sustainability

MDPI

Article

Industrial Big Data and Computational Sustainability: Multi-Method Comparison Driven by High-Dimensional Data for Improving Reliability and Sustainability of Complex Systems

Chunting Liu and Guozhu Jia *

School of Economics and Management, Beihang University, Beijing 100191, China
* Correspondence: jiaguozhu@buaa.edu.cn

Received: 20 July 2019; Accepted: 19 August 2019; Published: 22 August 2019

check for updates

Abstract: Sustainable development is of great significance. The emerging research on data-driven computational sustainability has become an effective way to solve this problem. This paper presents a fault diagnosis and prediction framework for complex systems based on multi-dimensional data and multi-method comparison, aimed at improving the reliability and sustainability of the system by selecting methods with relatively superior performance. This study took the avionics system in the industrial field as an example. Based on the literature research on typical fault modes and fault diagnosis requirements of avionics systems, three popular high-dimensional data-driven fault diagnosis methods—support vector machine, convolutional neural network, and long- and short-term memory neural network—were comprehensively analyzed and compared. Finally, the actual bearing failure data were used for programming in order to verify and compare various methods and the process of selecting the superior method driven by high-dimensional data was fully demonstrated. We attempt to provide a sustainable development idea that continuously explores multi-method integration and comparison, aimed at improving the calculation efficiency and accuracy of reliability assessments, optimizing system performance, and ultimately achieving the goal of long-term improvement of system reliability and sustainability.

Keywords: industrial big data; computational sustainability; multi-method comparison; reliability and sustainability; high-dimensional data

1. Introduction

Sustainable development has become the focus of the international community. However, how to implement technology-driven sustainability from the existing policy-driven approach is the bottleneck of realizing sustainability. In recent years, the emerging research on computational sustainability has become an effective way to solve this problem and thus a new research hotspot. The focus of computational sustainability research is to develop computational models, mathematical models, and related methods to help solve some of the most challenging problems related to sustainable development [1–3]. The advent of the era of big data brings opportunities for computational sustainability research, as well as new challenges such as complexity of problems, computational efficiency, and scalability of methods. Furthermore, the use of computer and information science technology can improve the necessity and effectiveness of resource management and allocation. Big data contains abundant information and potential knowledge, which provides a new research method driven by data, especially multi-source data. It will greatly improve the accuracy of the method and the efficiency of problem solving.

The development of massive multi-dimensional data and computational sustainability is critical for meeting the challenges of sustainability. It can help people make tradeoffs, familiarize themselves with complex systems, and explain uncertainties. At present, more and more scholars have started to use data mining technology to solve the problem of computational sustainability. Representational research involves many fields such as banking [4,5], healthcare [6,7], meteorology [8–11], ecological protection [12–14], agriculture [15,16], and disaster management [17,18]. However, the applications in industry, especially from a system perspective, are relatively rare. In fact, the reliability and sustainability of complex systems in industry, especially complex systems with high-end equipment, is a matter of concern, because it relates to efficiency, cost, resources, energy consumption, human–computer interaction, and many other aspects. The reliability and sustainability of complex systems are also closely related to each other. Continuous improvement of the reliability level is of great benefit to long-term sustainability. A complex system with good sustainability will greatly improve operational efficiency and re-source utilization.

The reliability and sustainability of high-end equipment systems determine the efficiency and costs of equipment operation and affect their products and service quality. Reliability and sustainability assessment of high-end equipment is a complex task, and there is a need to continuously improve the accuracy of this task. This study took the problem of fault diagnosis and reliability evaluation of avionics systems as an example to illustrate the application of big data and computational sustainability in industry. As a key component structure of modern aircraft, its intelligence and synthesis are constantly enhanced with the rapid development of electronic information technology, computer science, large aircraft, and unmanned aerial vehicle industry. The avionics systems are a complex system. There exists a correlation between failures, and fault data present high-dimensional attributes. Studying the degradation mode of the avionics system, especially the fault classification and diagnosis prediction driven by high-dimensional data, is of great significance for improving the reliability and sustainability of the system.

Massive multi-dimensional data, computational sustainability and multi-method comparisons are undervalued in terms of improving the reliability and sustainability of industrial complex systems. In this paper, a fault diagnosis and prediction framework for complex systems based on multi-dimensional data and multi-method comparison is proposed. We comprehensively analyzed and compared the popular fault diagnosis methods driven by high-dimensional data and analyzed the performance of each method. According to the characteristics of systems and data sets, methods with relatively superior performance were selected. To improve the reliability and sustainability of complex systems, it is necessary to continuously evaluate and select appropriate data-driven methods. On the one hand, it can improve the accuracy and efficiency of evaluation; on the other hand, it can save time, resources, and costs for the entire system. On the issue of improving the reliability and sustainability of industrial complex systems, we expect to propose a new approach that matches the characteristics of industrial complex systems and performs better than the traditional and isolated methods by using industrial big data, machine learning, and multi-method comparisons, rather than chasing hot spots and trends out of touch with reality.

2. Fault Diagnosis and Reliability Evaluation of Complex System

With the development of science and technology and information technology, the precision and complexity of high-end equipment are constantly improving, and the reliability level of complex systems is also required to be higher. Compared to traditional mechanical components, the avionics system has not only a complex hierarchical structure but also numerous software and hardware devices. Moreover, the reliability of the avionics system is also interfered by various external environmental factors such as external temperature, continuous propulsion, electromagnetic interference, pollution, and vibration. Therefore, the faults are not isolated, but rather interrelated to each other. Moreover, they are affected by each other because of the complex hierarchical structure. There are many ways to evaluate the reliability of the avionics system, such as analysis of the mechanism characteristics of massive and high-dimensional fault data in the big data environment, analysis of the modeling of correlation between faults, analysis of multi-fault mode modeling, evaluation of comprehensive reliability, and demonstration platform of fault diagnosis.

