


Article

Method for Fault Diagnosis of Temperature-Related MEMS Inertial Sensors by Combining Hilbert–Huang Transform and Deep Learning

Tong Gao ¹, Wei Sheng ^{1,*}, Mingliang Zhou ² , Bin Fang ³, Futing Luo ³ and Jiajun Li ³

¹ School of Instrumentation Science and Opto-electronics Engineering, Beihang University, 37, Xueyuan Road, Haidian District, Beijing 100083, China; by1517120@buaa.edu.cn

² State Key Laboratory of Internet of Things for Smart City Faculty of Science and Technology, University of Macau, Macau SAR 999078, China; mingliangzhou@cqu.edu.cn

³ School of Computer Science, Chongqing University, 174 Shazheng Street, Shapingba District, Chongqing 400044, China; fb@cqu.edu.cn (B.F.); lft@cqu.edu.cn (F.L.); lijiajun@cqu.edu.cn (J.L.)

* Correspondence: shengwei@buaa.edu.cn

Received: 16 July 2020; Accepted: 28 September 2020; Published: 1 October 2020



Abstract: In this paper, we propose a novel method for fault diagnosis in micro-electromechanical system (MEMS) inertial sensors using a bidirectional long short-term memory (BLSTM)-based Hilbert–Huang transform (HHT) and a convolutional neural network (CNN). First, the method for fault diagnosis of inertial sensors is formulated into an HHT-based deep learning problem. Second, we present a new BLSTM-based empirical mode decomposition (EMD) method for converting one-dimensional inertial data into two-dimensional Hilbert spectra. Finally, a CNN is used to perform fault classification tasks that use time–frequency HHT spectrums as input. According to our experimental results, significantly improved performance can be achieved, on average, for the proposed BLSTM-based EMD algorithm in terms of EMD computational efficiency compared with state-of-the-art algorithms. In addition, the proposed fault diagnosis method achieves high accuracy in fault classification.

Keywords: fault diagnosis; Hilbert-Huang transform; BLSTM; CNN

1. Introduction

The use of unmanned aerial vehicles (UAVs) is growing due to their advantages in mobility and economics. The reliability achieved by sensors as a measurement and control system has a great impact on the performance of UAVs [1–3]. Inertial sensors in UAVs measure the states of motion. Because micro-electromechanical system (MEMS) inertial sensors have obvious advantages in weight, cost, and power consumption, MEMS inertial sensors are widely used in UAVs to perform inertial measuring tasks. However, the performance of MEMS inertial sensors can be significantly affected by external temperature [4–6] so diagnosing this type of fault is crucial in guaranteeing the reliable control of UAVs.

Fault diagnosis (FD) performs tasks such as fault detection [7–9] and fault tolerance [10,11]. There are three main types of FD methods: hardware-redundant, model-based, and data-driven methods. Hardware-redundant FD handles the faults using redundant devices [12–14], which brings additional costs and weight consumption. Model-based FD adopts analytical models [15–19] to pattern the faults and relies on the precision and accuracy of mathematical models. Data-driven FD recognizes the fault

according to the fault features found from a large volume of historical data [20–22]. This method has low modeling dependency and has attracted much interest from researchers [23–25].

Data-driven methods have achieved high performance in aircraft FD applications and have focused on aircraft actuator FD [26], aircraft sensors FD [27], aero engine systems FD [28,29], and fault data analysis [30,31]. Deep learning (DL) learns features from big data [32,33] and avoids the complex processes stemming from handcrafted features. CNN is a powerful DL model for handling two-dimensional (2-D) images and has been used in FD research, such as mechanical systems FD [34,35], circuit systems FD [36], and avionics FD [37]. In FD applications, because raw data is often sampled in one-dimensional (1-D) format, researchers have turned to feature extraction operations that construct 2-D features for addressing FD problems using CNNs, such as sliding window [38,39], short time Fourier transform (STFT) [40], discrete wavelet transform (DWT) [41,42], and Hilbert–Huang transform (HHT) [43,44]. Structure health monitoring (SHM) is becoming a research hotspot in which CNN is applied and several methods have been proposed in the field of SHM combining CNN to solve mechanical system SHM problems [45–47].

HHT is a time–frequency analysis method that consists of empirical mode decomposition (EMD) and Hilbert transform (HT), and it offers high performance in analyzing nonstationary signals by adopting EMD to compute intrinsic mode functions (IMFs). Many researchers have adopted HHT-based methods to perform data analysis tasks [48–50]. The FD methods combining HHT and CNN are becoming research hotspots in different applications, such as power distribution system FD [51], geared system FD [52] and pipeline system FD [53]. Our research aims to recognize fault patterns hidden in large volume of real MEMS inertial sensors measurement data that are non-stationary due to the affection of variable temperature condition, study on HHT-based feature extraction algorithm offers a reliable fault feature representation strategy on handling nonstationary MEMS temperature related faults.

LSTM [54] is a type of recurrent neural network (RNN) used in serial information processes. As a DL model for processing sequential information, LSTM can directly process the signals of FD, which is the reason of the frequently usage of LSTM in FD study, such as useful life prediction [55], turbine FD [56], and gearbox FD [57].

Analysis on large volume of fault data of our MEMS inertial sensors shows that the temperature related MEMS faults have obvious non-linear and non-stationary characters, DL-based methods offer an effective way to recognize these faults in end-to-end manner. Although the above methods focus on applying deep learning methods to FD, these methods still have the following shortcomings for the UAV MEMS inertial sensors FD:

First, there is little research focused on adopting DL-based FD methods to solve the problems of MEMS inertial sensor FD. Second, traditional 1-D to 2-D data converting methods have limitations in handling nonstationary and nonlinear MEMS measurements. Sliding window-based methods can hardly acquire temperature-related fault features without a full range of temperature variation. Fourier transform-based methods such as STFT and DWT have limitations on processing nonstationary signals. Third, HHT has advantages in processing nonlinear and nonstationary signal, but current HHT-based methods need much iterative computation caused by the two-layer loops iteration method, which decreases computing efficiency. Fourth, LSTM is a powerful tool for performing sequential related tasks, and it can introduce the ability of using prior knowledge to HHT, which is useful in increasing HHT efficiency. However, there are no studies focusing on this strategy.

To address these problems, we have proposed a MEMS inertial sensor fault diagnosis method by combining BLSTM-based HHT and CNN. The contributions of our work are as follows:

- (1) We formulate the MEMS inertial sensor fault diagnosis method for deep learning problems. The proposed FD method uses the BLSTM-based HHT method to perform the feature extracting task and adopts a CNN to classify MEMS inertial sensor faults.
- (2) We propose a BLSTM-based HHT algorithm that performs a direct IMF computation method by introducing BLSTM to EMD, thereby improving the EMD efficiency by decreasing the number of

iterations in obtaining IMFs. Additionally, we adopt noise assistance and frequent shifting to further improve the HHT performance.

- (3) We use the multiscale CNN (MS-CNN) to perform the fault classification task, in which Hilbert spectrums generated by the proposed BLSTM-based HHT are used as the input of the MS-CNN model.

The remainder of this paper is constructed as follows: Section 2 reviews the related works on the proposed FD method. Section 3 describes the proposed FD method. Section 4 presents the detailed experiments. Section 5 presents the authors' conclusions.

2. Related Works

2.1. CNN-Based Fault Diagnosis

CNNs have various applications in image processing and are used in data-driven FD methods for complex systems. In [20], Guo et al. proposed a method for FD of UAV sensors by jointing an extended Kalman filter, STFT, and CNN. This FD method adopted CNNs to recognize various fault features, and the STFT is used to convert the 1-D signals to 2-D spectrums. In [34], Yao et al. presented a multi-scale CNN-based gear FD method. This method designs an attention mechanism based on multi-scale CNN to mine the relevant fault information and uses multi-scale CNN to recognize the faults. In [35], a hierarchical CNN-based FD method was proposed by Guo et al. for performing a mechanical FD task. In [36], Wang et al. proposed a 1-D CNN (1D-CNN)-based Hiller system fault diagnosis method that combines 1D-CNN and a gated recurrent unit to perform the fault identification task. In [37], a random oversampling-based CNN FD method was presented by Chen et al. to handle fault confusing problems and is applied in avionics FD. In [38], Wen et al. presented a CNN-based FD algorithm in which a CNN is used to learn fault features from reconstructed raw data directly without complex feature extracting operation. In [41], Aziz et al. adopted a 2-D CNN to perform the task of fault detection in photovoltaic (PV) arrays, with the CNN used to extract fault features in PV system scalograms.

SHM monitors the failure or degeneration of components in complex systems and plays an important role in maintenance and activation inspection, researcher have used CNN to perform the complex SHM tasks. In [45], Tabian et al. proposed a CNN-based meta-model to address the impact monitoring problem of complex composite structures, the method transferred the piezoelectric sensors signal to 2D images and used CNN to perform the health state classifications, this method has advantages in effective end-to-end state monitoring and well transferability to complex structures. In [46], Oliveira et al. proposed a novel electromechanical impedance SHM solution by combining electromechanical impedance piezoelectricity (EMI-PZT), the high accuracy of proposed method is guaranteed by CNN-based feature extraction which include several banks of filters. In [47], a graphic model and CNN-based bolt loosening SHM method was proposed by Pham et al. The proposed method performed a CNN-based bolt detection algorithm and used the Hough transfer-based method to estimate the bolt angle, this method proposed a convenient strategy to monitoring the bolt structures just using images sampled by a camera.

2.2. HHT-Based Fault Diagnosis

HHT is a new time–frequency method based on empirical mode decomposition (EMD) and Hilbert transform (HT), that has advantages in decomposing the frequency components of nonlinear and nonstationary signals.

Researchers are using HHT to perform signal analysis. In [48], Bagherzadeh et al. enhanced the EMD algorithm using multi-object optimization, which introduced a genetic algorithm to determine the best decision parameters. The proposed method is used to detect ice formation on aircraft. In [49] Zheng et al. presented a flutter test method using HHT, and the method mitigated the mode mixing

effect in HHT. In [50], an improved EMD, called IEEMD, added Gaussian noise into the EMD operation. The method, presented by Mokhtari et al., was used to identify the dynamic models of aircrafts.

As a time–frequency-based data converting method, the output of HHT can include the features to train the models in data-driven FD. In [44], Sheng et al. proposed a fault location method for power systems that employed the high frequency components of EMD to train CNNs to learn the features of the faults. In [58], a modified HHT constructed time- and frequency-domain-based nonlinear entropy features to reduce the effects of noise. Researchers have studied FD methods that combine HHT-based feature extraction and deep learning-based FD methods. In [43], Yang et al. proposed an opening damper evaluation method, which used HHT to convert a 1-D signal into 3-D time-frequency images and used these images for training CNNs in order to evaluate the vibration of power transmission systems. In [59], Liang et al. used the intrinsic mode function (IMF) components of EMD to construct the fault features, which were then used to train the CNN-based FD network, which was called CRNN. In [60], Chen et al. proposed an EMD-based decomposing method, called adaptive sparsest narrow-band decomposition (ASNBD) and applied the ASNBD to FD in roller bearings. In [51], a power distribution Fault classification method combining HHT and CNN is proposed by Guo et al., the proposed method adopted HHT to construct the time-frequency energy map and used CNN to classify the faults patterns. In [52], Han et al. employed a HHT-CNN- based method to address the geared system FD problem, the presented FD method used the two-dimension HHT spectrum of vibration acceleration signals as the input of CNN and the trained CNN is used to perform the fault classification task. In [53], Xie et al. used the HHT and CNN to address the pipeline leakage detection problem, in this paper, acoustic signals are converted to time-frequency image waves by HHT and the converted images are fed to a two-layer CNN for performing the leakage detecting task.

2.3. LSTM-Based Fault Diagnosis

LSTM is a well-known RNN model that can process data combined with contextual information and has been successfully applied in serial information processing. LSTM also has been used in FD-related research. In [61], Huang et al. adopted a BLSTM-based method to solve the prognostic problem of aircraft engine remaining useful life (RUL). In [62], Yang et al. proposed a method for FD for aircraft electromechanical actuators that used LSTM to analyze the correlation of sensors. In [63], an aircraft engine degradation assessment and RUL prediction framework based on LSTM was proposed by Miao et al.

3. Method for Fault Diagnosis of MEMS Inertial Sensors in Unmanned Aerial Vehicles by Combining Hilbert–Huang Transform and Deep Learning

3.1. Overview

The proposed FD method performs an end-to-end FD strategy that uses the inertial data (from the measurements of a gyroscope or accelerometer) as input and outputs the fault classifications. The proposed FD method includes two operations: feature extraction and fault classification. The feature extraction task converts the one-dimension inertial measurement data to a time-frequency spectrum by the proposed BLSTM-based HHT. The fault classification task classifies the inertial sensors' fault states by CNN.

