

Article



## Intelligent Fault Diagnosis Method for Blade Damage of Quad-Rotor UAV Based on Stacked Pruning Sparse Denoising Autoencoder and Convolutional Neural Network

Pu Yang <sup>1,2,\*</sup>, Chenwan Wen <sup>1,2</sup>, Huilin Geng <sup>2</sup> and Peng Liu <sup>2</sup>

- Key Laboratory of Advanced Aircraft Navigation, Control and Health Management, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; wenchenwan@nuaa.edu.cn
- <sup>2</sup> College of Automation Engineering, Nanjing University of Aeronautics and Astronautics,
- Nanjing 211106, China; hlgeng@nuaa.edu.cn (H.G.); llpeng@nuaa.edu.cn (P.L.)
  - Correspondence: ppyang@nuaa.edu.cn

check for updates

Citation: Yang, P.; Wen, C.; Geng, H.; Liu, P. Intelligent Fault Diagnosis Method for Blade Damage of Quad-Rotor UAV Based on Stacked Pruning Sparse Denoising Autoencoder and Convolutional Neural Network. *Machines* **2021**, *9*, 360. https://doi.org/10.3390/ machines9120360

Academic Editors: Hongtian Chen, Kai Zhong, Guangtao Ran and Chao Cheng

Received: 4 November 2021 Accepted: 14 December 2021 Published: 16 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Abstract: This paper introduces a new intelligent fault diagnosis method based on stack pruning sparse denoising autoencoder and convolutional neural network (sPSDAE-CNN). This method processes the original input data by using a stack denoising autoencoder. Different from the traditional autoencoder, stack pruning sparse denoising autoencoder includes a fully connected autoencoding network, the features extracted from the front layer of the network are used for the operation of the subsequent layer, which means that some new connections will appear between the front and rear layers of the network, reduce the loss of information, and obtain more effective features. Firstly, a one-dimensional sliding window is introduced for data enhancement. In addition, transforming one-dimensional time-domain data into the two-dimensional gray image can further improve the deep learning (DL) ability of models. At the same time, pruning operation is introduced to improve the training efficiency and accuracy of the network. The convolutional neural network model with sPSDAE has a faster training speed, strong adaptability to noise interference signals, and can also suppress the over-fitting problem of the convolutional neural network to a certain extent. Actual experiments show that for the fault of unmanned aerial vehicle (UAV) blade damage, the sPSDAE-CNN model we use has better stability and reliable prediction accuracy than traditional convolutional neural networks. At the same time, For noise signals, better results can be obtained. The experimental results show that the sPSDAE-CNN model still has a good diagnostic accuracy rate in a high-noise environment. In the case of a signal-to-noise ratio of -4, it still has an accuracy rate of 90%.

**Keywords:** intelligent fault diagnosis; stacked pruning sparse denoising autoencoder; convolutional neural network; anti-noise

## 1. Introduction

UAVs are very suitable for performing tasks in spacious indoor and outdoor environments, such as personnel search and rescue, material transportation, military patrol and surveillance, pesticide spraying, crop seeding, etc. Due to the increasing complexity of the tasks performed by drones, the sensors and actuators on the drone are becoming more and more complex, and the reliability requirements of the drone are getting higher and higher during the mission. Once the drone has a serious fault in flight, it will cause more serious property losses, and in more serious cases, it may cause casualties [1]. During the flight of the drone, any minor fault can easily cause the drone itself to malfunction, thereby affecting the sensors, actuators, and other related equipment on the drone. Therefore, the safety and reliability of UAVs is now an issue worthy of study and discussion. At the same time, we also need to specifically consider the different types of faults of different types of UAV [2]. For various faults on the drone, the drone control system can respond to the faults on the drone only when the system identifies and diagnoses each fault, respectively, so as to minimize the loss of personnel and property in the case of UAV faults. One of the main issues is the identification of faults in drones. The identification of faults is mainly divided into knowledge-based fault methods, model-based fault diagnosis methods, and data-based fault diagnosis methods.

The main knowledge-based fault diagnosis methods are symbolic expert systems [3], symbolic directed graph (SDG) methods, and fault tree methods. In [4], the symbolic directed graph is introduced, the symbolic directed graph is mainly a graphical model based on causality. In [5], the fault diagnosis method based on the fault tree is mainly introduced, the fault tree uses a graphical method for fault diagnosis. A fault tree is formed by connecting the fault in the system and the cause of the system fault. When the system fails, the cause of the system fault is deduced from the current fault state of the system from the bottom to the top. As a knowledge-based fault diagnosis method, the diagnosis model is simple, and the diagnosis results are easier to apply in practical engineering. However, because knowledge-based fault diagnosis requires learning the types of faults to be diagnosed, when a fault that is not in the knowledge base occurs in the system, the system will not be able to provide the correct diagnosis result.

The model-based fault diagnosis method [6] is based on the accurate mathematical model of the system. In the analytical model of the system, the residual signal between the input and output of the system is obtained by observation and measurement. By analyzing the residual signal in the system, the difference between the actual output and the expected output of the system can be obtained. Therefore, the system can be diagnosed based on these.

The data-driven fault diagnosis method is to classify and identify all the non-faulty and faulty data of the system, so the system's fault diagnosis can be realized without obtaining the precise mathematical model of the system. Data-based fault diagnosis methods mainly include machine learning methods [7], signal processing methods [8], information fusion methods [9], rough set methods [10], multivariate statistical analysis methods [11], etc. Because the data-based fault diagnosis method does not rely on the accurate model of the system for diagnosis, it is better to use the data-based method for fault diagnosis for complex high-level systems that are difficult to accurately model. However, because the data-based fault diagnosis method does not depend on the internal structure of the system, the interpretability of the results of system fault diagnosis is not very good [12].

At present, many intelligent fault diagnosis methods have been proposed in various research fields. In literature [13,14], the bearing is taken as the research object to study the relationship between the data collected by the bearing in different types of damage; in article [15,16], the fault diagnosis of drill is realized by analyzing the thermal image and vibration data of drill; in [17,18], the researchers took the battery pack as the research object and applied the intelligent fault diagnosis algorithm proposed by themselves to the actual battery system to diagnose the battery pack; in the research field of gearbox and high-speed train, a large number of fault diagnosis methods have also been proposed; in [19,20], it was studied how to judge the fault type through the collected signal when the gearbox fails; several new intelligent fault diagnosis methods are mainly proposed in [21–23], and good results have been achieved in the fault diagnosis of high-speed trains. Although many fault diagnosis methods have been proposed, there are still few intelligent fault diagnosis methods for UAVs. Therefore, we choose the quad-rotor UAVs as the research object in this paper.