In recent years, aircraft fault informatics has made great progress because of a rapid acceleration in the level of computer, modern monitoring technology, and signal processing technology, thus improving the diagnosis and positioning levels of aircraft faults. For example, large civil aircraft (such as A320, A340, and BoeNe777) use a central maintenance computer system to monitor the flight status of aircraft without intervals, record faults, and problems during the flight process, and send alarm signals to the computer [19]. For multi-fault diagnosis, great achievements have been made in recent years. Wang et al. studied the relationship between the factors causing problems in aircraft electronic equipment and the external environment [20]. Chunfei et al. established an engine fault diagnosis method using the neural network to study a turbofan engine, which can obviously remove the influence of noise in the signal, thus helping in obtaining correct analysis results of engine performance [21]. Furthermore, from the perspective of probability statistics, fault diagnosis of aircraft communication equipment was carried out by Petrov and Marinov [22]. Combining rule-based reasoning with fuzzy modeling, Yang proposed an overall intelligent fault diagnosis method for avionics systems [23]. Using fuzzy modeling, Zhou and Cao realized fault classification diagnosis [24]. Jia et al. considered aircraft as engineering systems and concluded that they had different energies under different states (failure or normal) [25]. Then, a multi-fault diagnosis model was established according to the law of conservation of energy. Multi-fault classification diagnosis has been implemented with several different neural network models, such as back propagation (BP) and self-organizing maps (SOM), by Fan [26]. Using the theory of rough neural network, Du proposed a new fault diagnosis method for avionics systems, but with higher reliability [27]. Similarly, Xu put forward a new fault classification diagnosis theory for avionics systems on the basis of the Lyapunov index spectrum and Lyapunov index spectral entropy [28]. Moreover, Shi proposed a rule-based fault diagnosis method [29]. From the perspective of the physical model, Baig proposed a multi-fault diagnosis expert system [30]. Palmer suggested that data/information collection and processing as well as database technology played an important role in aircraft fault diagnosis [31]. Kordestani proposed an integrated system to classify fault localization [32]. Likewise, a multi-fault diagnosis expert system was proposed by Anami, using a dynamic fault tree method with continuous optimization [33]. Multi-fault diagnosis methods for the aircraft mentioned above can usually meet traditional requirements. However, the definition of fault mechanism and transmission route of the system is not comprehensive, so faults with high complexity and multiple sources of avionics systems cannot be comprehensively and effectively diagnosed. At the same time, the above methods can only diagnose faults, and they lack reliable evaluation of the system. There are many research directions in the classical system reliability theory, including the modeling and analysis of failure modes, system reliability, reliable physical model, reliability design, maintainability, maintainability design, distribution pattern recognition of failure and maintenance, reliability evaluation and application, and reliability and maintenance implementation [34–37]. Existing evaluation for system reliability is usually completed by establishing a reliability model, for which the modeling and analysis of failure modes of the system should be done first, such as failure distribution model, constant failure rate model, and failure model related to time. No matter from external forms or internal logical mechanisms, reliability evaluation of the avionics system is completely different from the traditional single part (equipment) or the mechanical manufacture system composed of a set of parts (equipment). It is impossible to use the existing mathematical statistical model of reliability and the highly simplified and abstract system analysis method. The existing analysis and evaluation methods for system reliability include the reliability block diagram method, failure tree method, Markov model, Petri net method, and goal-oriented (GO) method. However, in evaluating the avionics system of an unmanned aerial vehicle (UAV), the disadvantages of these methods are as follows:

1. Most of the methods can only be used for static modeling, and they do not reflect the relatively complex dynamic temporal relationship of the system [22–24,26].
2. As the system states increase, there would appear an exponential explosion of state space for the reliability analysis model, resulting in great difficulty in model construction and solution [21,23,24,26,27,29,30,32,33].

3. With a larger system scale, it is difficult to establish a model for hierarchical requirements of the system [21,23–27,29–31,33].
4. It is difficult for the existing reliability analysis model of the system to describe system uncertainty as well as for some unconventional influence factors (such as environment and human factors) to be described in the model [20,21,24–28,30,32,33].

In a word, there are many methods for fault diagnosis and prediction of complex systems, and each method is not perfect. Considering the need to continuously improve the reliability and sustainability of systems, it is necessary to deal with such problems with the idea of sustainable development. That is, to continuously explore multi-method integration and comparison, aiming at improving the calculation efficiency and accuracy of reliability assessments, optimizing system performance, and ultimately achieving the goal of long-term improvement of system reliability and sustainability. Three popular high-dimensional data-driven fault diagnosis methods—support vector machine (SVM), convolutional neural network (CNN), and long- and short-term memory (LSTM) neural network—were chosen in this study to demonstrate the process of selecting the superior method. The actual bearing failure data, provided by the Case Western Reserve University Bearing Data Center which is located in Cleveland, OH, USA, have been used for programming in order to verify various methods.

3. Chosen Fault Diagnosis Methods

3.1. Support Vector Machine

On the basis of the statistical learning theory, SVMs are becoming increasingly popular in machine learning activities, including classification, regression, and outlier detection. Generally, the common binary classification problems are solved by the basic and traditional SVM at the beginning of this theory.

3.1.1. SVM Theory

The core principle of this SVM is how to transform data from the originally low-dimensional space to the feature space with a higher dimension, in which a so-called optimal hyperplane can be found to maximize the margin between the two classes. The methodology, algorithm, and software of this SVM have been widely applied [38,39]. From the start, SVM is two dimensional and develops from an optimal classification plane problem on the basis of linear separability. Its central idea can be illustrated by the two-dimensional situation shown in Figure 1 [40]. It displays two different kinds of data: Class A (circular) and Class B (pentagonal) points. The SVM tries to place a linear boundary H between the two classes, aiming to maximize the classification interval (margin); namely, the distance between the boundaries of the two classes H1 and H2 (H1, H2, and H being parallel to each other) is maximum. Moreover, the data points on the boundaries of H1 and H2, called support vectors, are used to define the boundaries.

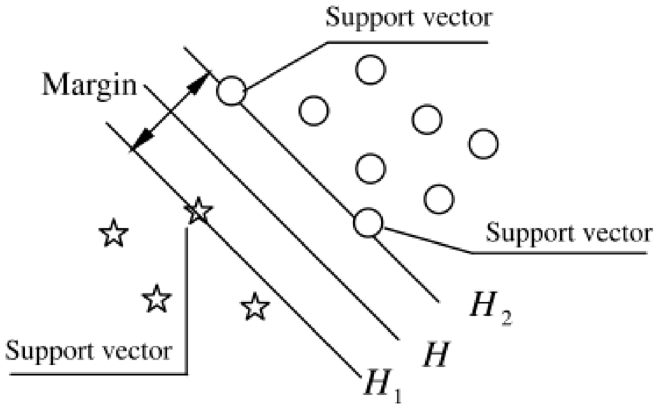


Figure 1. The principle of a support vector machine.

3.1.2. Multiclassification SVM

The SVM was originally designed for binary classification. In general, there was more than one fault condition in addition to the health condition. Therefore, the binary classifier was not suitable for fault diagnosis, which made it necessary to develop a method to deal with multiclassification problems. A brief introduction of the multiclassification SVM is as follows.

Multiclassification can be obtained by combining binary classifications. If there is a classifiable class k (k > 2), it can be separated by binary classification. On the contrary, if any two classes are separable in a class k event, class k must be separable. Therefore, a multiclassifier can be constructed by combining multiple binary classifiers. Several methods have been proposed now, such as one-to-one, one-to-all, and directed acyclic graph multiclass SVM [41,42]. By comparing these methods, Hsu and Lin indicated that the one-to-one method was more suitable for practical application than other methods [35].

For an event or data set containing k categories, the one-to-one method is used to construct M (M = k (k − 1)/2) classifiers, each of which contains two types of data training events or data sets.