The proposed FD method is shown in Figure 1. First, the inertial data is processed by a frequency shifting operation. Second, a BLSTM-based EMD algorithm is performed to obtain the IMFs, and the noise assistance algorithm is used to process the signals of computing different IMFs. Third, we use Hilbert transformation to construct the Hilbert spectrum, which is used to feed the 2-D features to the CNN. Finally, the MS-CNN outputs the fault classifications.

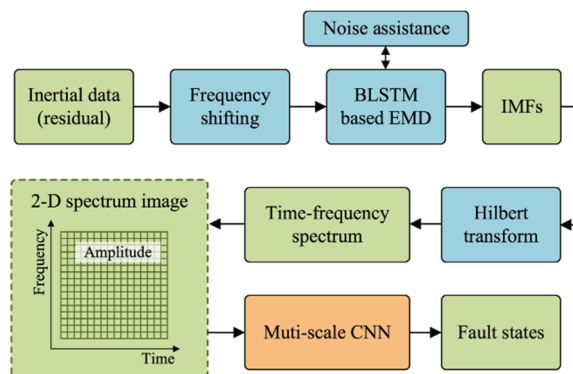


Figure 1. Process of the proposed fault diagnosis method.

3.2. Proposed BLSTM-Based HHT

In this section, we formulate the proposed HHT method. The proposed HHT structure is shown in Figure 2. First, the frequency shifting [49] is applied to reduce the mode mixing. Second, the BLSTM-based EMD is performed to compute the IMFs of inertial data and noise assistance analysis. Ensemble empirical mode decomposition (EEMD) [1] is used to further reduce the mode mixing. Finally, the Hilbert-transform (HT) is performed so that the IMFs can obtain the time–frequency spectrum.

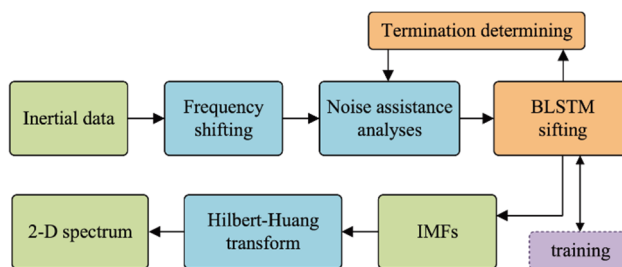


Figure 2. Proposed BLSTM-based HHT structure.

3.2.1. Modeling of End-to-End EMD

In traditional EMD, IMFs are obtained by two iteration loops. The inner loop computes each IMF by trend signal iteration, while the outer loop removes the IMF from the raw signal for the purpose of computing the next IMF. The inner loop is iterated as:

$$h_{i,1}(t) = r_i(t) \tag{1}$$

$$h_{i,2}(t) = h_{i,1}(t) - m_{i,1}(t) \tag{2}$$

$$h_{i,j+1}(t) = h_{i,j}(t) - m_{i,j}(t) \tag{3}$$

where $h_{i,j}(t)$ indicates the weakened component of the j -th iteration for computing the i th IMF, $m_{i,j}(t)$ is the trend signal in each iteration, which is the average of the upper and lower envelop curve of weakened residual $h_{i,j-1}(t)$, as follows:

$$m_{i,j}(t) = \frac{u_{i,j}(t) - l_{i,j}(t)}{2} \tag{4}$$

where $u_{i,j}(t)$ and $l_{i,j}(t)$ are the upper and lower values of the envelop curve of $h_{i,j}(t)$ by crossing the cubic splines.

Once the trend signal is removed from weakened component $h_{i,j}(t)$, the i -th IMF can be obtained. Then the inner iteration is stopped and the final weakened component is deemed as the i -th IMF. Then EMD removes the i -th IMF from the residual $r_i(t)$ and obtains the new residual, as follows:

$$r_{i+1}(t) = r_i(t) - IMF_i \quad (5)$$

where IMF_i indicates the i -th IMF.

During the EMD iterations, the trend signal $m_{i,j}(t)$ is the residual containing the nonlinear and the non-zero average components. We can use polynomial fitting to indicate the trend signal. Assume that the best weakened component is obtained in the J -th iteration, the final iteration is:

$$IMF_i = h_{i,J}(t) = h_{i,J-1}(t) - m_{i,J-1}(t) \quad (6)$$

The $(J-1)$ -th weakened component can be written by the $(J-2)$ -th weakened component as:

$$h_{i,J-1}(t) = h_{i,J-2}(t) - m_{i,J-2}(t) \quad (7)$$

Thus, the IMF_i can be expressed as the $(J-2)$ -th weakened component and the sum of the $(J-2)$ -th and the $(J-1)$ -th trend signals:

$$\begin{aligned} IMF_i &= h_{i,J-1}(t) - m_{i,J-1}(t) \\ &= h_{i,J-2}(t) - m_{i,J-2}(t) - m_{i,J-1}(t) \end{aligned} \quad (8)$$

Furthermore, IMF_i can be deemed as the sum of the initial weakened component $h_{i,1}(t) = r_i(t)$ and the negative total trend signal that contains all trend signals, from the first to the last:

$$\begin{aligned} IMF_i &= h_{i,1}(t) \\ &\quad - (m_{i,1}(t) + m_{i,2}(t) + \dots + m_{i,j}(t) + \dots + m_{i,J-1}(t)) \\ &= r_i(t) - \sum_{j=1}^{J-1} m_{i,j}(t) \\ &= r_i(t) - M_i(t) \end{aligned} \quad (9)$$

where $M_i(t)$ indicates the sum of all trend signals of the iterations. It is worth noting that this iteration is necessary for irregular signals; we cannot obtain the $M_i(t)$ directly according to $r_i(t)$ without specific abstract features. However, there are specific abstract features in the inertial data, which makes directly obtaining $M_i(t)$ from $r_i(t)$ feasible by training an end-to-end regression model, as follows:

$$\begin{cases} M_i(t) = f(r_i(t), \Theta) \\ M_i(t) = r_i(t) - IMF_i \end{cases} \quad (10)$$

where Θ indicates the parameters of the regression model.

Obviously, this method decreases the iteration epochs for computing IMF. In this regression model, $M_i(t)$, which is obtained according to $r_i(t)$ and the IMF_i by inner loop sifting, can be treated as the training label and the $r_i(t)$ is the input feature. The IMF_i can also be obtained by the j -th weakened component and trend signals from the $(j+1)$ -th to the end. Thus, Equations (9) and (10) could be expressed by Equations (11) and (12), as follows:

$$\begin{aligned} IMF_i &= h_{i,j}(t) - (m_{i,j+1}(t) + \dots + m_{i,J-1}(t)) \\ &= h_{i,j}(t) - \sum_{\beta=j+1}^{J-1} m_{i,\beta}(t) \\ &= h_{i,j}(t) - M_{i,j}(t) \end{aligned} \quad (11)$$

$$\begin{cases} M_{i,j}(t) = f(h_{i,j}(t), \Theta) \\ M_{i,j}(t) = h_{i,j}(t) - IMF_i \end{cases} \quad (12)$$

3.2.2. BLSTM-Based Sifting

We formulate the traditional sifting process of EMD into an end-to-end regression task. In this research, we propose a BLSTM-based coefficient regression algorithm to obtain the trend signal $M_i(t)$ from $r_i(t)$ directly.

Equation (10) represents an efficient way to obtain the training data set, in that the regression label, trend signal $M_i(t)$, can be obtained by removing IMF_i from $r_i(t)$.

$M_i(t)$ can be represented by piecewise segments, as follows:

$$M_i(t) = \sum_{k=1}^{K-1} M_i^H(k) \quad (13)$$

where $k \in \{1, \dots, K-1\}$ indicates the number of segments of trend signal $M_i(t)$, which is divided by local extrema points of $r_i(t)$, and $M_i^H(k)$ indicates the piecewise trend signal that is computed using cubic polynomial fitting as follows:

$$M_i^H(k) = \begin{cases} a_{i,k}^H t^3 + b_{i,k}^H t^2 + c_{i,k}^H t + d_{i,k}^H, t_{k-1} \leq t < t_k \\ 0, otherwise \end{cases} \quad (14)$$

where $(a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H)$ indicate the coefficients of fitting in each segment. The local extrema points are generated from $r_i(t)$, and this makes each segment in $r_i(t)$ monotonic, so that it can be easily fitted by a cubic polynomial. Furthermore, the trend signal is the residual by removing the high-frequency IMF_i from $r_i(t)$, which means that piecewise fitting can aptly describe a “low frequency” trend signal. Note that Equation (14) presents a strategy to code data in each segment to a fixed-dimension expression, which is essential for BLSTM.

Thus, according to Equations (10), (13), (14), the regression task is to recognize the piecewise cubic fitting parameters $(a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H)$ from the piecewise $r_i(t)$. We use BLSTM to address this regression problem, which is described as:

$$(a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H) = BLSTM(r_i(k), \Theta), k \in \{1, 2, \dots, K-1\} \quad (15)$$

where $r_i(k)$ indicates the piecewise $r_i(t)$ by local extrema points, k indicates the segment index, and K is the amount of local extrema points.

Different IMFs have different scales, which lead to reduced performance in the convergence and learning efficiency in the case of feeding unnormalized data into the neural network directly. Thus, we adopt the min-max normalization method to normalize the weakened component data, and the normalization and piecewise operation are:

$$\begin{cases} \hat{r}_i(t) = \frac{r_i(t) - \min(r_i(t))}{\max(r_i(t)) - \min(r_i(t))} \\ \hat{r}_i(k) = \text{piecewise}(\hat{r}_i(t)), k \in \{1, 2, \dots, K-1\} \end{cases} \quad (16)$$

The length of each piecewise weakened component is different due to the nonstationary and the non-linear nature of the raw signal. To meet the fixed input dimension length requirement of BLSTM,

we first perform a cubic polynomial fitting on $\hat{r}_i(k)$, then the coefficient of $\hat{r}_i(k)$ is used as the input of BLSTM. The operation is as follows:

$$\begin{cases} \hat{r}_i(k) = \begin{cases} p_{i,k,1}t^3 + p_{i,k,2}t^2 + p_{i,k,3}t + p_{i,k,4}, t_{k-1} \leq t < t_k \\ 0, otherwise \end{cases} \\ p_i(k) = [p_{i,k,1}, p_{i,k,2}, p_{i,k,3}, p_{i,k,4}] \\ \quad = \text{cubic polynomial fitting}(\hat{r}_i(k)) \end{cases} \tag{17}$$

where $\hat{r}_i(k)$ indicates the normalized segment obtained by local extrema points, and $p_i(k)$ indicates the polynomial fitting coefficient.

Thus, the BLSTM model in Equation (15) can be presented as:

$$\begin{aligned} p_i^H(k) &= (a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H) \\ &= \text{BLSTM}(p_i(k), \Theta), k \in \{1, 2, \dots, K-1\} \end{aligned} \tag{18}$$

Additionally, the template conditions of inner loop sifting can augment the robustness of the recognition pattern, and the weakened component-based trend signal regression model can be presented as:

$$\begin{cases} \hat{h}_{i,j}(k) = \begin{cases} q_{i,j,k,1}t^3 + q_{i,j,k,2}t^2 + q_{i,j,k,3}t + q_{i,j,k,4}, t_{k-1} \leq t < t_k \\ 0, otherwise \end{cases} \\ q_{i,j}(k) = [q_{i,j,k,1}, q_{i,j,k,2}, q_{i,j,k,3}, q_{i,j,k,4}] \\ \quad = \text{cubic polynomial fitting}(\hat{h}_{i,j}(k)) \end{cases} \tag{19}$$

$$\begin{aligned} q_{i,j}^H(k) &= (a_{i,j,k}^T, b_{i,j,k}^T, c_{i,j,k}^T, d_{i,j,k}^T) \\ &= \text{BLSTM}(q_{i,j}(k), \Theta), k \in \{1, 2, \dots, K-1\} \end{aligned} \tag{20}$$

where the annotation i, j indicates the j th iteration for computing i th *IMF*, $\hat{h}_{i,j}(k)$ indicates the normalized weakened component, $q_{i,j}^H(k)$ indicates the cubic polynomial coefficients of $\hat{h}_{i,j}(k)$, k indicates the k th segment, and $q_{i,j}^H(k)$ indicates the output of BLSTM model with input $q_{i,j}(k)$.

The structure of the proposed BLSTM-based sifting method is shown in Figure 3. The proposed BLSTM-based sifting algorithm is based on a deep BLSTM regression model that consists of several BLSTM layers and one output layer. The neurons in the BLSTM layer can be modified to mine detailed features. The output layer consists of a three-layer fully connected network with 32 neurons in the first layer, 16 neurons in the second layer, and four neurons in the third layer. The output layer is used to reconstruct the features from the BLSTM layers to four dimensions, which is equal to the dimensions of polynomial parameters of the trend signal.