During the operation of the quad-rotor UAV, the actuator or structure of the drone malfunctioned due to the operation problem of the pilot or due to some non-human reasons. In the literature [24], the researchers collected the vibration signal of the aircraft frame through the analysis of these data to diagnose whether the motor is malfunctioning. In [25], the researchers artificially damaged the rotor of the drone, and then collected the noise of the drone during the flight, and used the deep learning method to analyze and process the noise to realize the fault diagnosis of the system. The collection of sound

signals has strict requirements on the environment, so this method cannot be applied in practice. In the literature [26], the author introduced the convolutional neural network with a wide convolution kernel into the fault diagnosis method, and diagnosed the bearing data through the convolutional neural network. A wide convolution kernel can improve the anti-interference ability of convolutional neural network to some extent. In reference to the problem of inaccurate diagnosis results for data with large noise signals, the literature [27] proposed to denoise the data based on the stack denoising autoencoder, and achieved good results, but due to the introduction of a new network structure, the convergence speed of the training network has been adversely affected to a large extent. Most of the existing fault diagnosis algorithms need to preprocess the data to eliminate the noise interference in the data, thereby improving the accuracy of classification, but there are few methods to directly classify the original noisy data and obtain a good classification accuracy.

In response to the above-mentioned problems, we adopted a method called Stacked Pruning Sparse Denoising Autoencoder and Convolutional Neural Network (sPSDAE-CNN) to identify and classify the actuator damage fault of the UAV. The main contribution of this paper is as follows:

- 1. We use a new and improved convolutional neural network method, which can be directly applied to the original UAV data collected in practice. Compared with the traditional method, it does not require separate data preprocessing. The comparison is shown in Figure 1;
- 2. The method uses a stack denoising autoencoder as the first layer of the convolutional neural network, which is very robust against data with much noise in the data, and still has a relatively high fault diagnosis accuracy rate under high noise conditions;
- 3. Directly convert the sensor data collected by the drone into a gray sampling map. Expanding the dimensionality of the sample can further improve the feature extraction ability of the DL model;
- 4. This method is aimed at the problem that enough data cannot be collected during neural network training. We use a one-dimensional sliding window for overlapping sampling to enhance the data, increase the data scale, and improve the generalization of the neural network ability;
- 5. We use the feature maps learned by visualizing sPSDAE-CNN to explore the actual feature learning and classification mechanism of the sPSDAE-CNN model. At the same time, the pruning operation is introduced to speed up the training of SDAE.



**Figure 1.** Three kinds of intelligent fault diagnosis framework. (**A**) The feature extraction of unsupervised learning [28]. (**B**) The traditional method. (**C**) The method used in this article.

At present, there is much research on sensor fault and actuator fault of four-rotor UAVs. In article [29,30], it is mainly studied to diagnose the actuator fault of four-rotor UAV by using the traditional model class method, including hybrid observer and adaptive neural network observer. In [31], Kalman filter is mainly used to process the sensor data of UAV and then to diagnose the possible sensor faults. In [32], researchers proposed a disturbance observer to observe the faults in the system and then realized diagnosis and fault-tolerant control through sliding mode control method.

However, there is little research on the fault of UAV blade damage. In the process of a UAV mission, when the UAV blade is damaged to a certain extent, when the damage does not exceed the threshold, the UAV may still be able to perform the mission in the environment of small interference. However, at this time, the stability of the UAV has been greatly damaged, and there may be some risks during the mission. Therefore, we need to evaluate the blade damage of UAV through the proposed method, and timely evaluate the health state of UAV, so as to prevent UAV crashes. At the same time, we also introduce a sparse pruning stack noise reduction autoencoder to improve the adaptability of the model to high noise data. In addition, pruning operation is added to improve the algorithm complexity of the model. At present, most fault diagnosis methods for four-rotor UAVs are verified by numerical simulation. This paper collects experimental data on the actual aircraft and verifies the algorithm, which has good practicability.

There is not a simple linear relationship between the damage of the drone blades and the sensor data of the drone. Therefore, the sensor data of the drone blades under different damage conditions are analyzed by using the deep learning method, and a deep learning model about the relationship between the sensor data of the drone and the damage degree of the blades is obtained, and the model is optimized.

The remaining organizational structure of this article is as follows: Section 2 briefly introduces the convolutional neural network and the stack denoising autoencoder. Section 3 introduces the intelligent fault diagnosis method based on sPSDAE-CNN. In Section 4, we use experiments to verify the sPSDAE-CNN method, and compare and analyze it with some commonly used methods. At the end of Section 5, we draw conclusions and propose future work by summarizing the work.

# 2. Introduction to the Convolutional Neural Network and Stack Denoising Autoencoder

## 2.1. A Brief Introduction to Convolutional Neural Networks

In this part, we will briefly introduce the convolutional neural network and the stack denoising autoencoder. For more details about the neural network, please refer to the literature [33]. Convolution neural network is a multilevel deep neural network [34]. Its basic structure consists of the input layer, convolution layer, activation layer, pooling layer, full connection layer, and output layer. Generally, there are several convolution layers and pooling layers, and the general structure is a convolution layer connected with a pooling layer. Each neuron in the input is locally connected to the input, and the weighted summation with the local input through the corresponding connection weight and the bias is added to obtain the input of the neuron. This process is equivalent to the convolution process, so it is called a convolutional neural network.

## 2.1.1. Convolutional Layer

The convolutional layer uses a convolution kernel to perform convolution operations on our input data or local regions of features, and extract relevant features from the data. Figure 2 shows the structure diagram of the convolutional layer and the pooling layer. The top layer is the pooling layer, the middle is the convolutional layer, and the bottom is the input layer [34]. In Figure 2, convolution neurons are organized into feature planes, and each neuron in the convolution layer is locally connected to the feature surface in its input layer. The output of each neuron in the convolution layer can be obtained by passing the local weighting and transfer to the activation function.