For the training data from class *i* to class *j*, the following binary classification problems need to be solved:

$$\min{}_{\omega^{ij},b^{ij},\xi^{ij}} \frac{1}{2}\left(\omega^{ij}\right)^T \omega^{ij} + C\sum_t \xi_n^{ij}\left(\omega^{ij}\right)^T$$

$$\text{s.t.}\begin{cases} \left(\omega^{ij}\right)^T K(x_n) + b^{ij} \geq 1 - \xi_n^{ij} if y_n = i \\ \left(\omega^{ij}\right)^T K(x_n) + b^{ij} \leq -1 + \xi_n^{ij} if y_n = j \\ \xi_n^{ij} \geq 0 \end{cases} \tag{1}$$

Similar to the basic and traditional SVM, $K(x_n)$ represents the classifier of the kernel function; (x_n, y_n) is the *i*th or *j*th training sample, where $\omega \in R^n$ and $b \in R$ are the weight coefficient and intercept parameter; ξ_n^{ij} is the relaxation variable; and C is the parameter for the penalty term. The specific determination methods for the weight coefficients, relaxation variable, and penalty function can be found in literature [35]. After M classifiers are constructed, a variety of methods are applied in the subsequent test stage, usually using the following voting decision method: While determining whether a sample x belongs to class *i* or class *j*, if the classifier determines that the sample belongs to class *i*, the value of the corresponding voting function $V(x_i)$ should be increased by 1. On the contrary, the value of voting function $V\left(x_j\right)$ should be increased by 1. Finally, sample x is categorized as one class in which V is the largest, and this voting decision is also called a "max wins" decision. Please refer to [43] for more detailed introduction to an SVM.

3.2. *Convolutional Neural Network*

Computer vision (CV) is an important direction for in-depth learning practice. At present, CV problems are mainly divided into three categories: (1) image classification, (2) object detection, and (3) neural style transfer. If a traditional neural network is used to deal with the three categories of CV problems, it would be very difficult to solve the problem of the high dimension of the input layer. For a 1000×1000-pixel picture with three color channels, the input layer of the neural network would have a dimension of more than three million, which would result in a sharp increase in the network weight, W, as well as two unexpected consequences. On the one hand, the neural network structure would become complex, which would lead to overfitting as the amount of data is less than that of the network structure, but on the other, the higher requirement for the hardware environment is needed, and the required memory and computation are larger. These can be better solved by CNN.

3.2.1. Structure of CNN

The CNN is also a multistage neural network composed of multiple screening stages and a classification stage. The purpose of the screening stage is to extract features from the convolution layer and pooling layer. The classification stage closely following the screening stage is actually a multilayer perceptron composed of several fully connected layers. To sum up, taking input and output into account, the CNN contains five layers: (1) input layer, (2) convolution layer, (3) pooling layer, (4) full connection layer, and (5) output layer. The function of each type of layer is described as follows: The convolution layer, firstly, uses a filter (the size depends on the dimension of the input data) to convolve the local area of the input layer and then forms a convolution activation unit containing features extracted from the original data. The same convolution kernel (also known as weight sharing) is used by each filter to extract local features of the local area of the input layer. A filter corresponds to a frame in the next layer, and the number of frames is called the depth of the layer. K_i^l and b_i^l represent the weight and deviation of the ith convolution kernel in the lth layer, respectively, and $X^l(j)$ refers to the jth local area of the first layer. The convolution process is described as follows:

$$y_i^{(l\,+\,1)}(j) = \boldsymbol{K}_i^l * \boldsymbol{X}^l(j) + b_i^l \tag{2}$$

where * represents convolution, and it is used to calculate the dot product of the convolution kernel and the local area of the input layer. $y_i^{(l\,+\,1)}(j)$ represents the input of the jth neuron in the ith frame of the (l + 1)th layer. After the convolution operation, the rectified linear units (ReLUs) activation function is introduced to accelerate the convergence of the neural network. The introduction, types, advantages, and disadvantages of this activation function are covered in Section 3.2.2.

In the architecture of CNN, the pooling layer is usually connected with the convolution layer. The pooling layer plays a role in reducing the sampling operation as well as reducing the spatial size of features and network parameters. Furthermore, it is divided into two types: (1) max pooling and (2) average pooling. Max pooling is more frequently used than average pooling. It can conduct a local maximum operation on the input features, reduce the parameters, and obtain the feature of the same position. The transformation of max pooling is described as follows:

$$p_i^{(l\,+\,1)}(j) = \max_{(j\,-\,1)W\,+\,1\,\le\,t\,\le\,jW} \left\{q_i^l(t)\right\} \tag{3}$$

where $q_i^l(t)$ refers to the value of the tth neuron in the ith frame of the first layer, $t \in [(j-1)W+1, jW]$, where W is the width of the pooling region. $p_i^{(l+1)}(j)$ represents the value of the neuron in the (l + 1)th layer of the pooling operation. The CNN model structure used in this study is shown in Figure 2.

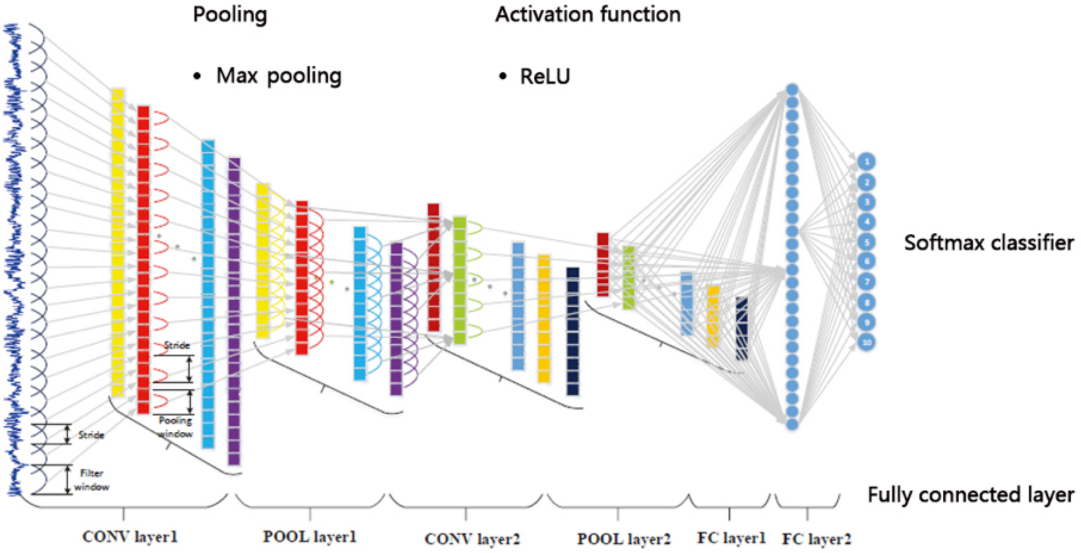


Figure 2. A convolutional neural network model structure. ReLU: rectified linear unit.

