Science Foundation of China (Grant no. 61673209, 71971115).

ORCID iD

Ruonan Wei https://orcid.org/0000-0002-2562-6021

References

Avram RC, Zhang X and Muse J (2017) Quadrotor actuator fault diagnosis and accommodation using nonlinear adaptive estimators. *IEEE Transactions on Control Systems Technology* 25(6): 2219–2226.

Cerrada M, Sánchez RV and Cabrera DR (2018) A semi-supervised approach based on evolving clusters for discovering unknown abnormal condition patterns in gearboxes. *Journal of Intelligent and Fuzzy Systems* 34(6): 3581–3593.

Dong Y (2018) An application of deep neural networks to the in-flight parameter identification for detection and characterization of aircraft icing. *Aerospace Science and Technology* 77: 34–49.

Girshick R, Donahue J, Darrell T, et al. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, USA, 23–28 June 2014, pp. 580–587. Piscataway, NJ: IEEE.

Guo D, Zhong M, Ji H, et al. (2018) A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors. *Neurocomputing* 319: 155–163.

Hassanabadi AH, Shafiee M and Puig V (2017) Actuator fault diagnosis of singular delayed LPV systems with inexact measured parameters via PI unknown input observer. *IET Control Theory and Applications* 11(12): 1894–1903.

He K, Zhang X, Ren S, et al. (2016) Deep residual learning for image recognition. In: *29th IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, 27–30 June 2016, pp. 770–778. Piscataway, NJ: IEEE.

Ji J, Qu J, Chai Y, et al. (2018) An algorithm for sensor fault diagnosis with EEMD-SVM. *Transactions of the Institute of Measurement and Control* 40(6): 1746–1756.

Jiang Y and Yin S (2019) Recent advances in key-performance-indicator oriented prognosis and diagnosis with a MATLAB toolbox: db-kit. *IEEE Transactions on Industrial Informatics* 15(5): 2849–2858.

Jiang Y, Yin S and Kaynak O (2021) Optimized design of parity relation based residual generator for fault detection: Data-driven approaches. *IEEE Transactions on Industrial Informatics* 17(2): 1449–1458.

Jing L, Zhao M, Li P, et al. (2017) A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement* 111: 1–10.

Krizhevsky A, Sutskever I and Hinton G (2012) Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25(2): 1097–1105.

Laurens VD and Hinton G (2008) Visualizing data using t-SNE. *Journal of Machine Learning Research* 9(11): 2579–2605.

Levent E, Turker I and Serkan K (2019) A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Journal of Signal Processing Systems* 91(2): 179–189.

Li C, Zhang Y and Li P (2017) Extreme learning machine based actuator fault detection of a quadrotor helicopter. *Advances in Mechanical Engineering*. DOI: 10.1177/1687814017705068. Available at: https://journals.sagepub.com/doi/full/10.1177/1687814017705068.

Liang Q, Han H, Cui X, et al. (2018) Comparative study of probabilistic neural network and back propagation network for fault diagnosis of refrigeration systems. *Science and Technology for the Built Environment* 24(4): 448–457.

Liu H, Jing J, Ma J, et al. (2018) Fault diagnosis of electromechanical actuator based on VMD multifractal detrended fluctuation analysis and PNN. *Complexity* 2018(2): 1–11.

Liu J, Yang L, Xu M, et al. (2020) Model-based detection of soft faults using the smoothed residual for a control system. *Measurement Science and Technology*. DOI: 10.1088/1361-6501/abaf2b. Available at: https://iopscience.iop.org/article/10.1088/1361-6501/abaf2b

Liu S and Deng W (2015) Very deep convolutional neural network based image classification using small training sample size. In: *3rd IAPR Asian Conference on Pattern Recognition*, Kuala Lumpur, Malaysia, 3–6 November 2015, pp. 730–734. Piscataway, NJ: IEEE.

Mao W, He L, Yan Y, et al. (2017) Online sequential prediction of bearings imbalanced fault diagnosis by extreme learning machine. *Mechanical Systems and Signal Processing* 85: 450–473.

Qin WL, Zhang WJ and Lu C (2016) A method for aileron actuator fault diagnosis based on PCA and PGC-SVM. *Shock and Vibration* 2016: 1–12.

Razavi-Far R, Hallaji E, Farajzadeh ZM, et al. (2019) Information fusion and semi-supervised deep learning scheme for diagnosing gear faults in induction machine systems. *IEEE Transactions on Industrial Electronics* 66(8): 6331–6342.

Roskam J (2001) *Airplane Flight Dynamics and Automatic Flight Controls*. Lawrence: Design, Analysis and Research Corporation.

Song S, Lan C, Xing J, et al. (2018) Spatio-temporal attention-based LSTM networks for 3D action recognition and detection. *IEEE Transactions on Image Processing* 27(7): 3459–3471.

Tidriri K, Chatti N, Verron S, et al. (2016) Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control* 42: 63–81.

Wang L, Zhao X, Wu J, et al. (2017) Motor fault diagnosis based on short-time Fourier transform and convolutional neural network. *Chinese Journal of Mechanical Engineering* 30(6): 1357–1368.

Yang B, Liu R, Zio E, et al. (2019a) Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Transactions on Industrial Electronics* 66(12): 9521–9530.

Yang Q, Ruan J, Zhuang Z, et al. (2019b) Condition evaluation for opening damper of spring operated high-voltage circuit breaker using vibration time-frequency image. *IEEE Sensors Journal* 19(18): 8116–8126.

You ZH, Ye JM, Li KM, et al. (2019) Adversarial noise layer: Regularize neural network by adding noise. In: *26th IEEE International Conference on Image Processing (ICIP)*, Taipei, China, 22–25 September 2019, pp. 909–913. Piscataway, NJ: IEEE.

Youssef A, Delpha C, Diallo D, et al. (2016) An optimal fault detection threshold for early detection using Kullback-Leibler divergence for unknown distribution data. *Signal Processing* 120: 266–279.

Zagoruyko S and Komodakis N. (2017) Deep compare: A study on using convolutional neural networks to compare image patches. *Computer Vision and Image Understanding* 164: 38–55.

Zhang C, Feng J, Hu C, et al. (2020a) An intelligent fault diagnosis method of rolling bearing under variable working loads using 1-D stacked dilated convolutional neural network. *IEEE Access* 8: 63027–63042.

Zhang QH (2018) Adaptive Kalman filter for actuator fault diagnosis. *Automatica* 93: 333–342.

Zhang R, Zhu F, Liu J, et al. (2020b) Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis. *IEEE Transactions on Information Forensics and Security* 15: 1138–1150.

Zhang Y, Ye D, Liu Y, et al. (2018) Machinery fault diagnosis via an improved multi-linear subspace and locally linear embedding. *Transactions of the Institute of Measurement and Control* 40(14): 4014–4026.

Zhao Z, Chen W, Wu X, et al. (2017) LSTM network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems* 11(2): 68–75.