An important feature of convolutional neural networks is weight sharing. The weights of convolutional neural networks in the plane of the same input feature and the same output feature are shared. Weight sharing also reduces the complexity of the network model to a certain extent. It also avoids the over-fitting problem caused by too many parameters. In actual operations, most of the related operations can be replaced by convolution operations, which can avoid the problem of reversing the convolution kernel during backpropagation. The formula for convolution operation is shown in (1):

$$y_i^{k+1}(j) = \kappa_i^k \times \chi^k(j) + b_i^k,\tag{1}$$

where  $\kappa_i^k$  and  $b_i^k$  respectively represent the weight and bias of the *k*th filter kernel of the *i*th layer of the neural network, and use  $\chi^k(j)$  to represent the *j*th local region of the *k*th layer. Where  $\times$  is used to calculate the inner product of the kernel and the local area, and  $y_i^{k+1}(j)$  represents the input of the *j* neuron in the frame *i* of the k + l layer.



Figure 2. Schematic diagram of the convolutional layer and the pooling layer structure.

#### 2.1.2. Activation Layer

After the convolutional layer, we need to use the activation function to introduce nonlinear modeling capabilities to our neural network, eliminate redundant data in the data, and enhance the learning ability of the neural network, so that the features in the data can be further processed for segmentation. The commonly used activation functions mainly include sigmoid function, tanh function, ReLu function, ELU function, etc. For details, please refer to the literature [35]. In our convolutional neural network, we choose to use the ReLu function as the activation function. Its main feature is compared with linear functions. ReLu has better expression ability compared with nonlinear functions, as ReLu does not have the problem of gradient disappearance and can maintain the convergence rate of the model in a stable state. The ReLu function is expressed as follows (2):

$$x_i^{k+1}(j) = \operatorname{ReLu}(y_i^{k+1}(j)) = \max\{0, y_i^{k+1}(j)\},$$
(2)

where  $y_i^{k+1}(j)$  represents the output of the first convolutional layer, and  $\alpha_i^{k+1}(j)$  represents the result of  $y_i^{k+1}(j)$  activated by ReLu.

## 2.1.3. Pooling Layer

The pooling layer is also one of the most common and basic mechanisms of convolutional neural networks. It is actually a form of downsampling, and there are many forms of nonlinear pooling functions in convolutional neural networks. Max pooling function is the most common one. The principle of this mechanism is that when a feature of data is discovered, its exact location is far less important than its relative location with other features. Pooling reduces the size of the data space by constantly reducing the number of network parameters and the amount of computation. Overfitting can also be suppressed to some extent. The max-pooling operation can be expressed as shown in Figure 3:

The expression is (3):

$$a^{k}_{(nh,nw,c)} = \max(a^{k-1}_{(nh\times stride:nh\times stride+f,nw\times stride:nw\times stride+f,c)}),$$
(3)

where *nh* represents the height in the current pixel, *nw* represents the width of the current pixel, and *c* represents the channel, *f* represents the size of the pooling core, and *stride* represents the step size of the pooling core movement.



Max Pool with 2 by 2 filter and stride 2

Figure 3. Maximum pooling operation.

#### 2.1.4. Batch Normalization

Batch standardization was proposed in [36] to accelerate the training speed of deep neural networks by reducing the transfer of internal covariates. The batch normalization layer is usually added after the convolutional layer or the fully connected layer, and before the activation layer. Given p-dimensional data into the BN layer  $X = (x^{(1)}, \ldots, x^{(p)})$  the operation of the BN layer can be expressed as the following expression (4):

$$\widehat{x}^{(i)} = \frac{x^{(i)} - E(x^{(i)})}{\sqrt{Var[x^{(i)}]}}$$

$$y^{(i)} = \gamma^{(i)}\widehat{x}^{(i)} + \beta^{(i)},$$
(4)

where  $y^{(i)}$  represents the p-dimensional output of the BN layer, and  $\gamma^{(i)}$  and  $\beta^{(i)}$  are the scaling and bias that the BN layer needs to learn, which need to be learned in the neural network training.

#### 2.2. Stacked Denoising Autoencoder

The encoder is a commonly used learning model in deep learning. The structure of this model is shown in Figure 4. The stack noise reduction autoencoding network is based on the encoder. The encoder must learn to obtain noise-free input from the noisy data. Unlike the supervised learning model CNN and Recurrent Neural Networks (RNN) [34], it combines unsupervised data feature extraction with supervised overall fine-tuning, and it can mainly realize the noise reduction and dimensionality reduction of the features of high-noise information. The structure is shown in Figure 5. Stack noise reduction autoencoder and encoder are mainly composed of encoder and decoder, which can be used to extract hidden features of samples and reconstruct input.



Figure 4. The structure of the encoder.



Figure 5. The structure of the denoising autoencoder.

# 3. Proposed Convolutional Neural Network with Stacked Pruning Sparse Denoising Autoencoder

In this paper, an intelligent quadrotor UAV fault diagnosis method based on stacked pruning sparse noise reduction autoencoder and convolutional neural network is proposed. We mainly use sPSDAE as the first layer of the neural network to reduce noise and dimensionality of the original data. The introduction of stack pruning sparse noise reduction autoencoder can improve the model generalization ability of the neural network (CNN) is used to extract and classify system features. The algorithm model is shown in Figure 6:



Figure 6. sPSDAE-CNN algorithm model.

Firstly, collect the flight data of the drone. In order to simulate the damage of the blades of the drone in the actual flight, we collect the drone data by artificially damaging the blades of the quadrotor rotor drone in a laboratory environment. The individual blades of the UAV are set to have different degrees and types of damage. The main types and degrees of damage are shown in Table 1 below:

Table 1. Main types and degrees of damage.

Types of Damage to the Blades	Damage Degree of the Blade
No damage	0%
Broken blade	5%
Broken blade	10%
Broken blade	15%
Broken blade	20%
Blade crack	Slightly deformation
Blade crack	General deformation
Blade crack	Severely deformation

Eight different types and degrees of damage to the blades are shown in Figure 7:

20% slightly deformation General deformation Severely deformation

Figure 7. Eight different types of blade damage.

This paper chooses to use a quad-rotor drone with pixhawk4 flight control as the main control board for data collection. We let quad-rotor drones conduct flight experiments in different health states, collect data, and convert the collected data into a two-dimensional grayscale image. The paper selects the output of the four actuators in the flight log of the drone, the quaternion representing the attitude of the drone, the angular velocity on the three coordinate axes of the drone, and the position information, velocity information and acceleration information of the flight on the three coordinate axes of XYZ. Taking 20 sampling periods as a data state, a  $20 \times 20$  two-dimensional matrix is formed, which is converted into a  $20 \times 20$  grayscale image. As shown in Figure 8.