The input of the BLSTM model can be formulated as:

$$\begin{aligned} \vec{X} = \overleftarrow{X} &= [\vec{x}(1), \vec{x}(2), \dots, \vec{x}(k), \dots, \vec{x}(K-1)] \\ &= [\overleftarrow{x}(1), \overleftarrow{x}(2), \dots, \overleftarrow{x}(k), \dots, \overleftarrow{x}(K-1)] \\ &= [p_i(1), p_i(2), \dots, p_i(k), \dots, p_i(K-1)] \end{aligned} \tag{21}$$

where $\vec{x}(k)$ and $\overleftarrow{x}(k)$ are the inputs of the forward procedure and the backward procedure and $K-1$ indicates the total number.

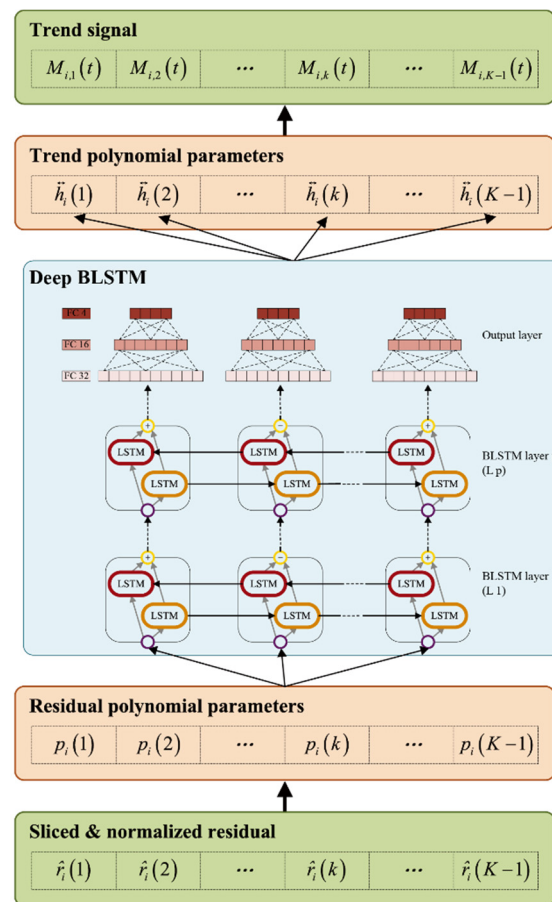


Figure 3. BLSTM-based sifting algorithm.

The regression task of the BLSTM is formulated as follows:

$$\begin{aligned} \overleftrightarrow{H} &= [\overleftrightarrow{h}(1), \overleftrightarrow{h}(2), \dots, \overleftrightarrow{h}(k), \dots, \overleftrightarrow{h}(K)] \\ &= f(\overleftrightarrow{X}; \overleftrightarrow{\Theta}_{LSTM}) \oplus f(\overleftarrow{X}; \overleftarrow{\Theta}_{LSTM}) \end{aligned} \quad (22)$$

where $\overleftrightarrow{h}(k)$ indicates the output of BLSTM, which is the coefficient of piecewise trend signal:

$$\overleftrightarrow{h}(k) = (a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H) \quad (23)$$

3.2.3. BLSTM Training Strategy

The performance of BLSTM-based EMD is dependent on the training dataset. We propose a strategy to generate the dataset, which is a modified EMD operation that records data and combines various efficient sifting methods to obtain the optimal IMFs.

Figure 4 indicates the modified EMD with data recording operation. Note that the symbol “ \sim ” indicates the normalized signal. The proposed three-part method includes the proposed EMD, feature coding, and feature recording. The proposed EMD consists of frequency shifting, classic EMD, and noise assistance analysis. The first step is the frequency shifting operation [49]. The second step is normalization, as seen in Equation (16), to obtain the normalized residual $\hat{r}_i(t)$; then the processed data is fed to the classic EMD framework. Note that the classic EMD in Figure 4 is represented as an inner loop and outer loop. The inner loop is used to compute each IMF and the outer loop is used to compute the IMFs. The outer loop of the proposed EMD adopts the EEMD [1] to reduce the mode

mixing. The feature coding operation converts the data of the improved EMD to the four-dimension fitting parameters. The first operation of the feature coding is piece wiseing the residual $\hat{r}_i(t)$. Each $\hat{r}_i(t)$ in an EMD operation is piecewisely by the extrema points of the original signal $\hat{x}(t) = \hat{r}_1(t)$, as follows:

$$\{Ex\} = \{ex_1(1), ex_1(2), \dots, ex_1(k), \dots, ex_1(K-1)\} \quad (24)$$

where $ex_1(k)$ indicates the time stamp of each local extrema of $\hat{r}_1(t)$. EMD is a method that removes frequency components from residual, which means that the local extrema points $\{Ex\}$ that partition a high-frequency signal into $(K-1)$ monotonical intervals can partition low-frequency weakened components without any information missing. After the $\{Ex(i)\}$ is obtained, the weakened component $\hat{h}_{i,j}(t)$, temporary trend $\hat{M}_{i,j}(t) = \hat{h}_{i,j}(t) - IMF_i$, and trend $\hat{M}_i(t) = \hat{r}_i(t) - IMF_i$ are piecewisely according to $\{Ex\}$. The final step of feature coding is cubic polynomial fitting each segment of these piecewisely signals, and this is what generates the polynomial coefficients. The data recording operation creates the dataset for training the BLSTM model.

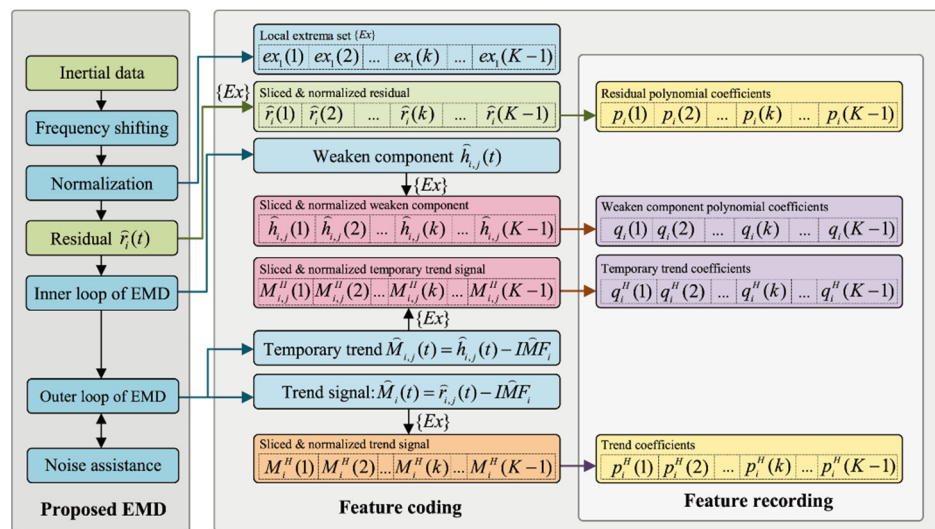


Figure 4. Modified EMD with data recording.

The basic dataset consists of the residual polynomial coefficients and the trend coefficients. The residual polynomial coefficients, $p_i(k) = \{p_{i,k,1}, p_{i,k,2}, p_{i,k,3}, p_{i,k,4}\}$, are the training features. The trend coefficients, $p_i^H(k) = \{a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H\}$, are the regression labels. The extension dataset consists of the “weakened component polynomial coefficients” and the “temporary trend coefficients”. The weakened component polynomial coefficients, $q_{i,j}(k) = \{q_{i,j,k,1}, q_{i,j,k,2}, q_{i,j,k,3}, q_{i,j,k,4}\}$, are the training features. The temporary trend coefficients, $q_{i,j}^H(k) = \{a_{i,j,k}^T, b_{i,j,k}^T, c_{i,j,k}^T, d_{i,j,k}^T\}$, are the regression labels. The extension dataset contains the temporary details of EMD, and it can be used to improve the robustness of BLSTM.

3.2.4. IMF Computation

After the BLSTM model is trained using the basic dataset and the extension dataset, a direct EMD, as in Equation (10), can be driven by the BLSTM, as in Algorithm 1. The BLSTM-based EMD method performs a BLSTM regression operation. First, the residual is normalized and piecewisely by local extrema points. Second, the cubic parameters of each segment are computed. Third, the BLSTM model is performed to regress the trend signal. Then, the piecewise trend signal is computed by the regressed parameters. Finally, the IMF is computed by removing the recovered trend signal from the residual.

3.3. CNN Structure

After the 2-D spectrum is obtained by HT using IMFs, the CNN is used to learn and recognize the fault patterns. We use MS-CNN within skipping connection to perform this task. The structure of the MS-CNN is shown in Figure 5. The MS-CNN has four convolutional layers, three max-pooling layers, one multiscale layer, two fully connecting (FC) layers and one output layer. The size of the input 2-D image is 32×32 . The convolution kernel size of the first three convolutional layers is 3×3 , and the kernel size of the last convolutional layer is 2×2 . The layer configuration is described in Table 1. A rectified linear unit (RELU) is used as the nonlinear activation function.

Algorithm 1: BLSTM-based EMD

Requirement: Trained BLSTM-based EMD model by dataset in Figure 4.

Input: Inertial measurement residual $r_i(t)$.

Output: $IMF_i(t)$.

Begin:

- 1: Normalize residual: $\hat{r}_i(t) = \frac{r_i(t) - \min(r_i(t))}{\max(r_i(t)) - \min(r_i(t))}$;
 - 2: Piecewise: $\{r_i(k)\} = \text{piecewise}(r_i(t))\{Ex\}$;
 - 3: **for** k in $\{1, 2, \dots, (K-1)\}$ **do**:
 - 4: Fitting p_i :
 - 5: $p_i(k) = \{p_{i,k,1}, p_{i,k,2}, p_{i,k,3}, p_{i,k,4}\} = \text{polynomial}(r_i(k))$;
 - 6: **end for**
 - 7: Regress trend coefficient:
 - 8: $\{a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H\} = \text{BLSTM}(p_i(k), \Theta) | k \in \{1, 2, \dots, (K-1)\}$;
 - 9: **for** k in $\{1, 2, \dots, (K-1)\}$ **do**:
 - 10: Compute piecewise trend signal:

$$\hat{M}_{i,k}^H(t) = \begin{cases} a_{i,k}^H t^3 + b_{i,k}^H t^2 + c_{i,k}^H t + d_{i,k}^H t_{k-1} & t_{k-1} \leq t < t_k \\ 0, & \text{otherwise} \end{cases}$$
 - 11: **end for**
 - 12: Compute trend signal: $\hat{M}_i(t) = \sum_{k=1}^{K-1} \hat{M}_{i,k}^H(t)$;
 - 13: Compute IMF: $\hat{IMF}_i(t) = \hat{r}_i(t) - \hat{M}_i(t)$;
 - 14: Recover IMF:

$$IMF_i(t) = \hat{IMF}_i(t) \times (\max(r_i(t)) - \min(r_i(t))) + \min(r_i(t))$$
 - 15: Update $r_i(t)$: $r_i(t) = r_i(t) - IMF_i(t)$;
 - 16: **End**
-

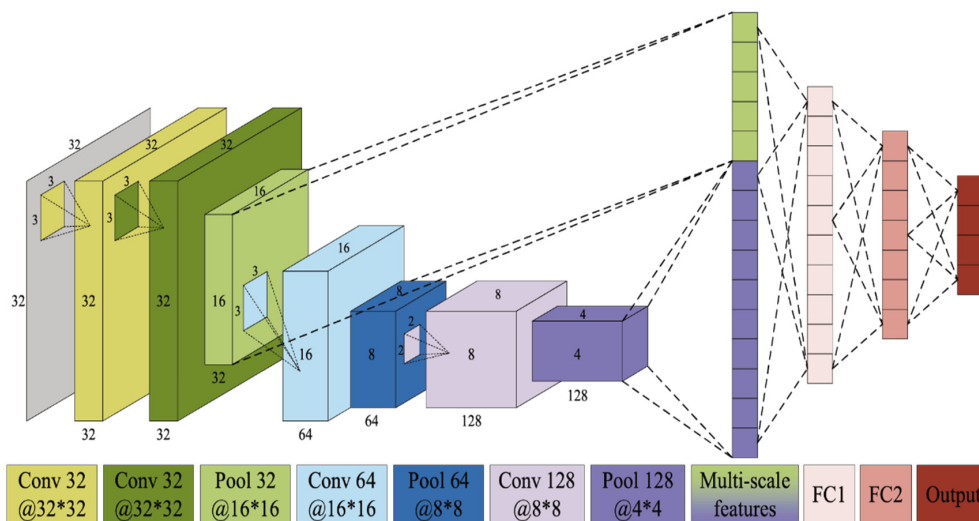


Figure 5. Structure of MS-CNN.

Table 1. Multi-scale CNN configuration.