The CNN model structure consists of two screening stages and one classification stage. The input of CNN includes the time series signals of bearing fault vibration. The first convolution layer directly extracts features from the original input signals without any other transformation. The features of the input signals are extracted from two convolution layers and two pooling layers. As the number of layers increases, the depth of each layer becomes larger, and the width of each frame becomes smaller, as shown in Figure 2. To complete the classification process, the classification phase consists of two fully connected layers. The Softmax function is used by the classification function, thus making the probabilities of 10 neurons conform to the probability distribution of 10 different bearing health conditions. The form of the Softmax function is as follows:

$$f\left(z_j\right) = \frac{e^{z_j}}{\sum_k^{10} e^{z_k}} \tag{4}$$

where z_j represents the probability of the *j*th output neuron.

Similar to multi-layer perceptron (MLP), the CNN can be trained. The loss function of the CNN model is the cross-entropy between the estimated Softmax output probability distribution and the target class probability distribution. Let $p(x)$ represent the target distribution and $q(x)$ represent the estimated distribution; then, the cross-entropy between $p(x)$ and $q(x)$ is

$$H(p,\ q) = -\sum_x p(x) \cdot \log q(x) \tag{5}$$

In order to minimize the loss function, the Adam stochastic optimization algorithm is used for training the CNN model. It is suitable for big data or a multiparameter model, with the advantages of direct implementation, higher calculation efficiency, small storage space requirement, and so on. For more details about the Adam stochastic optimization algorithm, the reader can refer to [40]. Dropout is applied in the training as one of the effective methods to control overfitting.

3.2.2. Activation Function

At present, there are mainly three frequently used activation functions: (1) sigmoid function, (2) tanh function, and ReLU function. The sigmoid function can precisely control the input to the real value range (0, 1). Particularly, if the input is a very large negative number, then the output is 0. On the contrary, if the input is a very large positive number, then the output is 1. The sigmoid function has been widely used, but fewer people have used it recently. The main reasons are as follows:

1. When the input is slightly away from the origin of coordinates, the gradient of the function becomes smaller—almost zero. During back propagation of neural network, the differential of each weight ω is calculated by the chain rule of differential. As back propagation passes through the sigmoid function, the differential on the chain is very small. Further, back propagation might pass through many sigmoid functions, finally resulting in little influence of weight ω on the loss function, which goes against weight optimization. This problem is called gradient saturation or gradient diffusion.
2. If the function output is not centered on 0, the weight updating efficiency would decrease.
3. The sigmoid function is applied in exponential operation, which is relatively slow for the computer.

Tanh function is a hyperbolic tangent function and its curve is similar to that of the sigmoid function. The same point is, as the inputs of the two functions are large or small, the outputs are almost smooth, and their gradients are very small, which is harmful to weight update. The difference is the output interval; the output interval of tanh is (−1, 1); and the whole function is centered on 0, which is better than the sigmoid function.

Compared to the first two activation functions, the ReLU function is one of the most popular activation functions and has the following advantages: There is no gradient saturation when the input is positive. The calculation speed is much faster. Since the ReLU function only has a linear relationship, it is much faster than the sigmoid and tanh functions in both forward and back propagation. Correspondingly, the ReLU function also has some disadvantages. When the input is negative, the ReLU function is not activated completely, which suggests that once the input is negative, it would die. During forward propagation, some areas are sensitive, while some are not, so the ReLU function would not die. However, when it comes to back propagation, if the input is negative, the gradient would exactly go to zero, which encounters the same problem as the sigmoid and tanh functions. The ReLU function is also not centered on 0. Please refer to [44] for more detailed introduction to the CNN.

3.3. Long- and Short-Term Memory Neural Network Model

As for the recurrent neural network (RNN), because the neurons have the looping structure, the characteristics of the previous state can be reserved. The structure of the RNN is shown in Figure 3. If the network input is a time series, it can be transformed into standard neurons connecting with each other.

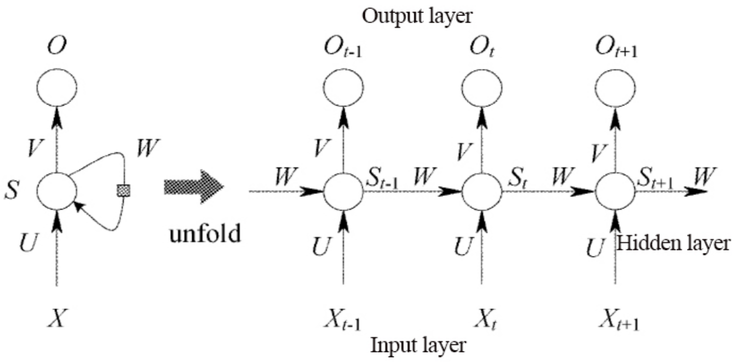


Figure 3. The neuron structure of the recurrent neural network.

As shown in Figure 3, each node represents a neuron at a time point in the time series. *U* represents the weight between the input layer and the hidden layer; *W*, the weight of circulating from the hidden layer to itself; and *V*, the weight between the hidden layer and the output layer. Furthermore, the weight coefficients are the same for each time series. The traditional RNN would lose certain information after each feedback because of its own structure, which means the original information would be lost and degraded, which is the so-called gradient disappearance. In order to solve the gradient disappearance problem, the gated recurrent unit (GRU) neural network is adopted. It is necessary to improve the neural network unit (the hidden layer of the RNN) and the memory unit should be added to form the GRU structure. However, compared to GRU, the LSTM neural network is a powerful solution to the gradient disappearance problem. As one of the RNN, the LSTM neural network can solve the gradient disappearance problem of the traditional RNN, so it can learn and train the relationship with long-term information transmission. In terms of the whole structure, the LSTM neural network is similar to the traditional RNN, including the input layer, hidden layer, and output layer. Unlike the GRU, it adds multiple control gates (memory modules) on the basis of the neuron structure of the traditional hidden layer in order to solve the gradient disappearance problem, thus realizing long-term memory and information transmission. Specific introduction of the LSTM neural network can be referred to in the literature [45].

4. Example Verification and Comparison

4.1. Introduction of Data Source

In this study, the rolling bearing vibration signal data have been collected from the Case Western Reserve University Bearing Data Center for fault classification (see Supplementary Materials). The specific structure of the test platform includes four parts: the motor is on the left, the torque sensor is in the middle, the power meter is on the right, and an electronic control device that cannot be seen. On the basis of the horsepower (1 hp = 746 W), the motor has four different working conditions—namely, 1 hp, 2 hp, 3 hp, and 4 hp. The bearing to be tested is used to support the motor shaft in the test platform. Before the operation of the whole mechanical system, a single-point fault is arranged on the bearing (adopting the spark erosion technology), with the fault diameters of 0.007, 0.014, and 0.021 in., respectively. There are four different kinds of bearing states: (1) normal, (2) inner ring fault, (3) ball fault, and (4) outer ring fault. The vibration data generated by bearing operation under these four different fault states would be different; each fault state also includes three fault diameters, 0.007, 0.014, and 0.021 in., respectively. Therefore, there are a total of 10 fault states (including normal state) corresponding to 1 hp.