Figure 8. Converting one-dimensional time-domain signals to two-dimensional gray-scale images.

## 3.1. Proposed sPSDAE-CNN Model Structure

We convert the drone flight data after batch normalization (BN) into a grayscale image. Using stacked pruning sparse denoising autoencoders to reduce the dimensionality and denoising of the original data, it can also initially extract data features. The data processed by the sparse noise reduction autoencoder will be directly used as the input of the convolutional neural network. On the whole, the structure of the sPSDAE-CNN proposed in this paper is roughly the same as the structure of the traditional convolutional neural network. The main difference is that the stack noise reduction autoencoder is introduced, but the introduction of the noise reduction encoder further increases the complexity of the network and increases the computational cost, so the sparse pruning operation is added to reduce the complexity of the network. The noise reduction autoencoder improves the adaptability of the network to high-noise data, and the pruning operation greatly improves



the calculation efficiency of the encoder. The specific structure of sPSDAE-CNN is shown in Figure 9.

Figure 9. The specific structure of sPSDAE-CNN.

Finally, in the classification stage of the model, the softmax function is used to perform logit transformation on the classification results, and eight different four-rotor UAV health state probability distributions are obtained (5).

l

$$I(z_j) = \frac{e^{z_j}}{\sum\limits_{k}^{8} e^{z_k}},$$
(5)

where *z* represents the logical value of the *j*th neuron.

3.2. Construction of Sparse Noise Reduction Autoencoding Network

In order to explore the deep-level features in the time-domain sequence signal, we convert the one-dimensional time-domain sequence signal into a two-dimensional gray-scale image by using a matrix transformation method. Figure 9 shows the structure of a stacked noise reduction autoencoder with four hidden layers. Since each layer of a traditional stacked noise reduction encoder has an impact on its subsequent network levels, we use the pruning method to cut off the layers that have no effect on the training of the next layer of the network, while ensuring the maximum information flow in the network. Therefore, the latter layer can obtain the maximum effective information of the previous layer, which improves the training speed and feature extraction performance. The schematic diagram of constructing the stacked pruning sparse denoising autoencoder(sUPSDAE) fully connected network model based on the DAE model is shown in Figure 10:



Figure 10. Schematic diagram of sUPSDAE fully connected network model.

sUPSDE adopts the feature fusion method for information sharing, which reduces the loss of information and broadens the transmission level of the network. As the number of training layers increases, the number of network calculations will increase sharply, and it is also prone to the problem of overfitting. We reduce the amount of calculation by introducing sparse pruning operations while suppressing overfitting.

In Figure 10, we can get that the model of the *i*th layer, which is related to the first *i* unit nodes when it is trained. In order to introduce sparse operations into sUPSDEA, this paper randomly selects some features of the input layer in the training loop, and uses Formula (6) [38] to randomly discard it, and then periodically introduce sparse operations in subsequent node training until all units have been trained.

$$v = Berboulli(1 - p_1)$$
  
$$\overline{\beta_i}^* = v \times \overline{\beta_i} , \qquad (6)$$

where  $p_1$  represents the probability of the current training unit being discarded, and  $\overline{\beta}_i$  represents the input matrix before discarding.  $\overline{\beta_i}^*$  is the input matrix after random discarding in one cycle.

After the sUPSDEA training is over, backpropagation is performed by using Back Propagation Neural Network (BPNN) [39], and the parameters and weights of the network are fine-tuned. In this process, the discarded units are added through Equation (7) to further reduce the possible overfitting of the model.

$$\tau = Berboulli(1 - p_2)$$

$$\overline{X_i^*} = \tau \times \overline{X_i} , \qquad (7)$$

where  $p_2$  is the probability of discarding irrelevant nodes in the fine-tuning process,  $X_i$  is the output of the network in the fine-tuning process, and  $\overline{X_i}^*$  is the input data randomly discarded in one cycle of the fine-tuning process.

#### 3.3. The Influence of Various Parts of the Model on the Results

#### 3.3.1. The Effect of Sparse Pruning and Noise Reduction Autoencoder on the Results

The stack sparse noise reduction autoencoder transforms the original two-dimensional  $20 \times 20$  grayscale images into  $10 \times 10$  grayscale images by dimensionality reduction, which dramatically reduces the computational cost of the subsequent convolutional neural network. At the same time, the noise signal contained in the data can be filtered out, which also realizes the prediction of the original signal of the signal destroyed by the noise. By training the model parameters of the model, the model can finally achieve an accurate prediction of the original signal and eliminate the interference of noise to the original signal to a large extent, which can effectively improve the final diagnosis effect of the model.

#### 3.3.2. The Effect of Convolutional Neural Networks on Results

The convolutional neural network uses the output of the dimensionality reduction of the stack sparse noise reduction autoencoder as the input of the convolutional neural network, and uses the convolutional neural network to extract the characteristics of the data collected by the drone. By combining the high-dimensional input data, the feature is mapped to the low-dimensional UAV health status, which can easily convert the original data into the UAV health status. At the same time, it has a very good non-linear fitting ability, which is very beneficial to the fault diagnosis of the quad-rotor UAV, which improves the adaptive ability of the model to a certain extent.

#### 3.4. Data Augmentation

In order to recognize images using in-depth learning, a large amount of image data needs to be prepared for model training, especially when using neural network model algorithms. For example, most common data collections in in-depth learning contain a large amount of image data, including 60,000 training data and 10,000 test set data in the Mixed National Institute of Standards and Technology (MNIST) dataset. There are 60,000 color images in the Canadian Institute For Advanced Research-10 (CIFAR-10) dataset, of which 50,000 are training data and 10,000 are test data. Therefore, in order to train our own neural network model, we need to prepare a large amount of experimental data for the training model. However, the experimental data cannot meet the actual training requirements of the neural network, and data enhancement methods need to be introduced to increase the amount of sample data. In the field of computer vision, data enhancement is usually achieved by introducing operations such as flip, rotation, clipping, distortion, scaling, etc. However, such methods cannot be used in time domain sequence signals. We enhanced the data in fault diagnosis by introducing a fixed-length sliding window to slice sequential time-domain signals in turn, as shown in Figure 11.





Using this method, we will get 79,980 training samples from 80,000 original data collected by UAV. This method can effectively solve the problem of insufficient training samples in actual training, but this method has been ignored in many articles [40–43], because they do not use overlapping sampling methods for data enhancement. As a result, there are only hundreds or thousands of training samples during model training. At the bottom of the article, we will verify the necessity of data enhancement through actual experiments.