Layer	Name	Details
1.	Conv1	Conv ($3 \times 3 \times 32$); stride: (1×1)
2.	Conv2	Conv ($3 \times 3 \times 32$); stride: (1×1)
3.	Pool 1	Max pool ($2 \times 2 \times 32$); stride: (1×1)
4.	Conv3	Conv ($3 \times 3 \times 64$); stride: (1×1)
5.	Pool 2	Max pool ($3 \times 3 \times 64$); stride: (1×1)
6.	Conv4	Conv ($2 \times 2 \times 128$); stride: (1×1)
7.	Pool 3	Max pool ($2 \times 2 \times 128$); stride: (1×1)
8.	Multi	Concatenate features from layer 3 and 7
9.	FC1	Fully connect 1
10.	FC2	Fully connect 2
11.	Output	Soft max (4)

3.4. Summary of the Proposed FD Algorithm

We summarize the proposed FD method in Algorithm 2. Note that some procedures, such as frequency shifting and noise assistance, are the same as the methods described in [50,64]. However, the emphases of the BLSTM-based EMD approaches are entirely different. First, we propose the direct trend signal estimating strategy, which avoids the sifting iterating operation and stopping judgment used in current EMD methods. Second, we introduce the BLSTM model to the task of EMD operation by using the piecewise polynomial fitting-based features coding method. This coding method offers an efficient way to feed the variation dimension data to the BLSTM model with fixed data dimension, which is essential for the BLSTM model. Third, an efficient dataset-generating method is proposed, which can automatically generate the dataset for training BLSTM without a complex data annotation operation.

Algorithm 2: Proposed FD method

Requirements:

Trained BLSTM-based EMD model by dataset in Figure 4;

Trained MS-CNN by Hilbert spectrums and corresponding fault labels.

Input: Inertial measurement residual $r_i(t)$.

Output: Fault classifications.

Begin:

```

1:      Frequency shifting:  $r_i(t) = \text{FrequencyShifting}(r_i(t))$ ;
2:      for  $i$  in  $\{1, 2, \dots, I\}$  do:
3:          Compute  $IMF_i$  according to Algorithm I with EEMD;
4:      end for
5:      Compute Hilbert spectrum by HT:  $sp = HT(IMF_i)_{i \in \{1, 2, \dots, I\}}$ ;
6:      Classify fault by MS-CNN:  $fault = MS - CNN(sp)$ ;
7:      End.

```

4. Experiments

In this section, we describe the experiments in which we evaluated the performance of our proposed methods and compared them with other state-of-the-art FD methods. The experiments consisted of BLSTM comparisons, EMD performance evaluations and comparisons, multiscale CNN testing, and FD comparisons. The BLSTM comparison included data recording testing and BLSTM testing, with the data recording testing performed to find the best dataset generating strategy for training BLSTM, and the BLSTM testing comparing various BLSTM structures for finding the best BLSTM model. The EMD performance evaluations of the BLSTM-based EMD involved comparisons with other EMD methods, which included decomposing performance comparison and computing efficiency performance. The multiscale CNN comparison experiment compared the hyperparameters

of the CNN and selected optimal parameters. The FD comparison compared our FD method with other data-driven FD methods on our inertial dataset.

4.1. Setting

We selected four common temperature-related MEMS inertial sensor faults, which occurred in the production of our UAV micro guidance and navigation controllers (UAV-MGNC). The faults are shown in Figure 6. Note that the data are acquired in variable temperature conditions, the polyfit curves are used to remove the test rig-related trend term from samples, and the parting line is used to select the fault data in a temperature section that triggers the faults.

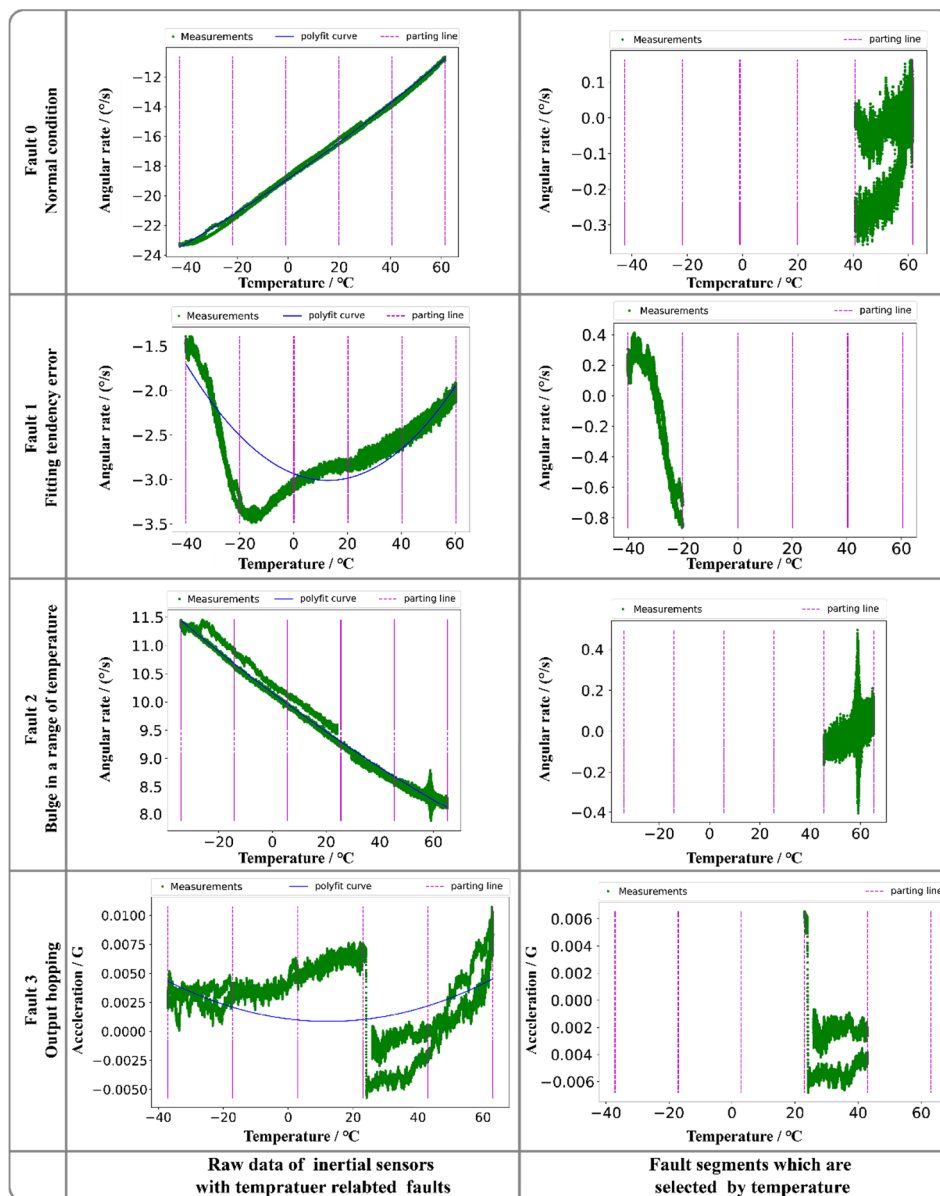


Figure 6. Temperature-related inertial sensors faults.

These four types of conditions from real data contain all typical states of temperature related MEMS inertial faults, precisely recognizing on these faults can address much of temperature related FD problems. For the normal condition, the measurement can be well compensated using polynomial fitting, and the residual is stationary. A fitting tendency error indicates that the residual after compensation

shows the obvious trends, and to fit this error, the UAV controller should use a higher-order model to compensate the measurement, which increases the computation load of the controller. Bulge in a range of temperature is a hotspot in MEMS inertial data compensation filed [65–67], which is caused by the limitation of MEMS technology. Output hopping indicates a step functional characteristic in the measured data that is caused by an abnormal station in the peripheral circuit. We selected the data according to the temperature section in which the fault occurs and performed the EMD operation. The EMD results in traditional EMD are shown in Figure 7.

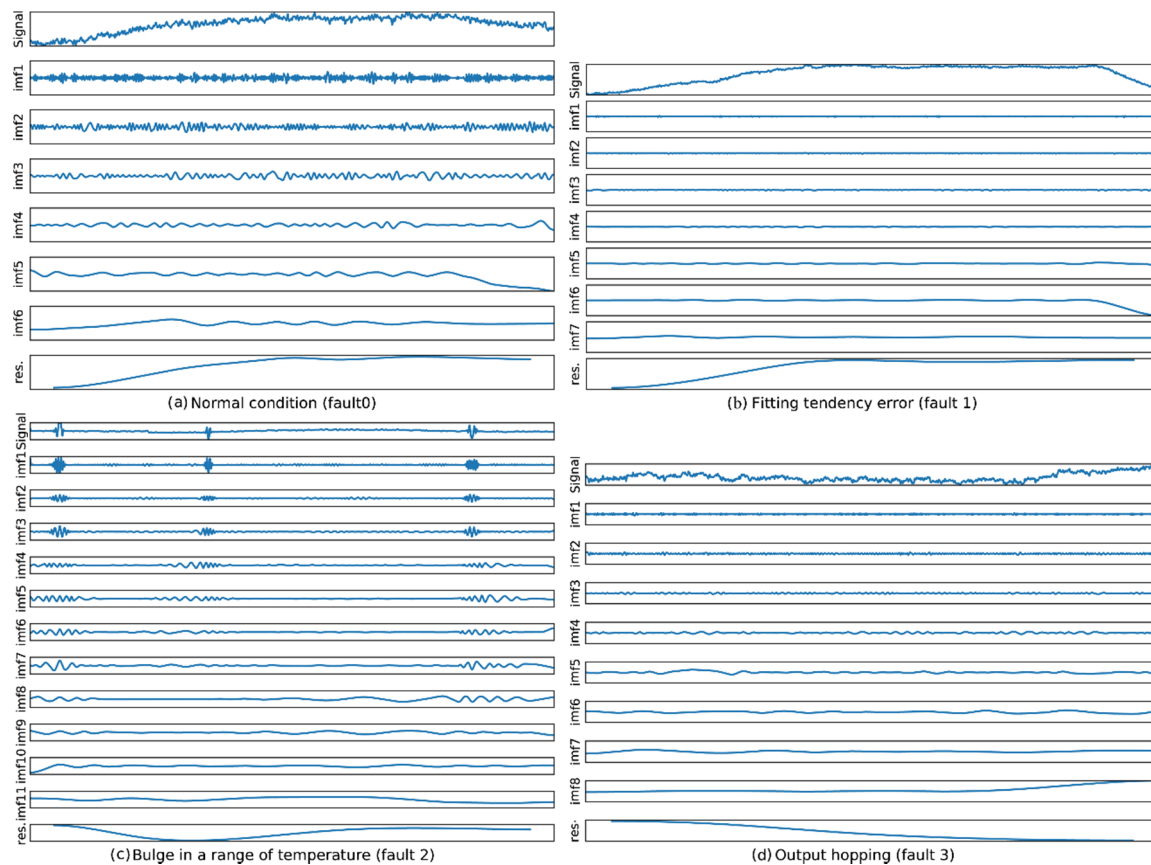


Figure 7. IMFs of each inertial sensor fault. (a) the IMFs in case of normal condition; (b) the IMFs in case of fault 1 occurs; (c) the IMFs in case of fault 2 occurs; (d) the IMFs in case of fault 3 occurs.

The MEMS inertial dataset (which consists of measurements made by gyroscopes and accelerometers) is collected from the calibration operation of our UAV-MGNCs. In the temperature calibration process, the UAV-MGNCs are fixed inside of variable temperature rig. The descriptions of UAV-MGNC and variable temperature rig are shown in Figure 8. The data are acquired in a variable temperature environment from $-40\text{ }^{\circ}\text{C}$ to $68\text{ }^{\circ}\text{C}$. The sample rate is set to 100 Hz and the temperature-related calibration time is set to 4 h. Of the samples, 80% are used as the training set and 20% are used as the testing set. Because the BLSTM-based HHT and CNN are two phases of the proposed FD method, the proposed BLSTM model and CNN model are trained use the same dataset. Table 2 indicates the size of the dataset and the labels.

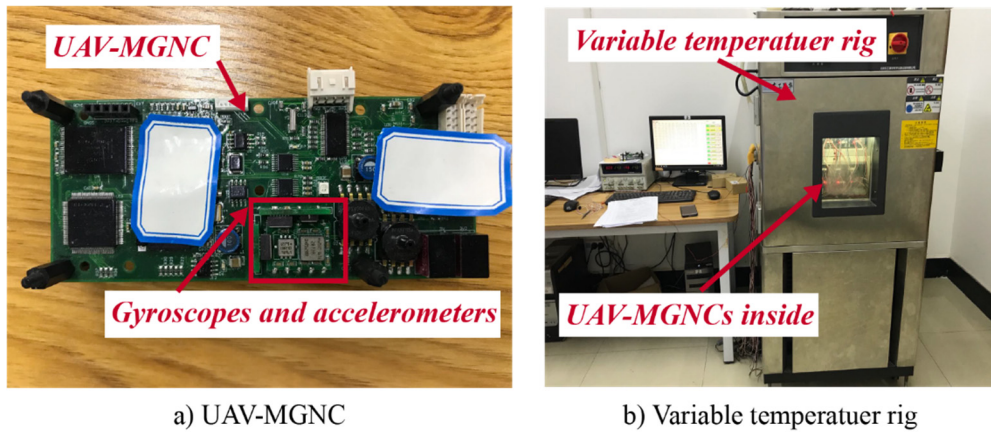


Figure 8. The descriptions of UAV-MGNC and variable temperature rig. (a) Description of UAV-MGNC, gyrosopes and accelerometers are the build-in components of UAV-MGNC; (b) Description of variable temperature rig, UAV-MGNCs are set inside of variable temperature rig.