4.2. Data Processing

In this study, the data are the time series data of 480,000 * 1 fault under 3 hp. The data processing method is as follows: Since the upper limit of the data to be read is limited by the learner, the original vibration data are segmented with a step length of 2000 to obtain the vibration data of 240 * 2000 in each state. Since the bearing contains 10 fault states, the data set (including labels) of 2400 * 2000 is finally obtained.

4.3. Parameter Setting

In order to compare the results of the SVM, CNN, and LSTM, it is necessary to set the same parameters for all of them. The specific setting is as follows: The sample size is 8000, and the proportion of the training, validation, and test sample is 7:2:1, batch = 200. The Adam stochastic optimization algorithm is selected for the optimization algorithm. The learning rate is set as 0.06; the ReLU function is selected as the activation function. The Softmax classifier is adopted. The number of hidden layers is 2, as well as stride = 1, padding = same, and dropout = 1.

4.4. Analysis of Results

4.4.1. SVM

The parameters of the SVM are as follows: (1) C, penalty coefficient; (2) gamma, coefficient of the kernel function; (3) kernel, kernel function. We used rbf as the kernel function. Grid search with cross-validation (GridSearchCV) was used to select penalty coefficient parameters. The following figure shows the performance of the learner when C takes different values. Macro avg was used as the evaluation function. The value of C was obtained by three-fold cross-validation.

It can be seen that when C exceeds 32, the accuracy of the learner on the training set and test set remains basically invariant (Figure 4). Considering that the higher the value of C, the higher the training cost, therefore we set the parameter C = 32 on the premise that the performance of the learner was relatively stable.

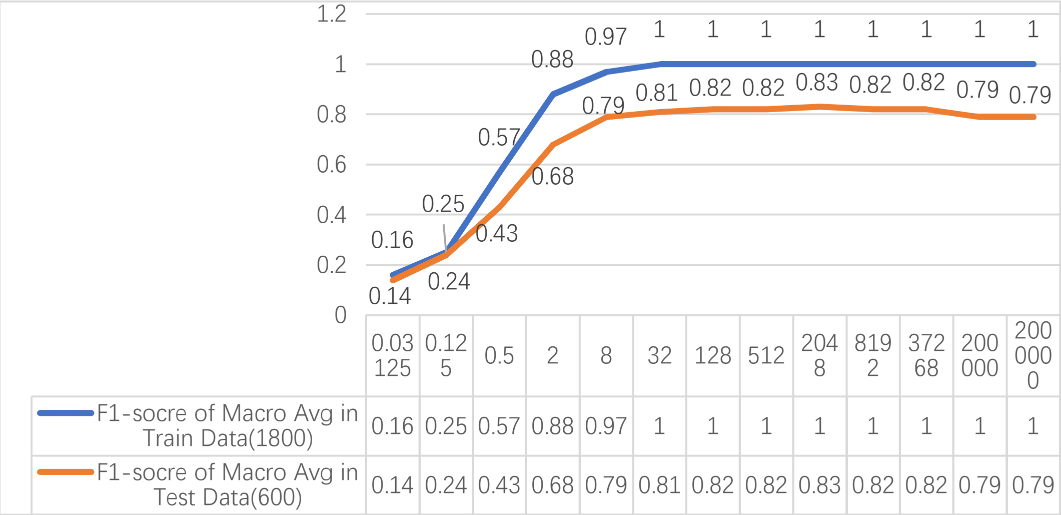


Figure 4. Classification effect of the support vector machine (SVM). Kernel = rbf, gamma = 1/n features.

Then we randomly split the data to a training set and a test set 10 times to improve the reliability of the results. Results are shown in Table 1 below.

Table 1. Classification effect of the support vector machine (SVM).

Times	1	2	3	4	5	6	7	8	9	10	Average
F1-Score Training	1	1	1	1	1	1	1	1	1	1	1
F1-Score Test	0.81	0.80	0.83	0.77	0.79	0.81	0.82	0.81	0.79	0.81	0.804
Accuracy Test	0.8	0.81	0.83	0.83	0.81	0.83	0.82	0.81	0.81	0.79	0.814

4.4.2. CNN and LSTM

When choosing the parameter C of SVM, we used GridSearchCV, which combines grid search with three-fold cross-validation. In the SVM, CNN, and LSTM, 10-fold cross-validation was used to validate the learner. The training set and validation set were put into the learner according to the 10-fold cross strategy to ensure that the actual training data amount reached 90% of the original data set, and then the test set (10%) was used to test the learner. We also trained many times to improve the reliability of the results. The results of the CNN and LSTM are as follows (Tables 2 and 3).

Table 2. Classification effect of convolutional neural network (CNN).

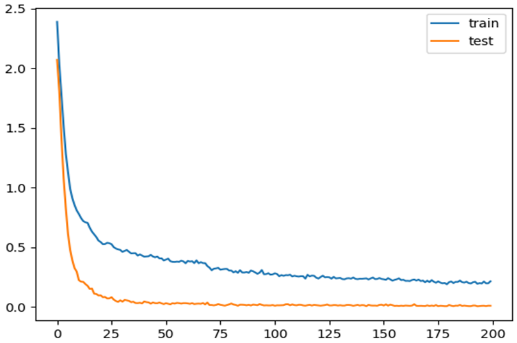
Times	Accuracy	F1-Score	Loss
1	0.975	0.989	