## 4. Validation of the sPSDAE-CNN Model

## 4.1. Data Description

The training of a neural network model requires a lot of data to be collected from a laboratory P200 quad-rotatory UAV. The main control panel of the UAV is pixhawk4, which is also equipped with jeson tx2, binocular camera, and other sensors, As shown in Figure 12.



Figure 12. P200 drone.

Over 90,000 data were collected in flight, of which 80,000 were valid. The training data is pre-processed and divided into four datasets, of which there are eight types of pre-defined faults, and eight types are considered to be the eight states of the UAV. The actual experimental data are shown in the Table 2.

Types of Dan	nage to the Blades	No Damage		Broker	n Blade			Blade Crack	
D	ata Set	0	5%	10%	15%	20%	slightly deformation	General deformation	Severely deformation
А	Train	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
	Test	200	200	200	200	200	200	200	200
В	Train	14,000	14,000	14,000	14,000	14,000	14,000	14,000	14,000
	Test	280	280	280	280	280	280	280	280
С	Train	18,000	18,000	18,000	18,000	18,000	18,000	18,000	18,000
	Test	360	360	360	360	360	360	360	360
D	Train	20,000	20,000	20,000	20,000	20,000	20,000	20,000	20,000
	Test	400	400	400	400	400	400	400	400

Table 2. Description of UAV datasets.

#### 4.2. Experimental Settings

This paper compares our methods with traditional convolution neural networks, SVM [44] and traditional unsupervised learning stack autoencoders. Consider the impact of different size datasets on the performance of the neural network, then compare the changes of the performance of the neural network before and after sparse pruning operation. Finally, experiments are carried out in different noise levels to compare and analyze the anti-noise ability of the sPSDAE-CNN model.

#### 4.2.1. Parameters of the Proposed Network

The sPSDAE-CNN network model proposed by this paper consists of an sPSDAE sparse pruning noise reduction autoencoder and a convolution neural network. The sPS-DAE consists of one input layer, one output layer, and four hidden layers. The specific structure is shown in Figure 9, The specific structural parameters of convolutional neural networks are shown in Table 3.

No	Layer Type	KernelSize Stride	Output Size (Width $ imes$ Depth)	Padding
1	Convolution1	$4 \times 4/1$	10  imes 10  imes 8	Yes
2	Pooling1	$4 \times 4/1$	$3 \times 3 \times 8$	No
3	Convolution2	$2 \times 2/1$	$6 \times 6 \times 8$	No
4	Pooling2	$3 \times 3/1$	$3 \times 3 \times 8$	Yes
5	Convolution3	$1 \times 1/1$	$3 \times 3 \times 16$	No
6	Pooling3	$2 \times 2/1$	$3 \times 3 \times 16$	Yes
7	Fully-connected	144	144  imes 1	/
8	Softmax	8	8	/

Table 3. Structural parameters of convolutional neural networks.

The introduced pruning operation also improves the training speed of the network. The output of sPSDAE is used as input of CNN. The main structure of a convolution network is three convolution layers and pooling layers. Then there is a hidden layer of full connection layer. Finally, the output layer is reached by a softmax layer. The convolution cores of the system select small convolution cores to convolute, the pooling layer chooses maximum pooling, and the activation function of the neural network chooses RELU function. In order to improve the performance of the network, a batch normalization operation is added behind each convolution layer and the full connection layer. The batch normalization operation can accelerate the convergence speed of the training of the neural network and suppress the over-fitting phenomenon in the network. The convolution and pooling parameters of convolution neural networks are detailed in Table 3.

## 4.2.2. Hyperparameter Optimization of the Proposed Network

We use PyTorch, a deep learning framework developed by an American company called Facebook, to conduct our actual experiment. To minimize our loss function, this paper uses the random gradient descent method to optimize our convolution neural network model. In the actual experiment, we choose the Adam optimizer as the final hyperparameter optimization method. The Adam optimizer combines the advantages of AdaGrad and RMSProp algorithms, and calculates the update step by step, comprehensively considering the first-order moment estimation and the second-order moment estimation of the loss function. Adam has the advantages of simple implementation, high computational efficiency, and low requirement of memory. In addition, the parameter update of this method is not affected by the scaling transformation of gradient, and it is also very suitable for the application of large-scale data and parameters in practice. Finally, it has good performance even in the case of large gradient noise. We set the learning rate of Adam's optimizer to 0.001. At the same time, the cross-entropy loss function is trained as the objective function. Reference in detail [45].

#### 4.2.3. The Effect of the Number of Training Data on the Results

As a type of convolution network, there are a large number of parameters in the sPSDAE-CNN model that need to be determined during the training process of the model. In order to improve the recognition accuracy of the network and suppress over-fitting in the system, a large amount of experimental data is needed to train the network. To study the training results of the neural network under different training samples, the number of training data of the neural network is set to 100, 200, 300, 900, 1500, 3000, 6000, 12,000, 15,000, and 20,000 training samples to study the performance of sPSDAE-CNN. In deep learning, there are balanced and unbalanced data collections. In Table 2, our data is fully balanced, so accuracy can still be used to evaluate the algorithm.

Because the data set is completely balanced, the data samples of UAVs under each fault condition are the same. In the actual experiment, the first three datasets do not use the sliding window method to enhance and expand the data. To reduce the influence of the random initial values of the neural network on the training results of the network, 30 repeated experiments were performed on each sample to calculate the average value. The paper uses AMD Ryzen<sup>™</sup> 5 4600H processor, NVIDIA GTX1650 graphics card, and 16GB of memory. The test data collection is tested using DataSet D in Table 3, and the test results are shown in Figure 13.



Figure 13. Diagnostic results for different data volumes.