Table 2. MEMS inertial sensors fault data set.

State	Size	Label	Label
Normal condition	6000	0	1000
Fitting tendency error	6000	1	0100
Bulge in a range of temperature	6000	2	0010
Output hopping	6000	3	0001

4.2. BLSTM Comparison

In this section, we study the two factors that have the greatest influence on the performance of the BLSTM-based EMD method: the dataset and the BLSTM structure. The features contained in the dataset rely on the data recording strategy, and the BLSTM structure relies on the number of hidden layers and the number of neurons in each hidden layer.

We choose the hyperparameters of BLSTM, which include batch size, dropout rate, training epochs, and unfold steps. The best parameters are selected through grid research, which is performed in a deep BLSTM, as shown in Figure 3, with one BLSTM layer which has 128 neurons. The selected hyperparameters are shown in Table 3.

Table 3. Selected parameters.

Parameter	Description	Selected Value
Batch size	Training samples in each training epoch	16
Dropout rate	Dropout probability	0.25
Training epochs	Number of training iterations	200
Unfold steps	Data length of each training sample in time steps	256

4.2.1. Comparison of Data Recording Strategies

We test the effect of the training set with different extension datasets on BLSTM training performance. In the dataset, the feature-label pairs, $\{\hat{p}_i(k), (a_{i,k}^H, b_{i,k}^H, c_{i,k}^H, d_{i,k}^H)\}$, are the basic BLSTM training set. The robust features are learned by using the extension dataset, $\{\hat{h}_{i,J}(k), (a_{i,J,k}^T, b_{i,J,k}^T, c_{i,J,k}^T, d_{i,J,k}^T)\}$. The index J belongs to the set as follows:

$$J \in \{2, 3, 4, \dots, J_s\}, J_s = \text{int}(RI \cdot \max(j)) \quad (25)$$

where $\max(j)$ indicates the maximum iteration numbers of inner loop sifting. J_s indicates the maximum value of J , which is controlled by the parameter $RI \in (0, 1)$. That means all $\left\{ \widetilde{h}_{i,j}(k), \left(a_{i,j,k}^H, b_{i,j,k}^H, c_{i,j,k}^H, d_{i,j,k}^H \right) \right\}$ with J in Equation (25) will be recorded to generate the extension dataset.

The recording process is shown in Table 4. We used a deep BLSTM with two BLSTM layers (one with 128 neurons and one with 192 neurons) to test the different RI . The deep BLSTM is trained by the basic dataset and various extension datasets. The performance of the BLSTM with different RI s is evaluated by training loss (the lower the training loss, the more robust the dataset). The mean square error function is the loss function. The Adam optimizer is the back-propagation algorithm. The training batch is set to 16, and there are 200 training epochs.

Table 4. Recording process details.

RI	Dataset	Name
1/8	$\left\{ \widetilde{h}_{i,j}(k), \left(a_{i,j,k}^H, b_{i,j,k}^H, c_{i,j,k}^H, d_{i,j,k}^H \right) \right\}, J \in [2, \text{int}((1/8) \times \max(j))]$	RI (1/8)
1/7	$\left\{ \widetilde{h}_{i,j}(k), \left(a_{i,j,k}^H, b_{i,j,k}^H, c_{i,j,k}^H, d_{i,j,k}^H \right) \right\}, J \in [2, \text{int}((1/7) \times \max(j))]$	RI (1/7)
1/6	$\left\{ \widetilde{h}_{i,j}(k), \left(a_{i,j,k}^H, b_{i,j,k}^H, c_{i,j,k}^H, d_{i,j,k}^H \right) \right\}, J \in [2, \text{int}((1/6) \times \max(j))]$	RI (1/6)
1/5	$\left\{ \widetilde{h}_{i,j}(k), \left(a_{i,j,k}^H, b_{i,j,k}^H, c_{i,j,k}^H, d_{i,j,k}^H \right) \right\}, J \in [2, \text{int}((1/5) \times \max(j))]$	RI (1/5)
1/4	$\left\{ \widetilde{h}_{i,j}(k), \left(a_{i,j,k}^H, b_{i,j,k}^H, c_{i,j,k}^H, d_{i,j,k}^H \right) \right\}, J \in [2, \text{int}((1/4) \times \max(j))]$	RI (1/4)
1/3	$\left\{ \widetilde{h}_{i,j}(k), \left(a_{i,j,k}^H, b_{i,j,k}^H, c_{i,j,k}^H, d_{i,j,k}^H \right) \right\}, J \in [2, \text{int}((1/3) \times \max(j))]$	RI (1/3)

Figure 9 shows the training losses with different RI s. The lowest values of training loss are decreased with RI increasing. After RI achieves 1/5, the lowest training loss tends to be steady. That confirmed our theory that more related features of $r_i(t)$ enhance the robustness of BLSTM. However, the overlarge RI contains many features that are of no use to enhance the robustness and slow down the convergence speed of training loss. Thus, we use $RI = 1/5$ to generate the dataset and then use this dataset to perform the following experiments.

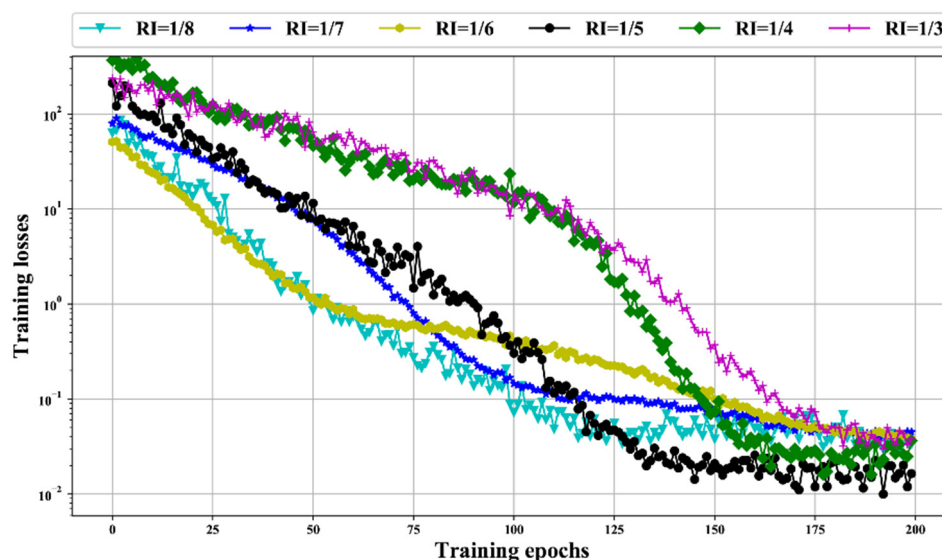


Figure 9. BLSTM training losses in Semi-log coordinate using data sets with various RI s.

4.2.2. BLSTM Structure Testing

We tested the performance of the BLSTM models with different structures that have various hidden layers and hidden neurons in each hidden layer. This experiment was performed in dataset

$RI(1/5)$, and the training labels were the trend signal coefficients shown in Figure 4. The depths 1, 2, 3, and 4 were tested. We compared mean square error training losses and used Adam to train the BLSTM model.

The mean final loss values of each compared BLSTM are listed in Table 5. The mean final loss of each structure is the mean of 10 times training. Note that the BLSTM structures are presented in the format of “ $Lp(q)$,” in which p indicates the layer number of BLSTM layer, and q indicates the number of neurons in BLSTM layer p . For example, “L1(128) L2(256)” indicates a two-layer BLSTM with 128 neurons in the first layer BLSTM and 256 neurons in the second layer BLSTM. The test was started with one-layer BLSTM, based on the structure with the best mean final losses, and then one additional BLSTM layer was added. We tested structures that have layers from one to four. The structure with the best mean final loss is shown in bold in Table 5, and the loss curve with the best structure and the lowest final loss in each group are plotted in Table 5, BLSTM “L1(128) L2(192)” achieves the best mean final loss values of 0.0258 and “L1(128) L2(192) L3(128) L4(128)” has the worst loss of 0.8284. The two-layer BLSTM performs better than others. Figure 10 indicates that “L1(128) L2(192)” can achieve a better convergence performance and obtain a lower loss than the others. Therefore, the BLSTM “L1(128) L2(192)” is used to perform the following experiments.

Table 5. Mean final losses of each BLSTM structure.

Group	BLSTM Structure	Mean Loss
1 layer	L1(64)	0.5198
	L1(128)	0.2109
	L1(192)	0.3917
	L1(265)	0.2392
2 layers	L1(128) L2(64)	0.0317
	L1(128) L2(128)	0.0301
	L1(128) L2(192)	0.0258
	L1(128) L2(256)	0.0279
3 layers	L1(128) L2(192) L3(64)	0.0882
	L1(128) L2(192) L3(128)	0.0791
	L1(128) L2(192) L3(192)	0.1003
	L1(128) L2(192) L3(256)	0.0932
4 layers	L1(128) L2(192) L3(128) L4(64)	0.4503
	L1(128) L2(192) L3(128) L4(128)	0.8284
	L1(128) L2(192) L3(128) L4(192)	0.5017
	L1(128) L2(192) L3(128) L4(256)	0.6697

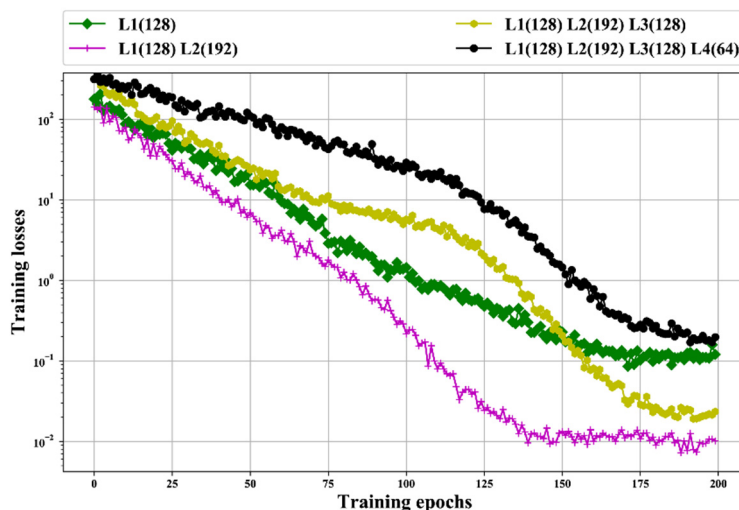


Figure 10. Loss curves in semi-log coordinate with the best structure and the lowest final loss in each group.

4.3. EMD Performance Comparison

We evaluated the performance of the proposed BLSTM-based EMD algorithm, as shown in Figure 4. We compared the proposed two-layer BLSTM-based EMD algorithm, which is trained by the RI(1/5) dataset, to the other EMD methods, with respect to the EMD performance on our MEMS inertial datasets. The compared methods were as follows: traditional EMD [43], multiobjective optimization-based EMD [48], frequency-shifting-based EMD [49] and noise assistance analysis-based EMD [50]. We first evaluated EMD performance that includes the IMFs and Hilbert spectrum of the proposed method and the orthogonality of the IMFs. Then we compared the EMD computing efficiency, which is the time consumption in various data lengths.

4.3.1. Evaluation of EMD Results

We evaluated the EMD performance in terms of decomposition performance on different frequency components. Figures 11 and 12 display the EMD results and the corresponding Hilbert spectrums of the proposed BLSTM-based method. Figure 11 shows that the proposed BLSTM-based EMD can decompose various IMFs of a signal, and the features of each fault can be represented according to the IMFs. Figure 12 shows the Hilbert spectrums of the IMFs. It can be seen that the time–frequency features of each fault differ.

To evaluate the IMF decomposing performance, we compared the orthogonality index criterion presented by Bagherzadeh et al. [48] as follows:

$$OI = \frac{\int_t IMF_i(t) \cdot (r_i(t) - IMF_i(t)) dt}{\int_t r_i(t) dt} \quad (26)$$

where $r_i(t)$ indicates the residual for computing $IMF_i(t)$.