Table 2. *Cont.*

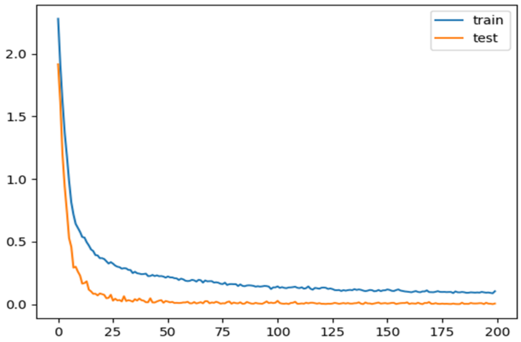
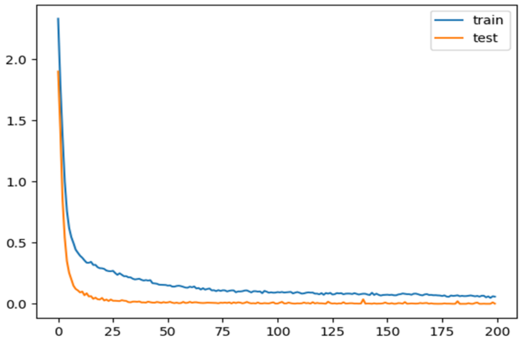
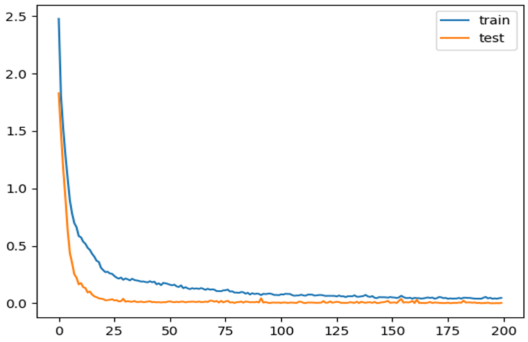
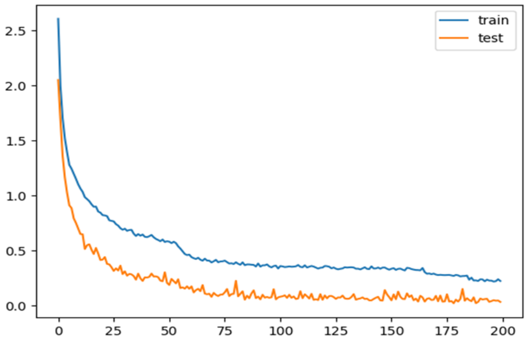
Times	Accuracy	F1-Score	Loss
2	0.9817	0.998	
3	0.982	0.995	
4	1.0	1.0	
5	0.9817	0.986	

Table 2. *Cont.*

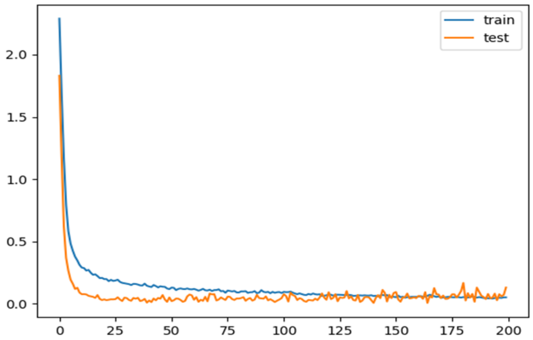
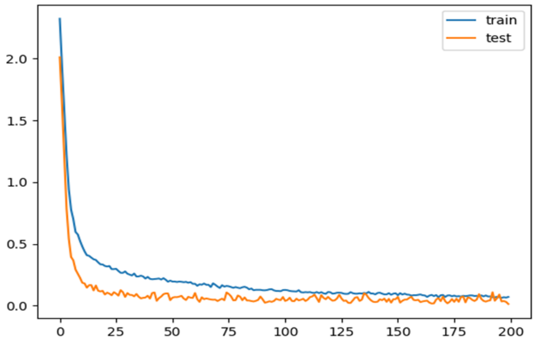
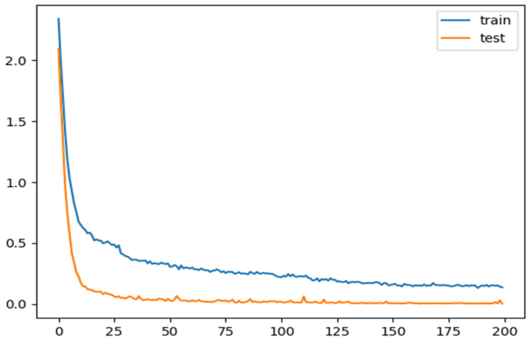
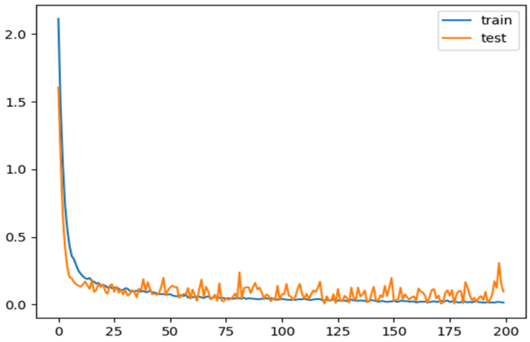
Times	Accuracy	F1-Score	Loss
6	0.9833	0.977	
7	0.9834	0.997	
8	1.0	1.0	
9	0.9843	0.988	

Table 2. *Cont.*

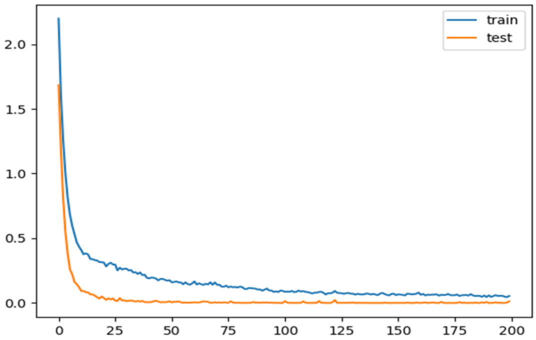
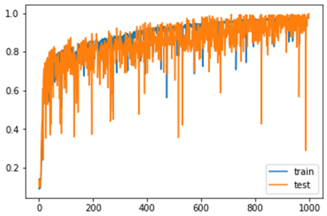
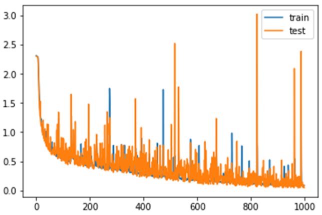
Times	Accuracy	F1-Score	Loss
10	0.9921	0.995	
Average	0.9828	0.9925	

Table 3. Classification effect of long- and short-term memory (LSTM) neural network.

Val_acc of the Last 10 Iterations	Accuracy	Loss	F1-Score
0.9733333388964335, 0.9466666777928671, 0.9333333373069763, 0.9900000095367432, 0.9933333396911621, 0.9900000095367432, 0.9733333388964335, 0.9900000095367432, 0.9900000095367432, 0.996666669845581			0.97

4.4.3. Comparison of Results

After finishing the classification for the SVM, CNN, and LSTM, the typical indexes are selected to compare the results, as shown in Figure 5.

It was found that the neural network method is better than the SVM in classification accuracy. The reason is that when the sample size is sufficient, the neural network method can show a greater superiority than the SVM. In addition, compared to LSTM, the CNN was promoted in the accuracy of the training set and test set. Therefore, it is concluded that when the sample size is sufficient, the classification by the neural network method is more accurate than the SVM in the field of classification and fault diagnosis for the time series data, and the CNN has greater superiority than LSTM. It is worth mentioning that LSTM was difficult to train. The following table shows the timing performance of the three chosen methods (Table 4). Our hardware environment and platform settings are as follows:

- Operation system: Windows 10, 64bit
- Central processing unit: I7-8750H@2.20GHz, 12-core
- Graphics processing unit: Nvidia Geforce GTX 1060max-Q (6 GB)
- Memory: DDR4-2666 8G+ DDR4-2666 4G
- Hard disk: KBG30ZMS128G NVME TOSHIBA
- Programming language and development environment: Python 3.6, Anaconda3-5.4.0
- Machine learning platform: TensorFlow 1.13.0

In order to see if the out-performances was by chance, we used the random forest method to test whether the results were accidental. The table below shows the results of random forest (Table 5).