In Figure 13, it is clear that the accuracy of the test dataset increases significantly as the training data goes from a smaller number of samples to a more significant number of samples. When the training data increases from 100 to 300, the accuracy of the test set data increases by about 20%. With the increase of training data, the accuracy of the neural network gradually approaches and converges to 100%. When the training data is increased from 100 to 300, the accuracy of the neural network gradually approaches and converges to 100%. When the training data is increased from 100 to 300, the accuracy of the neural network approaches and converges to 100%, and the standard deviation converges to 0. Secondly, we can observe that the average time of a signal diagnosed by the training model of the neural network is 4 ms, which meets the requirements of test data. By comparing the training time of different test

sets, we can find that the increase of the number of training data has little effect on the test time. In Figure 14, the points of different colors represent that the blades of the UAV are in different fault states. In the beginning, due to the small number of training samples, it is not easy to segment the characteristics between different types of data. With the increase of the number of training samples of the model, the data in the same fault situation begin polymerization, and the characteristics of different types of fault data become easier to segment. At the same time, this shows that by using the data enhancement method to enhance the original data collected by UAV, we can greatly increase the data scale and data diversity of neural network training samples, which can further improve the generalization ability of the model. Therefore, in subsequent experiments, this paper selected as many as 20,000 samples as possible for training.



**Figure 14.** Visualizing test samples from the last hidden fully connected layer with t-SNE under different training data numbers.

In the subsequent model training, we choose 20,000 training samples. The parameters in the neural network model are determined through the training set, and then we use t-SNE visualization to make the t-SNE diagram of the neural network of each layer, as shown in Figure 15. It can be seen from the figure that the separability of different features in the unprocessed original data is very poor. After successively passing through each layer of the neural network, different features in the data begin to separate. In the last layer of the neural network, we can clearly see that different types of features in the data have been completely separated. Finally, different types of faults of UAV are diagnosed through the softmax layer.

In order to evaluate the accuracy of the model for different types of fault diagnosis, we introduce the Confusion Matrix. The Confusion Matrix can evaluate the performance of the classification model by counting the number of correct and wrong classifications. The Confusion Matrix of the model is shown in Figure 16. It can be seen from the figure that the accuracy of unmanned fault diagnosis of different types of four rotors remains basically

the same, At the same time, the final model can be obtained through the Confusion Matrix, and the accuracy of fault diagnosis for four rotor UAV can reach about 98%, which can meet the needs of our actual projects and experiments.



Figure 15. t-SNE Visualization of each layer of neural network.



Figure 16. Confusion Matrix of the proposed model.

#### 4.2.4. Training Speed of sPSDAE-CNN

Because this paper adds a stack denoising autoencoder in the front part of the neural network, it will not only improve the performance of the neural network, but also increase the time and cost of model training. Pruning operation is proposed for the stack denoising autoencoder, which not only introduces the noise reduction performance of stack denoising autoencoder, but also reduces the time cost of neural network training as much as possible. In the training of the model, we can find that under the same amount of data, the training speed of the neural network with pruning operation is basically the same as that without stack noise reduction encoder, but its training speed is much better than that without pruning operation. The specific network training speed is shown in Figure 17.



**Figure 17.** Comparison of training time required by different neural network models under different data scales.

#### 4.2.5. Performance under Different Noise Interferences

Taking the collected UAV data as the original data, the drone may be disturbed by various signals during the execution of relevant tasks, so as to introduce noise signals to the data of sensors of the drone. It is impossible to obtain all the noisy data through experiments, so Gaussian white noise is artificially added to the original collected data to simulate the noise interference signals that may appear in the actual drone, and the signals with different signal-to-noise ratios are obtained. SNR is defined as follows (8):

$$SNR_{db} = 10\log_{10}(\frac{P_{signal}}{P_{noise}}),\tag{8}$$

where  $P_{noise}$  and  $P_{signal}$  represent the energy of signal and noise, respectively. It can be seen in Figure 18 that the UAV data collected in the laboratory environment is ideal and contains relatively little noise. In order to simulate the flight data under different interference in the actual flight environment, Gauss white noise of different degrees is added to the data, because Gauss white noise is the most common noise signal in nature. Therefore, we obtain aircraft data with different sizes of Gaussian white noise, that is, data with different signal-to-noise ratios. Finally, it can be seen from Figure 18 that the data after adding Gaussian white noise is closer to the UAV data in the actual flight environment. We evaluate the performance of the proposed model in different noise environments by studying the performance of the algorithm model with a signal-to-noise ratio of -4 dB to 10 dB.

In order to verify the efficiency of our proposed algorithm, we use the same test data to test the performance of CNN, SVM ,and SDAE, as shown in Figure 19:

As can be seen from Figure 19 that, firstly, because the sparse pruning noise reduction autoencoding convolutional neural network proposed by us has good noise reduction characteristics, it can be clearly seen that when the noise in the signal is considerable, the fault diagnosis effect of the model is obviously better than several other intelligent fault diagnosis methods. Secondly, due to the introduction of sparse pruning operation in the stack noise reduction autoencoder, this operation can improve the computational efficiency of the network to a certain extent, and make our proposed model still have very good performance in the case of low noise.



**Figure 18.** Original UAV data, noise data to be added, and final synthesized data containing Gaussian white noise.



Figure 19. Comparison of accuracy of different fault diagnosis algorithms under different noise levels.

The experimental conditions are mainly flight experiments outdoors, and fault diagnosis is carried out by using the UAV data collected and saved by pixhawk4 flight control board. We artificially add different degrees of Gaussian white noise signals to the collected data to simulate the actual noise signals, and obtain the experimental data with different degrees of noise. SVM is used for classification. The  $20 \times 20$  gray image obtained from UAV data processing is transformed into a feature vector with a length of 400. Multi classification support vector machine is used, in which the radial basis function (RBF) kernel function is selected as the kernel function. Gamma is set to the best value of 0.001 through many experiments. Finally, the experimental results in this paper are obtained. Secondly, in the use of convolutional neural network, we directly use the convolutional neural network proposed in the article, add a convolution layer to the previous layer of convolutional neural network to extract the data features, reduce the dimension, convert the original 20  $\times$  20 graphics into 10  $\times$  10 gray images, and carry out subsequent operations and classification. When DEA is used for recognition and classification, DAE is used for dimensionality reduction and optimization of the original data. BCE error is used in the training process. Adam optimizer is used, and then convolutional neural network is used for data feature extraction and classification.