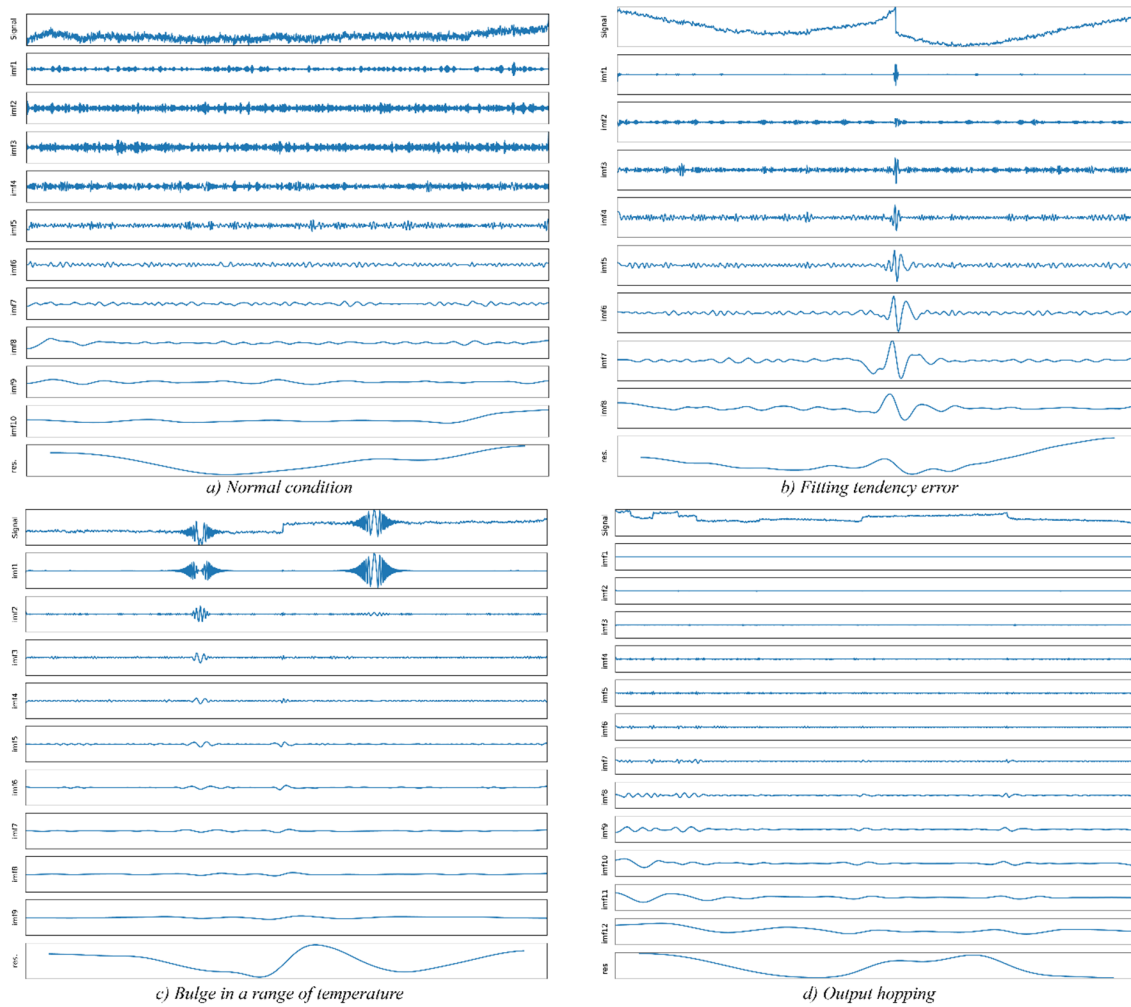


Figure 11. EMD results using proposed BLSTM-based EMD. (a) EMD result in case of normal condition; (b) EMD result in case of fitting error fault occurs; (c) EMD result in case of bulge in a range of temperature fault occurs; (d) EMD result in case of output hopping error fault occurs.

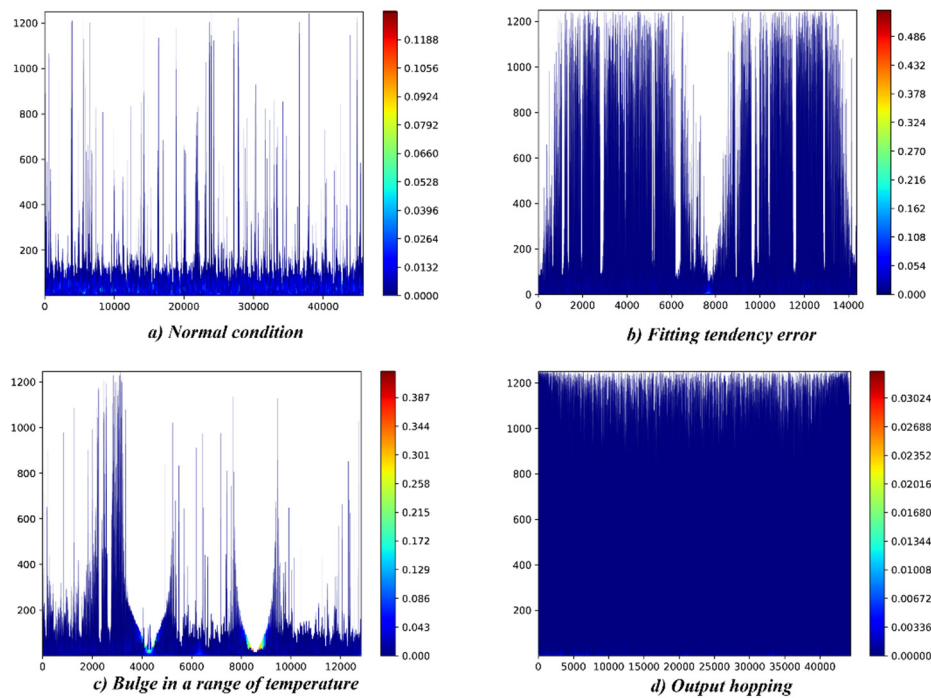


Figure 12. Hilbert spectrum based on EMD results which are shown in Figure 11. (a) Hilbert spectrum in case of normal condition; (b) Hilbert spectrum in case of fitting error fault occurs; (c) Hilbert spectrum in case of bulge in a range of temperature fault occurs; (d) Hilbert spectrum in case of output hopping error fault occurs.

Table 6 lists the orthogonality index criterion of each fault in the proposed method and the compared methods. For the proposed EMD method, the orthogonality index criterion in each fault has small values, which means that the proposed method can decompose various frequency components orthogonally. The orthogonality index values of the proposed method and the compared methods are approximate values because our dataset for training BLSTM is generated by the algorithm which is improved from current methods.

Table 6. Orthogonality index criterion.

Method	Fault0	Fault1	Fault2	Fault3
Proposed EMD	5.0601×10^{-5}	1.0698×10^{-4}	2.6451×10^{-5}	2.8746×10^{-4}
Traditional EMD	4.8373×10^{-4}	6.8474×10^{-3}	4.8372×10^{-4}	3.3746×10^{-5}
Bagherzadeh et al. [48]	8.3865×10^{-5}	3.9473×10^{-4}	8.4937×10^{-4}	1.3836×10^{-3}
Zheng et al. [49]	4.8362×10^{-5}	7.8924×10^{-5}	9.3837×10^{-4}	8.3972×10^{-4}
Mokhtari et al. [50]	4.9732×10^{-4}	5.1198×10^{-4}	4.9471×10^{-5}	3.7765×10^{-4}

4.3.2. Comparison of EMD Efficiency

We compared the EMD efficiency between our proposed EMD method and other EMD methods in terms of the time required for performing an EMD operation. To perform this comparison, we chose various inertial measurement unit (IMU) data with different lengths, which were 11,996, 16,679, 20,836, 25,726, 29,899, 33,002, 37,086, 42,429, 46,861, and 50,854. Results are shown in Figure 13, Our proposed method had the lowest time consumption, with a mean of 7.6475 s for each EMD. The EMD method in 45 had the longest time, 23.4739 s, for performing each EMD, whereas the traditional EMD took 12.0634 s to perform each EMD. The EMD methods in 46 and 47 took approximately the same amount of time as the traditional EMD, 13.2865 s and 14.1303 s respectively, to perform each EMD.

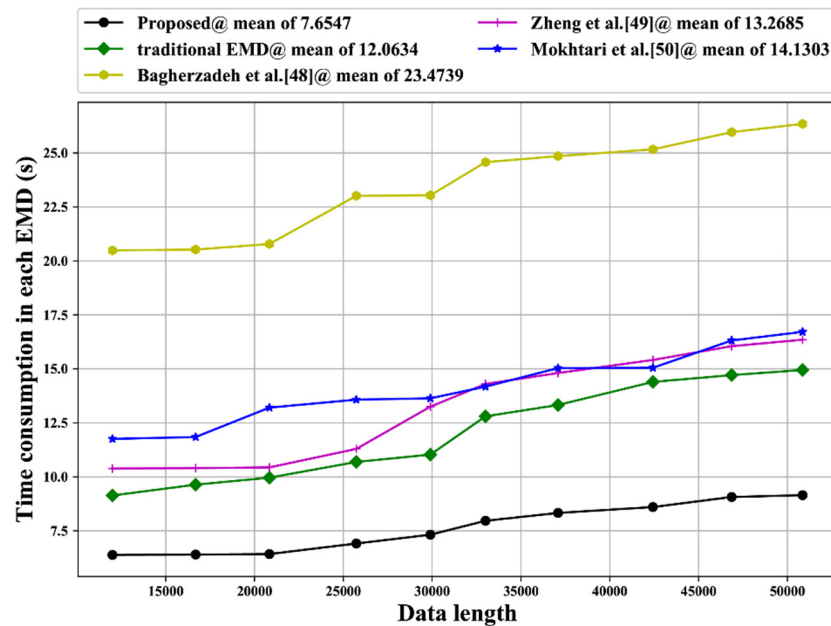


Figure 13. Time consumption comparison.

As in our proposed method EMD directly operates with BLSTM, our algorithm does not need various inner loop iterations. The largest time consumption of an EMD method in [48] is due to the time required for the genetic algorithm to perform the evolution computing. The similarity in time consumption among the EMD methods of [49,50] and traditional EMD is due to these three methods having the same sifting manner in EMD operation.

4.4. Multiscale CNN Comparison

We tested the classification accuracy of the proposed multiscale CNN in the case of different fully connecting (FC) layer configuration. The results are shown in Figure 14. We first tested the one-layer FC performance in a different number of neurons to find the best one-layer FC configuration, then added an FC layer based on the optimal one-layer FC structure and found the two-layer configuration structure with the best classifying performance. The classification accuracy was evaluated according to a 10-times mean accuracy test. The metrics we chose were minimum, maximum, mean, and standard deviation. The adam optimizer and cross-entropy were used to train the CNN. The training epochs were set to 200, and the batch size was 16.

Figure 14 shows the mean classification accuracy for different FC configurations. Note that “FC-*i*” indicates the one-layer FC configuration and “FC-*i-j*” indicates the two-layer FC configuration, and that “*i*” indexes the neurons in the first FC layer and “*j*” indexes the neurons in the second FC layer. For the one-layer CNN, the FC-512 has the highest mean accuracy of 94.9992%, with a standard deviation of 0.3142. Then we added one FC layer to the network “FC-512.” We found that the network “FC-512-128” obtained the best mean classification accuracy of 97.1571%, with a standard deviation of 0.3218. Therefore, the multiscale CNN, FC-512-128, was adopted for FD operation.

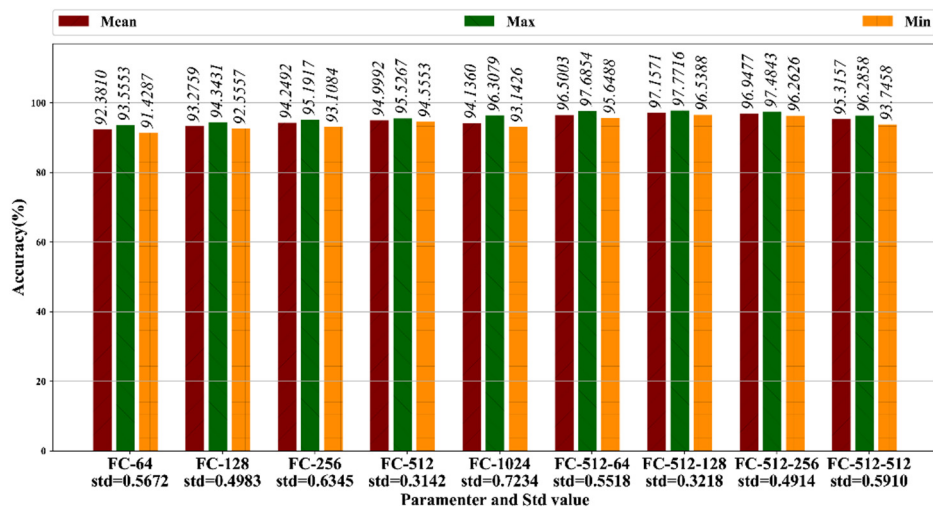


Figure 14. Accuracy test results (in percent).