The random forest method performed well on the training set, and the training time was very short. However, the method is not very effective on the test set. The cause may be because the large number of sample features leads to the poor distribution of feature weights in decision trees in the random forest. The following is the structure diagram of a single decision tree in random forest (Figure 6). Testing results show that the out-performances of the three chosen methods are relatively accurate.

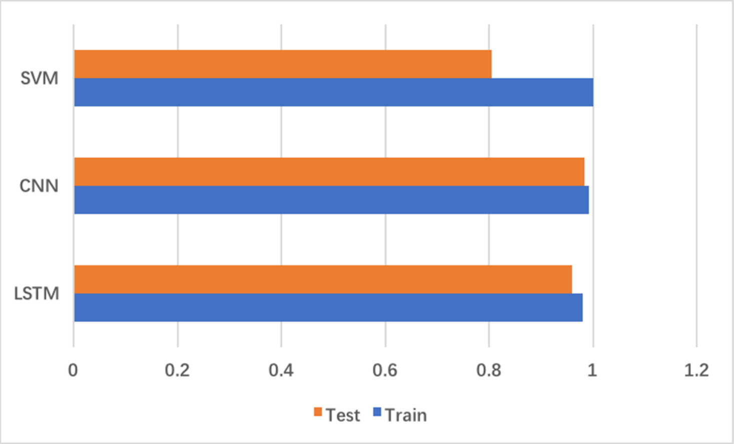


Figure 5. Comparison for the classification effect of the support vector machine (SVM), convolutional neural network (CNN), and long- and short-term memory (LSTM) neural network.

Table 4. Training time of SVM, CNN, and LSTM.

Method	Training Time
SVM-RBF	13.72 s
CNN epoch = 200	9 min 17 s
LSTM epoch = 1000	62 h 5 min

Table 5. Classification effect of random forest.

Data set	Precision	Recall	F1-Score
Train macro avg	1.00	0.91	0.95
Test macro avg	0.75	0.12	0.21
Training time		3.2S	

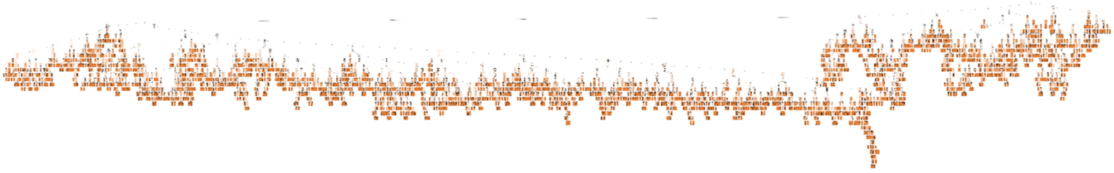


Figure 6. Structure diagram of a single decision tree in random forest.

5. Conclusions

The reliability and sustainability of complex systems in industry, especially complex systems with high-end equipment, is a matter of concern, because it relates to efficiency, cost, resources, energy consumption, human–computer interaction, and many other aspects. Using large data, sustainability calculation, and multi-method comparison can improve the cost-benefit of complex systems and minimize the risk of failure while avoiding waste and redundancy. A more reasonable selection of fault diagnosis methods will greatly reduce the overall demand for human resources, energy and time, and help to improve efficiency and accuracy. This paper presents a fault diagnosis and prediction framework for complex systems based on multi-dimensional data and multi-method comparison, aimed at improving the reliability and sustainability of the system by selecting methods with relatively superior performance. Three methods—SVM, CNN, and LSTM—were used for programming, then their results were compared, thus drawing the following conclusions:

1. The new method is better than the traditional and single statistical analysis method.
2. For the classification of the time series fault data, the accuracy of the neural network is higher than that of the SVM.
3. The CNN and LSTM both performed well. The CNN has slight superiority than LSTM regarding accuracy. More than that, LSTM is much more difficult to train. It takes much more time and requires higher equipment conditions. Generally, the CNN has greater advantages in the classification of the time series fault data than LSTM.

Through this method and train of thought, we can choose the most effective fault diagnosis and prediction method according to the characteristics of different systems and data sets, so as to improve the reliability and sustainability of complex systems.

In addition to these findings, there are some problems to be further studied. There is no doubt that model-based reasoning and data-supported reasoning are different. With the development of big data technology and the shift of projects from simulation to the physical world, data will play an increasingly important role, especially in reliability and sustainability applications. In this paper, a framework of production operation using industrial big data and sustainable development guiding ideology in the industrial field is proposed, which aims to illustrate that industrial big data and computational sustainability are of great benefit in improving the reliability and sustainability of complex systems. Due to the variety and rapid development of data mining and machine learning methods, the method selection in this paper is representative but also has limitations. The machine learning methods for fault diagnosis are still not comprehensive enough, and some methods such as other neural networks, integrated learning, and random forest should be further developed to compare the comprehensive effect for fault classification. Follow-up research can further broaden and supplement algorithms with better performance according to the trends and hotspots of sustainable computing.

Supplementary Materials: The dataset used is available at: http://csegroups.case.edu/bearingdatacenter/pages/download-data-file.

Author Contributions: Conceptualization, C.L. and G.J.; methodology, C.L.; software, C.L.; validation, G.J.; data curation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, G.J.; funding acquisition, G.J.

Funding: This research was funded by the State Administration of Science, Technology and Industry of China, grant number JSZL2016601A004.