## 5. Conclusions

In this paper, we adopt a new intelligent fault diagnosis method based on sPSDAE-CNN. Through a matrix transformation of the data collected from the UAV flight experiment, the one-dimensional time-series signal is transformed into two-dimensional gray image data, which expands the dimension of the sample and enhances the processing ability of the DL model. Secondly, by introducing a sparse pruning stack noise reduction autoencoder, the accuracy of a fault diagnosis algorithm in a high noise environment can be improved, and the input dimension of CNN data can also be reduced. In addition, pruning operation is used to reduce the complexity of the encoder, which can make the encoder converge quickly when minimizing the loss function. The combination of sPSDAE and the convolutional neural network can greatly improve the robustness and generalization ability of the fault diagnosis model. In order to verify the effectiveness of the model, this paper chooses CNN, SVM, and SDAE to compare. The experimental results show that under the condition of normal experimental data, sPSDAE-CNN has good results compared with other algorithms, but when the noise signal in the signal gradually begins to increase, the performance of other algorithms decreases significantly. Among them, when the signal-to-noise ratio reaches -4 dB, sPSDAE-CNN still has an accuracy of about 90%, the accuracy of the other three algorithms decreased to less than 80%, and SVM is less than 60%. Therefore, the fault diagnosis sPSDAE-CNN algorithm used in this paper can be used as a fault diagnosis method of four-rotor UAV in an actual high noise environment.

The method proposed in the article first converts a one-dimensional time-domain signal into a two-dimensional grayscale image, which expands the dimensionality of the data and can improve the ability of subsequent algorithms to extract features from the data. Secondly, the method of resampling was used to enhance the flight data of the quad-rotor UAV, which greatly improved the problem of the insufficient data set. Finally, the sparse pruning noise reduction autoencoder is introduced to perform noise reduction, dimensionality reduction, and feature extraction on the data. After processing, the noise in the original data can be filtered to a large extent, and the pruning operation can also improve the model—the calculation efficiency and noise reduction performance. All the data used in the article are balanced data sets. In the actual environment, it is impossible for all data to be unbalanced data sets. In the follow-up research, the application scope of unbalanced data sets will be further expanded.

In addition, in this paper, balanced data sets are used, but during the actual UAV mission, the data we collect can not be completely balanced data sets. Therefore, in future research, we will improve and expand the application scope of the algorithm based on the performance of sPSDAE-CNN on unbalanced data sets.

Secondly, the data used in this paper are all offline data collected at the end of the UAV flight. At present, it is not possible to collect the data of four-rotor UAV in real-time in the actual flight process to realize fault diagnosis. In future research, we can try to diagnose the fault of UAV in real-time and online with the algorithm used in this paper; this problem needs to be further studied and solved.

**Author Contributions:** Conceptualization, C.W.; methodology, P.Y.; software, C.W.; validation, C.W.; formal analysis, C.W.; investigation, H.G.; resources, P.Y.; data curation, H.G.; writing—original draft preparation, C.W. and P.Y.; writing—review and editing, P.Y.; visualization, P.L.; supervision, P.L.; project administration, P.Y.; funding acquisition, P.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by Key Laboratories for National Defense Science and Technology (6142605200402), the Aeronautical Science Foundation of China (20200007018001), the National Natural Science Foundation of China (61922042), the Aero Engine Corporation of China Industry University Research Cooperation Project (HFZL2020CXY011), and the Research Fund of State Key Laboratory of Mechanics and Control of Mechanical Structures (Nanjing University of Aeronautics and Astronautics) MCMS-I-0121G03. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

#### References

- Bateman, F.; Noura, H.; Ouladsine, M. Fault diagnosis and fault-tolerant control strategy for the aerosonde UAV. *IEEE Trans.* Aerosp. Electron. Syst. 2011, 47, 2119–2137. [CrossRef]
- 2. Yu, X.; Jiang, J. A survey of fault-tolerant controllers based on safety-related issues. Annu. Rev. Control 2015, 39, 46–57. [CrossRef]
- Haupt, F.; Berding, G.; Namazian, A.; Wilke, F.; Böker, A.; Merseburger, A.; Geworski, L.; Kuczyk, M.A.; Bengel, F.M.; Peters, I. Expert system for bone scan interpretation improves progression assessment in bone metastatic prostate cancer. *Adv. Ther.* 2015, 34, 986–994. [CrossRef] [PubMed]
- 4. Zhu, Y.; Geng, L. Research on SDG fault diagnosis of ocean shipping boiler system based on fuzzy granular computing under data fusion. *Pol. Marit. Res.* 2018, 25, 92–97. [CrossRef]
- Hu, G.; Phan, H.; Ouache, R.; Gandhi, H.; Hewage, K.; Sadiq, R. Fuzzy fault tree analysis of hydraulic fracturing flowback water storage failure. J. Nat. Gas Sci. Eng. 2019, 72, 103039. [CrossRef]
- Liu, D.; Gu, X.; Li, H. A complete analytic model for fault diagnosis of power systems. *Proc. Chin. Soc. Electr. Eng.* 2011, *31*, 85–92.
   Chen, X.; Qi, X.; Wang, Z.; Cui, C.; Wu, B.; Yang, Y. Fault diagnosis of rolling bearing using marine predators algorithm-based
- support vector machine and topology learning and out-of-sample embedding. *Measurement* 2021, *176*, 109116. [CrossRef]
  Chen, F.; Wyer, R.S., Jr. The effects of affect, processing goals and temporal distance on information processing: Qualifications on temporal construal theory. J. Consum. Psychol. 2015, *25*, 326–332. [CrossRef]
- 9. Xiao, Y.; Li, C.; Song, L.; Yang, J.; Su, J. A multidimensional information fusion-based matching decision method for manufacturing service resource. *IEEE Access* 2021, *9*, 39839–39851. [CrossRef]
- 10. Chady, T.; Sikora, R.; Misztal, L.; Grochowalska, B.; Grzywacz, B.; Szydłowski, M.; Waszczuk, P.; Szwagiel, M. The application of rough sets theory to design of weld defect classifiers. *J. Nondestruct. Eval.* **2017**, *36*, 40. [CrossRef]
- 11. Esteki, M.; Farajmand, B.; Kolahderazi, Y.; Simal-Gandara, J. Chromatographic fingerprinting with multivariate data analysis for detection and quantification of apricot kernel in almond powder. *Food Anal. Methods* **2017**, *10*, 3312–3320. [CrossRef]
- Jiang, Y.; Zhiyao, Z.; Haoxiang, L.; Quan, Q. Fault detection and identification for quadrotor based on airframe vibration signals: A data-driven method. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 6356–6361.
- 13. Tang, S.; Yuan, S.; Zhu, Y. Convolutional neural network in intelligent fault diagnosis toward rotatory machinery. *IEEE Access* **2020**, *8*, 86510–86519. [CrossRef]
- 14. Tao, H.; Wang, P.; Chen, Y.; Stojanovic, V.; Yang, H. An unsupervised fault diagnosis method for rolling bearing using STFT and generative neural networks. *J. Frankl. Inst.* 2020, *357*, 7286–7307. [CrossRef]
- 15. Glowacz, A. Fault diagnosis of electric impact drills using thermal imaging. *Measurement* 2021, 171, 108815. [CrossRef]
- 16. Polat, K. The fault diagnosis based on deep long short-term memory model from the vibration signals in the computer numerical control machines. *J. Inst. Electron. Comput.* **2020**, *2*, 72–92. [CrossRef]
- 17. Xiong, R.; Sun, W.; Yu, Q.; Sun, F. Research progress, challenges and prospects of fault diagnosis on battery system of electric vehicles. *Appl. Energy* 2020, *2*, 72–92. [CrossRef]
- Hu, X.; Zhang, K.; Liu, K.; Lin, X.; Dey, S.; Onori, S. Advanced fault diagnosis for lithium-ion battery systems: A review of fault mechanisms, fault features, and diagnosis procedures. *IEEE Ind. Electron. Mag.* 2020, 14, 65–91. [CrossRef]
- 19. Azamfar, M.; Singh, J.; Bravo-Imaz, I.; Lee, J. Multisensor data fusion for gearbox fault diagnosis using 2-D convolutional neural network and motor current signature analysis. *Mech. Syst. Signal Process.* **2020**, *144*, 106861. [CrossRef]
- 20. He, Z.; Shao, H.; Wang, P.; Lin, J.J.; Cheng, J.; Yang, Y. Deep transfer multi-wavelet auto-encoder for intelligent fault diagnosis of gearbox with few target training samples. *Knowl.-Based Syst.* **2021**, *191*, 105313. [CrossRef]
- 21. Chen, H.; Jiang, B.; Ding, S.X.; Huang, B. Data-driven fault diagnosis for traction systems in high-speed trains: A survey, challenges, and perspectives. *IEEE Trans. Intell. Transp. Syst.* 2020, 64, 1–3. [CrossRef]
- 22. Chen, H.; Jiang, B. A review of fault detection and diagnosis for the traction system in high-speed trains. *Trans. Intell. Transp. Syst.* **2019**, *21*, 450–465. [CrossRef]
- Huang, D.; Li, S.; Qin, N.; Zhang, Y. Fault diagnosis of high-speed train bogie based on the improved-CEEMDAN and 1-D CNN algorithms. *IEEE Trans. Instrum. Meas.* 2021, 70, 1–11.
- 24. Iannace, G.; Ciaburro, G.; Trematerra, A. Fault diagnosis for UAV blades using artificial neural network. *Robotics* **2019**, *8*, 59. [CrossRef]
- Liu, W.; Chen, Z.; Zheng, M. An Audio-Based Fault Diagnosis Method for Quadrotors Using Convolutional Neural Network and Transfer Learning. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 1367–1372.
- 26. Zhang, W.; Peng, G.; Li, C.; Chen, Y.; Zhang, Z. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* 2015, *17*, 425. [CrossRef] [PubMed]