4.5. FD Performance Comparison

We compared our proposed FD method with five state-of-the-art data-driven FD methods: Kordestani et al.'s ANN-based method [21], Baskaya et al.'s SVM-based method [31], Guo et al.'s CNN-based method [20], Yang et al.'s HHT-CNN-based method [43] and Wen et al.'s CNN-based method [38]. We set training epochs to 300 and set the batch size to 16. The cross-entropy and the Adam optimizer were used to train the neural networks. We performed the comparisons on mean accuracy and confusion matrix to evaluate the accuracy and the misclassified performance.

4.5.1. Comparison on Mean Accuracy

We compared the mean fault classification accuracy by performing our inertial dataset on each method. The results, as shown in Figure 15, indicate the mean, minimum and maximum classification accuracy after being run 10 times. The results show that our method achieved the best fault classification accuracy, at 97.3007%. The CNN-based methods, which are proposed by Guo et al. [20], Yang et al. [43] and Wen et al. [38] achieved mean FD accuracies of 94.6702%, 95.3258%, and 95.8644%, respectively. The machine learning-based methods, which are Kordestani et al.'s ANN-based method [21] and Baskaya et al.'s SVM-based method [31] achieved FD accuracies of 90.1147% and 91.1357%, respectively.

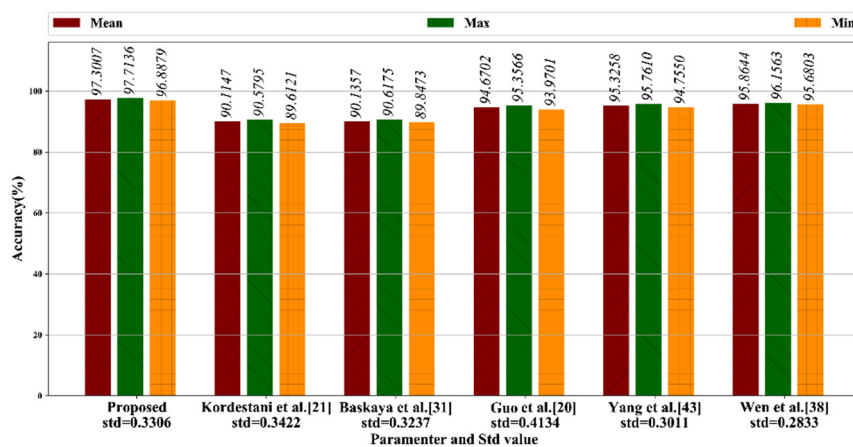


Figure 15. Fault classification accuracy comparison (in percent).

4.5.2. Comparison on Confusion Matrix

We further compared the misclassification performance by using confusion matrix analysis. Figure 16 shows the FD accuracy confusion matrix of each FD method. Our proposed method obtained the best accuracy, at 97.2434%. The fault 1 and fault 3 achieved lower accuracies, respectively, whereas fault 0 had the worst accuracy, and fault 2 achieved the best accuracy. Kordestani et al. [21] had the worst total accuracy of 90.0904% with the largest misclassification. Similar to the mean accuracy results, the CNN-based methods, which are proposed in 20, 43, and 38, have smaller misclassifying results than the machine learning-based FD methods proposed in [21,31].

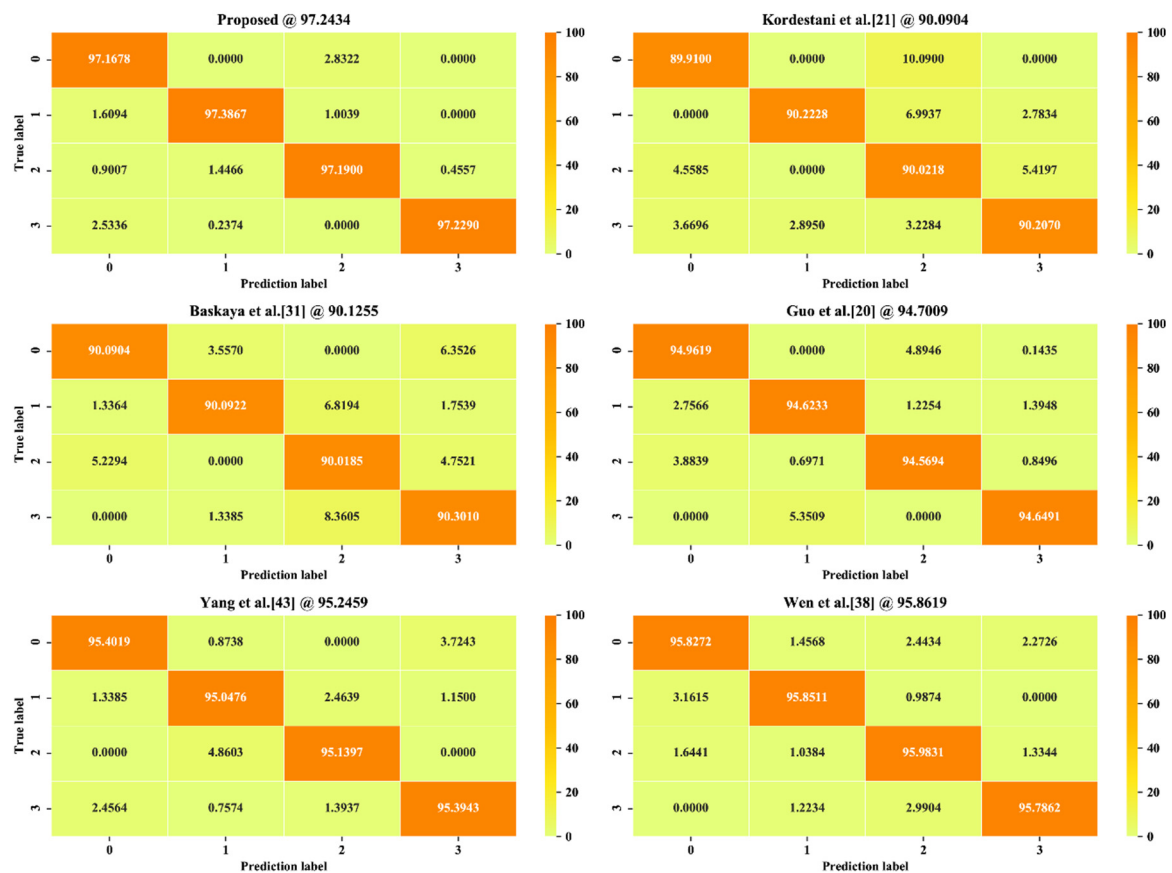


Figure 16. Confusion matrix comparison (in percent).

The results show that the CNN-based methods performed better than the machine learning-based methods, benefited by the advantages of the DL algorithm in handling complex signals. The HHT- and CNN-based methods, which are the proposed methods and Yang et al.'s method [43], performed better than the time domain sliding window-based method [38] and the shot time Fourier transform-based method [20], proving that HHT has obvious advantages in handling nonstationary and non-linear signals. The best performance was achieved by our proposed FD method, which benefits from the proposed HHT-based feature extraction method and the multiscale CNN. The proposed HHT-based feature extraction method has advantages in mining nonstationary features because the use of HHT and the combination of frequency shifting and EEMD improves the EMD performance by resolving the mode mixing problem. The multiscale CNN combines the shallow features and deep features to feed more detailed information to classify the faults, which increases the FD classification performance.

5. Conclusions

In this paper, we propose a method for fault diagnosis of MEMS inertial sensors by combining BLSTM-based HHT and CNN. The inertial feature extraction is performed by our proposed BLSTM-based HHT, which offers high EMD efficiency through the use of a priori knowledge of MEMS inertial data. A training dataset generation algorithm for training the BLSTM model of the proposed BLSTM-based HHT is proposed. The task of fault classification is performed by the proposed MS-CNN. The experiment results show that the proposed BLSTM-based EMD has excellent computation efficiency and achieved satisfactory decomposing performance and that the proposed CNN can precisely classify the fault patterns.

Comparing with the relevant FD methods, the proposed FD method has following strengths: (1) the HHT-based feature extraction algorithm offers advantages in processing variable-temperature fault features. (2) the directly HHT-based feature extraction algorithm by combining BLSTM and HHT offers the obvious time consumption advantages. (3) CNN offers advantages in fault features mining. Note that the preferable feature coding strategy of BLSTM-based EMD is not considered in our paper, that makes BLSTM-based EMD operation also needs additional time frequency shifting and EEMD to improve its performance. In the future, an improved feature coding method that contains the state of the EEMD and frequency shifting can be studied for integrating the additional operation into the BLSTM-based regression model in order to further improve the efficiency of EMD.

Author Contributions: The contributions of authors are as follows: study design, T.G. and W.S.; data collection, T.G.; data analysis, T.G.; writing—original draft preparation, T.G. and M.Z.; writing—review and editing, T.G. and M.Z.; literature search, T.G., F.L. and J.L.; figures, F.L. and J.L.; final approval, W.S., M.Z. and B.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of China under Grant 61274117, in part by the Start Up Project of Chongqing University under Grant 02160011044118, in part by the Natural Science Foundation of China under Grant 61876026, in part by University level scientific research project of Qiannan Normal University for Nationalities under Grant No.qnsy2018006, in part by the Discipline construction subproject of computer and Information College of Qiannan Normal University for Nationalities under Grant CST_2019SN02, in part by Qiannan science and technology project under Grant Zi(2018)No.7, in part by Education quality improvement project of Qiannan Normal University for Nationalities under Grant 2018xjg0524, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ20F020006, and in part by the Scientific Research Fund of Zhejiang Provincial Education Department under Grant Y201941813, in part by the General Program of National Natural Science Foundation of Chongqing under Grant cstc2020jcyj-msxmX0790, in part by the Fundamental Research Funds for the Central Universities under Grant 2020CDJ-LHZZ-052, and in part by the Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS201906.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rossi, M.; Brunelli, D. Autonomous Gas Detection and Mapping with Unmanned Aerial Vehicles. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 765–775. [[CrossRef](#)]
2. Manfreda, S.; McCabe, M.F.; Miller, P.E.; Lucas, R.; Pajuelo Madrigal, V.; Mallinis, G.; Ben Dor, E.; Helman, D.; Estes, L.; Ciraolo, G.; et al. On the use of unmanned aerial systems for environmental monitoring. *Remote Sens.* **2018**, *10*, 641. [[CrossRef](#)]
3. Adão, T.; Hruška, J.; Pádua, L.; Bessa, J.; Peres, E.; Morais, R.; Sousa, J.J. Hyperspectral Imaging: A Review on UAV-Based Sensors, Data Processing and Applications for Agriculture and Forestry. *Remote Sens.* **2017**, *9*, 1110. [[CrossRef](#)]
4. Yang, H.; Zhou, B.; Wang, L.; Xing, H.; Zhang, R. Lixin A Novel Tri-Axial MEMS Gyroscope Calibration Method over a Full Temperature Range. *Sensors* **2018**, *18*, 3004. [[CrossRef](#)] [[PubMed](#)]
5. Zhang, B.; Chu, H.; Sun, T.; Guo, L. Thermal calibration of a tri-axial MEMS gyroscope based on Parameter-Interpolation method. *Sens. Actuators A Phys.* **2017**, *261*, 103–116. [[CrossRef](#)]
6. Araghi, G. Temperature compensation model of MEMS inertial sensors based on neural network. In Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 23–26 April 2018; pp. 301–309.