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

References

1. Gomes, C.P. Computational sustainability: Computational methods for a sustainable environment, economy, and society. *Bridge* **2009**, *39*, 5–13.
2. Frenkel, K.A. Computer science meets environmental science. *Commun. ACM* **2009**, *52*, 23. [CrossRef]
3. Murgante, B.; Borruso, G.; Lapucci, A. *Geocomputation, Sustainability and Environmental Planning*; Springer-Verlag: Berlin, Germany, 2011; pp. 18–39.
4. Hassani, H.; Huang, X.; Silva, E. Digitalisation and Big Data Mining in Banking. *Big Data Cogn. Comput.* **2018**, *2*, 18. [CrossRef]
5. Hassani, H.; Huang, X.; Silva, E. Big-Crypto: Big Data, Blockchain and Cryptocurrency. *Big Data Cogn. Comput.* **2018**, *2*, 34. [CrossRef]
6. Chen, M.; Yang, J.; Hao, Y.; Mao, S.; Hwang, K. A 5G cognitive system for healthcare. *Big Data Cogn. Comput.* **2017**, *1*, 2. [CrossRef]
7. Chen, M.; Li, W.; Hao, Y.; Qian, Y.; Humar, I. Edge cognitive computing based smart healthcare system. *Future Gener. Comput. Syst.* **2018**, *86*, 403–411. [CrossRef]

8. Steinhaeuser, K.; Chawla, N.V.; Ganguly, A.R. An exploration of climate data using complex networks. *ACM Sigkdd Explor. Newsl.* **2010**, *12*, 25. [CrossRef]
9. Steinhaeuser, K.; Chawla, N.V.; Ganguly, A.R. Complex networks as a unified framework for descriptive analysis and predictive modeling in climate science. *Stat. Anal. Data Min.* **2011**, *4*, 497–511. [CrossRef]
10. Lee, K.J.; Kahng, H.; Kim, S.B. Improving Environmental Sustainability by Characterizing Spatial and Temporal Concentrations of Ozone. *Sustainability* **2018**, *10*, 4551. [CrossRef]
11. Hassani, H.; Huang, X.; Silva, E. Big Data and Climate Change. *Big Data Cogn. Comput.* **2019**, *3*, 12. [CrossRef]
12. Anderson, R.P.; Dud, K.M.; Ferrier, S. Novel methods improve prediction of species' distributions from occurrence data. *Ecography* **2006**, *29*, 129–151.
13. Hastie, T.J.; Tibshirani, R.J.; Friedman, J.J.H. *The Elements of Statistical Learning*; Springer-Verlag: Berlin, Germany, 2009; pp. 56–97.
14. Feng, X.; Li, Q.; Zhu, Y.; Hou, J.; Jin, L.; Wang, J. Artificial neural networks forecasting of PM 2.5 pollution using air mass trajectory based geographic model and wavelet transformation. *Atmos. Environ.* **2015**, *107*, 118–128. [CrossRef]
15. Ekasingh, B.; Ngamsomsuke, K.; Letcher, R. A data mining approach to simulating farers' crop choices for integrated water resources management. *J. Environ. Manag.* **2005**, *77*, 315–325. [CrossRef] [PubMed]
16. Juan, R.P.; Emmanuel, L.; Juan, C.C.A. Estimating avocado sales using machine learning algorithms and weather data. *Sustainability* **2018**, *10*, 3498.
17. Mazzoni, D.; Logan, J.; Diner, D. A data-mining approach to associating MISR smoke plume heights with MODIS fire measurements. *Remote Sens. Environ.* **2007**, *107*, 138–148. [CrossRef]
18. Wang, L.; Zhao, Q.; Wen, Z. RAFFIA: Short-term Forest Fire Danger Rating Prediction via Multiclass Logistic Regression. *Sustainability* **2018**, *10*, 4620. [CrossRef]
19. Zhang, J.; Wang, H. Looking at the development direction of aviation maintenance from the CMCS of Boeing 777. *Aviat. Eng. Maint.* **2000**, *2*, 25–36.
20. Wang, B.K.; Chen, Y.F.; Liu, D.T.; Peng, X.Y. An embedded intelligent system for on-line anomaly detection of unmanned aerial vehicle. *J. Intell. Fuzzy Syst.* **2018**, *34*, 3535–3545. [CrossRef]
21. Yuan, C.; Yao, H. Aeroengine intelligent performance diagnosis under sensor measurement deviation. *Aviat. Dyn. Bull.* **2007**, *22*, 126–131.
22. Petrov, N.; Marinov, B.M. Methods for statistical valuation of the technical condition of aviation communication equipment. *Acad. Open Internet J.* **2000**, *1*, 133–147.
23. Yang, Z. Research on Intelligent Fault Diagnosis Expert System for Aircraft Aviation and Electrical Equipment. *Meas. Control Technol.* **2006**, *225*, 4–7.
24. Zhou, Y.; Cao, M. Application of Fuzzy Outward-Bound Fault Diagnosis in Aircraft Power System. *J. Aviat.* **1999**, *20*, 368–370.
25. Bu, J.; Sun, R.; Bai, H.; Xu, R.; Xie, F.; Zhang, Y.; Ochieng, W.Y. Integrated method for the UAV navigation sensor anomaly detection. *IET Radar Sonar Navig.* **2017**, *11*, 847–853. [CrossRef]
26. Fan, Z. Application of Kohonen Network in Engine Fault Diagnosis. *Aviat. Dyn. Dly.* **2000**, *15*, 89–92.
27. Du, C. Fault Diagnosis Method of Avionics System Based on Rough Neural Network. *Fire Power Command Control* **2006**, *31*, 48–50.
28. Xu, K.; Jiang, L. Aeroengine fault diagnosis based on Lyapunov exponential spectrum. *J. Appl. Mech.* **2006**, *23*, 488–492.
29. Shi, R. Fault Diagnosis Expert System. *J. Beijing Univ. Aeronaut. Astronaut.* **1995**, *21*, 7–12.
30. Baig, M.F.; Sayeed, N. Model-based reasoning for fault diagnosis of twin-spool turbofans. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **1998**, *212*, 109–116. [CrossRef]
31. Palmer, K.A.; Bollas, G.M. Analysis of transient data in test designs for active fault detection and identification. *Comput. Chem. Eng.* **2019**, *122*, 93–104. [CrossRef]
32. Kordestani, M.; Samadi, M.F.; Saif, M. 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]
33. Anami, B.S.; Pagi, V.B. Localisation of multiple faults in motorcycles based on the wavelet packet analysis of the produced sounds. *IET Intell. Transp. Syst.* **2013**, *7*, 296–304. [CrossRef]
34. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Internet Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]

35. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [PubMed]
36. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000; Available online: https://doi.org/10.1017/CBO9780511801389 (accessed on 2 July 2019).
37. Vapnik, V.N. *Statistical Learning Theory*; John Wiley & Sons Inc.: New York, NY, USA, 1998; pp. 736–768.
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, (ICLR 2015), San Diego, CA, USA, 7 May 2015; p. 13.
39. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer: Berlin, Germany, 2005; pp. 878–887.
40. Van, J.; Koc, M.; Lee, J. A prognostic algorithm for machine performance assessment and its application. *Prod. Plan. Control* **2004**, *15*, 796–801.
41. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
42. Hussain, S.F. A novel robust kernel for classifying high-dimensional data using Support Vector Machines. *Expert Syst. Appl.* **2019**, *131*, 116–131. [CrossRef]
43. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [CrossRef]
44. Guo, Y.M.; Liu, Y.; Bakker, E.M.; Guo, Y.H.; Lew, M.S. CNN-RNN: A large-scale hierarchical image classification framework. *Multimed. Tools Appl.* **2018**, *77*, 10251–10271. [CrossRef]
45. Felix, A.G.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471.