- 27. Che, C.; Wang, H.; Ni, X.; Fu, Q. Intelligent fault diagnosis method of rolling bearing based on stacked denoising autoencoder and convolutional neural network. *Ind. Lubr. Tribol.* 2020, 72, 947–953. [CrossRef]
- 28. Cheng, Y.; Lin, M.; Wu, J.; Zhu, H.; Shao, X. Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural network. *Knowl.-Based Syst.* **2021**, *216*, 106796. [CrossRef]
- Okada, K.F.Á.; de Morais, A.S.; Oliveira-Lopes, L.C.; Ribeiro, L. Neuroadaptive Observer-Based Fault-Diagnosis and Fault-Tolerant Control for Quadrotor UAV. In Proceedings of the 2021 14th IEEE International Conference on Industry Applications, São Paulo, Brazil, 15–18 August 2021; pp. 285–292.
- 30. Guo, J.; Qi, J.; Wu, C. Robust fault diagnosis and fault-tolerant control for nonlinear quadrotor unmanned aerial vehicle system with unknown actuator faults. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 17298814211002734. [CrossRef]
- 31. Patan, M.G.; Caliskan, F. Sensor fault–tolerant control of a quadrotor unmanned aerial vehicle. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* 2021. [CrossRef]
- 32. Wang, B.; Huang, P.; Zhang, W. A Robust Fault-Tolerant Control for Quadrotor Helicopters against Sensor Faults and External Disturbances. *Complexity* 2021, 2021, 6672812. [CrossRef]
- 33. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. arXiv 2014, arXiv:1404.2188.
- 34. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. Nature 2015, 521, 436–444. [CrossRef] [PubMed]
- 35. Sibi, P.; Jones, S.A.; Siddarth, P. Analysis of different activation functions using back propagation neural networks. *J. Theor. Appl. Inf. Technol.* **2013**, *47*, 1264–1268.
- 36. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Int. Conf. Mach. Learn.* **2015**, *37*, 448–456.
- 37. Lu, C.; Wang, Z.Y.; Qin, W.L.; Ma, J. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Process.* **2017**, *130*, 377–388. [CrossRef]
- 38. Heys, J.J.; Holyoak, N.; Calleja, A.M.; Belohlavek, M.; Chaliki, H.P. Revisiting the simplified Bernoulli equation. *Open Biomed. Eng. J.* **2010**, *4*, 123. [CrossRef]
- 39. Werbos, P.J. Backpropagation through time: What it does and how to do it. Proc. IEEE 1990, 78, 1550–1560. [CrossRef]
- 40. 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]
- Kiranyaz, S.; Ince, T.; Abdeljaber, O.; Avci, O.; Gabbouj, M. 1-d convolutional neural networks for signal processing applications. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 8360–8364.
- 42. Avci, O.; Abdeljaber, O.; Kiranyaz, S.; Inman, D. Structural damage detection in real time: Implementation of 1D convolutional neural networks for SHM applications. *Struct. Health Monit. Damage Detect.* **2017**, *7*, 49–54.
- 43. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van de Walle, R.; Van Hoecke, S. Convolutional neural network based fault detection for rotating machinery. *J. Sound Vib.* **2016**, 377, 331–345. [CrossRef]
- Tan, Y.; Wang, J. A support vector machine with a hybrid kernel and minimal Vapnik-Chervonenkis dimension. *IEEE Trans. Knowl. Data Eng.* 2004, 16, 385–395.
- 45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. arXiv 2014, arXiv:1412.6980.