7. Mistry, P.; Lane, P.; Allen, P. Railway Point-Operating Machine Fault Detection Using Unlabeled Signaling Sensor Data. *Sensors* **2020**, *20*, 2692. [[CrossRef](#)]
8. Luong, T.T.N.; Kim, J.M. The Enhancement of Leak Detection Performance for Water Pipelines through the Renovation of Training Data. *Sensors* **2020**, *20*, 2542. [[CrossRef](#)]
9. Dong, L.I.; Shulin LI, U.; Zhang, H. A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples. *Pattern Recognit.* **2017**, *64*, 374–385. [[CrossRef](#)]
10. Kim, S.Y.; Kang, C.H.; Song, J.W. 1-point RANSAC UKF with Inverse Covariance Intersection for Fault Tolerance. *Sensors* **2020**, *20*, 353. [[CrossRef](#)]
11. Qin, J.; Zhang, G.; Zheng, W.X.; Kang, Y. Neural Network-Based Adaptive Consensus Control for a Class of Nonaffine Nonlinear Multiagent Systems with Actuator Faults. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3633–3644. [[CrossRef](#)]
12. Pan, J.; Luo, D.; Wu, L.; Zhang, J. FlexRay based treble-redundancy UAV flight control computer system. In Proceedings of the IEEE International Conference on Control & Automation, Ohrid, Macedonia, 3–6 July 2017.
13. Pan, J.H.; Zhang, S.B.; Ma, H. Design of Flight Control System Bus Controller of UAV Based on Double CAN-Bus. In *Applied Mechanics and Materials*; Trans Tech Publications Ltd.: Baech, Switzerland, 2014; Volume 479, pp. 641–645.
14. Najjar, N.; Gupta, S.; Hare, J.; Kandil, S.; Walthall, R. Optimal sensor selection and fusion for heat exchanger fouling diagnosis in aerospace systems. *IEEE Sens. J.* **2016**, *16*, 4866–4881. [[CrossRef](#)]
15. Deckert, J.C.; Desai, M.; Deyst, J.; Willsky, A. F-8 DFBW sensor failure identification using analytic redundancy. *IEEE Trans. Autom. Control* **1977**, *22*, 795–803. [[CrossRef](#)]
16. Chi, C.; Deng, P.; Zhang, J.; Pan, Z.; Li, T.; Wu, Z. A Fault Diagnosis Method of Temperature Sensor Based on Analytical Redundancy. In Proceedings of the 2019 Prognostics and System Health Management Conference (PHM-Paris), Paris, France, 2–5 May 2019; pp. 156–162.
17. Lyu, P.; Lai, J.; Liu, J.; Liu, H.H.; Zhang, Q. A thrust model aided fault diagnosis method for the altitude estimation of a quadrotor. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 1008–1019. [[CrossRef](#)]
18. Guo, D.; Zhong, M.; Zhou, D. Multisensor data-fusion-based approach to airspeed measurement fault detection for unmanned aerial vehicles. *IEEE Trans. Instrum. Meas.* **2017**, *67*, 317–327. [[CrossRef](#)]
19. Zhang, Z.H.; Yang, G. Distributed fault detection and isolation for multiagent systems: An interval observer approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *50*, 2220–2230. [[CrossRef](#)]
20. Guo, D.; Zhong, M.; Ji, H.; Liu, Y.; Yang, R. A Hybrid Feature Model and Deep Learning Based Fault Diagnosis for Unmanned Aerial Vehicle Sensors. *Neurocomputing* **2018**, *319*, 155–163. [[CrossRef](#)]
21. Kordestani, M.; Samadi, M.F.; Saif, M.; Khorasani, K. A New Fault Diagnosis of Multifunctional Spoiler System Using Integrated Artificial Neural Network and Discrete Wavelet Transform methods. *IEEE Sens. J.* **2018**, *18*, 4990–5001. [[CrossRef](#)]
22. Zhu, T.B.; Lu, F. A Data-Driven Method of Engine Sensor on Line Fault Diagnosis and Recovery. *Appl. Mech. Mater.* **2014**, *490*, 1657–1660. [[CrossRef](#)]
23. Cheng, H.; Zhang, Y.; Lu, W.; Yang, Z. A bearing fault diagnosis method based on VMD-SVD and Fuzzy clustering. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 1950018. [[CrossRef](#)]
24. Liu, P.; Zhang, W. A Fault Diagnosis Intelligent Algorithm Based on Improved BP Neural Network. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 1959028. [[CrossRef](#)]
25. Chen, B.; Huang, D.; Zhang, F. The Modeling Method of a Vibrating Screen Efficiency Prediction Based on KPCA and LS-SVM. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 1950009. [[CrossRef](#)]
26. Naderi, E.; Khorasani, K. Data-driven fault detection, isolation and estimation of aircraft gas turbine engine actuator and sensors. *Mech. Syst. Signal Process.* **2018**, *100*, 415–438. [[CrossRef](#)]
27. Fravolini, M.L.; Napolitano, M.R.; Del Core, G.; Papa, U. Experimental interval models for the robust fault detection of aircraft air data sensors. *Control Eng. Pract.* **2018**, *78*, 196–212. [[CrossRef](#)]
28. Xuyun, F.U.; Chen, H.; Zhang, G.; Tao, T. A New Point Anomaly Detection Method About Aero Engine Based on Deep Learning. In Proceedings of the 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Xi'an, China, 15–17 August 2018; pp. 176–181.
29. Gao, Z.; Ma, C.; Song, D.; Liu, Y. Deep quantum inspired neural network with application to aircraft fuel system fault diagnosis. *Neurocomputing* **2017**, *238*, 13–23. [[CrossRef](#)]

30. He, Y.; Peng, Y.; Wang, S.; Liu, D.; Leong, P.H. A structured sparse subspace learning algorithm for anomaly detection in UAV flight data. *IEEE Trans. Instrum. Meas.* **2017**, *67*, 90–100. [[CrossRef](#)]
31. Baskaya, E.; Bronz, M.; Delahaye, D. Fault detection & diagnosis for small uavs via machine learning. In Proceedings of the 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), St. Petersburg, FL, USA, 17–21 September 2017; pp. 1–6.
32. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)]
33. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)]
34. Yao, Y.; Zhang, S.; Yang, S.; Gui, G. Learning attention representation with a multi-scale CNN for gear fault diagnosis under different working conditions. *Sensors* **2020**, *20*, 1233. [[CrossRef](#)]
35. Guo, X.; Chen, L.; Shen, C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement* **2016**, *93*, 490–502. [[CrossRef](#)]
36. Wang, Z.; Dong, Y.; Liu, W.; Ma, Z. A Novel Fault Diagnosis Approach for Chillers Based on 1-D Convolutional Neural Network and Gated Recurrent Unit. *Sensors* **2020**, *20*, 2458. [[CrossRef](#)]
37. Chen, S.; Ge, H.; Li, J.; Pecht, M. Progressive Improved Convolutional Neural Network for Avionics Fault Diagnosis. *IEEE Access* **2019**, *7*, 177362–177375. [[CrossRef](#)]
38. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5990–5998. [[CrossRef](#)]
39. Chong, U.P. Signal model-based fault detection and diagnosis for induction motors using features of vibration signal in two-dimension domain. *Stroj. Vestn.* **2011**, *57*, 655–666.
40. Zhong, D.; Guo, W.; He, D. An Intelligent Fault Diagnosis Method based on STFT and Convolutional Neural Network for Bearings Under Variable Working Conditions. In Proceedings of the 2019 Prognostics and System Health Management Conference (PHM-Qingdao), Qingdao, China, 25–27 October 2019.
41. Chu, W.; Lin, C.; Kao, K. Fault Diagnosis of a Rotor and Ball-Bearing System Using DWT Integrated with SVM, GRNN, and Visual Dot Patterns. *Sensors* **2019**, *19*, 4806. [[CrossRef](#)]
42. Cabrera, D.; Sancho, F.; Li, C.; Cerrada, M.; Sánchez, R.V.; Pacheco, F.; de Oliveira, J.V. Automatic Feature Extraction of Time-Series applied to Fault Severity Assessment of Helical Gearbox in Stationary and Non-Stationary Speed Operation. *Appl. Soft Comput.* **2017**, *58*, 53–64. [[CrossRef](#)]
43. Yang, Q.; Ruan, J.; Zhuang, Z.; Huang, D. Condition Evaluation for Opening Damper of Spring-Operated High-Voltage Circuit Breaker Using Vibration Time-Frequency Image. *IEEE Sens. J.* **2019**, *19*, 8116–8126. [[CrossRef](#)]
44. Lan, S.; Chen, M.J.; Chen, D.Y. A Novel HVDC Double-Terminal Non-Synchronous Fault Location Method Based on Convolutional Neural Network. *IEEE Trans. Power Deliv.* **2019**, *34*, 848–857. [[CrossRef](#)]
45. Tabian, I.; Fu, H.; Sharif Khodaei, Z. A convolutional neural network for impact detection and characterization of complex composite structures. *Sensors* **2019**, *19*, 4933. [[CrossRef](#)]
46. De Oliveira, M.A.; Monteiro, A.V.; Vieira Filho, J. A new structural health monitoring strategy based on PZT sensors and convolutional neural network. *Sensors* **2018**, *18*, 2955. [[CrossRef](#)]
47. Pham, H.C.; Ta, Q.B.; Kim, J.T.; Ho, D.D.; Tran, X.L.; Huynh, T.C. Bolt-Loosening Monitoring Framework Using an Image-Based Deep Learning and Graphical Model. *Sensors* **2020**, *20*, 3382. [[CrossRef](#)]
48. Bagherzadeh, S.A.; Asadi, D. Detection of the ice assertion on aircraft using empirical mode decomposition enhanced by multi-objective optimization. *Mech. Syst. Signal Process.* **2017**, *88*, 9–24. [[CrossRef](#)]
49. Zheng, H.; Liu, J.; Duan, S. Flutter test data processing based on improved Hilbert-Huang transform. *Math. Probl. Eng.* **2018**, *2018*, 3496870. [[CrossRef](#)]
50. Mokhtari, S.A.; Sabzehparvar, M. Application of Hilbert–Huang Transform with Improved Ensemble Empirical Mode Decomposition in Nonlinear Flight Dynamic Mode Characteristics Estimation. *J. Comput. Nonlinear Dyn.* **2019**, *14*, 011006. [[CrossRef](#)]
51. Guo, M.F.; Yang, N.C.; Chen, W.F. Deep-Learning-Based Fault Classification Using Hilbert–Huang Transform and Convolutional Neural Network in Power Distribution Systems. *IEEE Sens. J.* **2019**, *19*, 6905–6913. [[CrossRef](#)]
52. Han, B.; Yang, X.; Ren, Y.; Lan, W. Comparisons of different deep learning-based methods on fault diagnosis for geared system. *Int. J. Distrib. Sens. Netw.* **2019**, *15*. [[CrossRef](#)]
53. Xie, Y.; Xiao, Y.; Liu, X.; Liu, G.; Jiang, W.; Qin, J. Time-Frequency Distribution Map-Based Convolutional Neural Network (CNN) Model for Underwater Pipeline Leakage Detection Using Acoustic Signals. *Sensors* **2020**, *20*, 5040. [[CrossRef](#)]

54. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [[CrossRef](#)]
55. Jiang, J.R.; Lee, J.E.; Zeng, Y.M. Time Series Multiple Channel Convolutional Neural Network with Attention-Based Long Short-Term Memory for Predicting Bearing Remaining Useful Life. *Sensors* **2019**, *20*, 166. [[CrossRef](#)]
56. Liu, G.; Gu, H.; Shen, X.; You, D. Bayesian Long Short-Term Memory Model for Fault Early Warning of Nuclear Power Turbine. *IEEE Access* **2020**, *8*, 50801–50813. [[CrossRef](#)]
57. Yin, A.; Yan, Y.; Zhang, Z.; Li, C.; Sánchez, R.V. Fault Diagnosis of Wind Turbine Gearbox Based on the Optimized LSTM Neural Network with Cosine Loss. *Sensors* **2020**, *20*, 2339. [[CrossRef](#)]
58. Hoseinzadeh, M.S.; Khadem, S.E.; Sadooghi, M.S. Modifying the Hilbert-Huang transform using the nonlinear entropy-based features for early fault detection of ball bearings. *Appl. Acoust.* **2019**, *150*, 313–324. [[CrossRef](#)]
59. Liang, K.; Qin, N.; Huang, D.; Fu, Y. Convolutional recurrent neural network for fault diagnosis of high-speed train bogie. *Complexity* **2018**, *2018*, 4501952. [[CrossRef](#)]
60. Peng, Y.; Chen, J.; Liu, Y.; Cheng, J.; Yang, Y.; Kuanfang, H.; Wang, G.; Liu, Y. Roller Bearing Fault Diagnosis Based on Adaptive Sparsest Narrow-Band Decomposition and MMC-FCH. *Shock Vib.* **2019**, *2019*, 7585401. [[CrossRef](#)]
61. Huang, C.G.; Huang, H.Z.; Li, Y.F. A Bi-Directional LSTM prognostics method under multiple operational conditions. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8792–8802. [[CrossRef](#)]
62. Yang, J.; Guo, Y.; Zhao, W. Long short-term memory neural network based fault detection and isolation for electro-mechanical actuators. *Neurocomputing* **2019**, *360*, 85–96. [[CrossRef](#)]
63. Miao, H.; Li, B.; Sun, C.; Liu, J. Joint Learning of Degradation Assessment and RUL Prediction for Aeroengines via Dual-Task Deep LSTM Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5023–5032. [[CrossRef](#)]
64. Damaševičius, R.; Napoli, C.; Sidekerskienė, T.; Woźniak, M. IMF mode demixing in EMD for jitter analysis. *J. Comput. Sci.* **2017**, *22*, 240–252. [[CrossRef](#)]
65. Wu, Z.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41. [[CrossRef](#)]
66. Gu, H.; Liu, X.; Zhao, B.; Zhou, H. The In-Operation Drift Compensation of MEMS Gyroscope Based on Bagging-ELM and Improved CEEMDAN. *IEEE Sens. J.* **2019**, *19*, 5070–5077. [[CrossRef](#)]
67. Hosseini-Pishrobat, M.; Keighobadi, J. Robust Vibration Control and Angular Velocity Estimation of a Single-Axis MEMS Gyroscope Using Perturbation Compensation. *J. Intell. Robot. Syst.* **2019**, *94*, 61–79. [[CrossRef](#)]

