

## Article

# Deep Learning with LPC and Wavelet Algorithms for Driving Fault Diagnosis

Cihun-Siyong Alex Gong <sup>1,2,3,\*</sup> , Chih-Hui Simon Su <sup>1</sup>, Yuan-En Liu <sup>1</sup>, De-Yu Guu <sup>1</sup> and Yu-Hua Chen <sup>1</sup>

<sup>1</sup> Department of Electrical Engineering, School of Electrical and Computer Engineering, College of Engineering, Chang Gung University, Taoyuan 33302, Taiwan

<sup>2</sup> Portable Energy System Group, Green Technology Research Center, College of Engineering, Chang Gung University, Taoyuan 33302, Taiwan

<sup>3</sup> Department of Neurosurgery, Chang Gung Memorial Hospital, Linkou, Taoyuan 33302, Taiwan

\* Correspondence: alexgong@mail.cgu.edu.tw

**Abstract:** Vehicle fault detection and diagnosis (VFDD) along with predictive maintenance (PdM) are indispensable for early diagnosis in order to prevent severe accidents due to mechanical malfunction in urban environments. This paper proposes an early voiceprint driving fault identification system using machine learning algorithms for classification. Previous studies have examined driving fault identification, but less attention has focused on using voiceprint features to locate corresponding faults. This research uses 43 different common vehicle mechanical malfunction condition voiceprint signals to construct the dataset. These datasets were filtered by linear predictive coefficient (LPC) and wavelet transform (WT). After the original voiceprint fault sounds were filtered and obtained the main fault characteristics, the deep neural network (DNN), convolutional neural network (CNN), and long short-term memory (LSTM) architectures are used for identification. The experimental results show that the accuracy of the CNN algorithm is the best for the LPC dataset. In addition, for the wavelet dataset, DNN has the best performance in terms of identification performance and training time. After cross-comparison of experimental results, the wavelet algorithm combined with DNN can improve the identification accuracy by up to 16.57% compared with other deep learning algorithms and reduce the model training time by up to 21.5% compared with other algorithms. Realizing the cross-comparison of recognition results through various machine learning methods, it is possible for the vehicle to proactively remind the driver of the real-time potential hazard of vehicle machinery failure.

**Keywords:** vehicle early fault diagnosis; machine learning (ML); linear predictive coefficient (LPC); wavelet transform (WT); convolutional neural network (CNN); deep neural network (DNN); long short-term memory (LSTM)



**Citation:** Gong, C.-S.A.; Su, C.-H.S.; Liu, Y.-E.; Guu, D.-Y.; Chen, Y.-H. Deep Learning with LPC and Wavelet Algorithms for Driving Fault Diagnosis. *Sensors* **2022**, *22*, 7072. <https://doi.org/10.3390/s22187072>

Academic Editors: Jose Alfonso Antonino-Daviu, Francesc Pozo, Steven Chatterton and Mohammad N. Noori

Received: 3 September 2022

Accepted: 15 September 2022

Published: 19 September 2022

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



**Copyright:** © 2022 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/>).

## 1. Introduction

Issues such as “Metaverse”, “Big Data”, “Artificial Intelligence (AI)”, and “Digital Transformation” are in full swing [1,2], and the most critical point is the use of data acquisition (DAQ), data analysis, and machine learning, etc., to realize the integration of digitalization and smart manufacture of the system. With the popularization of 5G communication and the rapid development of Industry 4.0, the integration of technologies such as AI, Internet of Things (IoT), and cloud computing has become a very important development key in the field [3–5]. Recently, the scientific and technological circles are quite looking forward to realizing the integration of virtual and real (Digital Twin), thereby leading technology to another metaverse of a new digital world. These ever-changing cross-domain integrations show that the application of AI is leading the development of future technology and has penetrated into every corner.

The development of Industry 4.0 has attracted increased attention to fault diagnosis in recent years. For equipment automation, effective fault diagnosis can save time by allowing

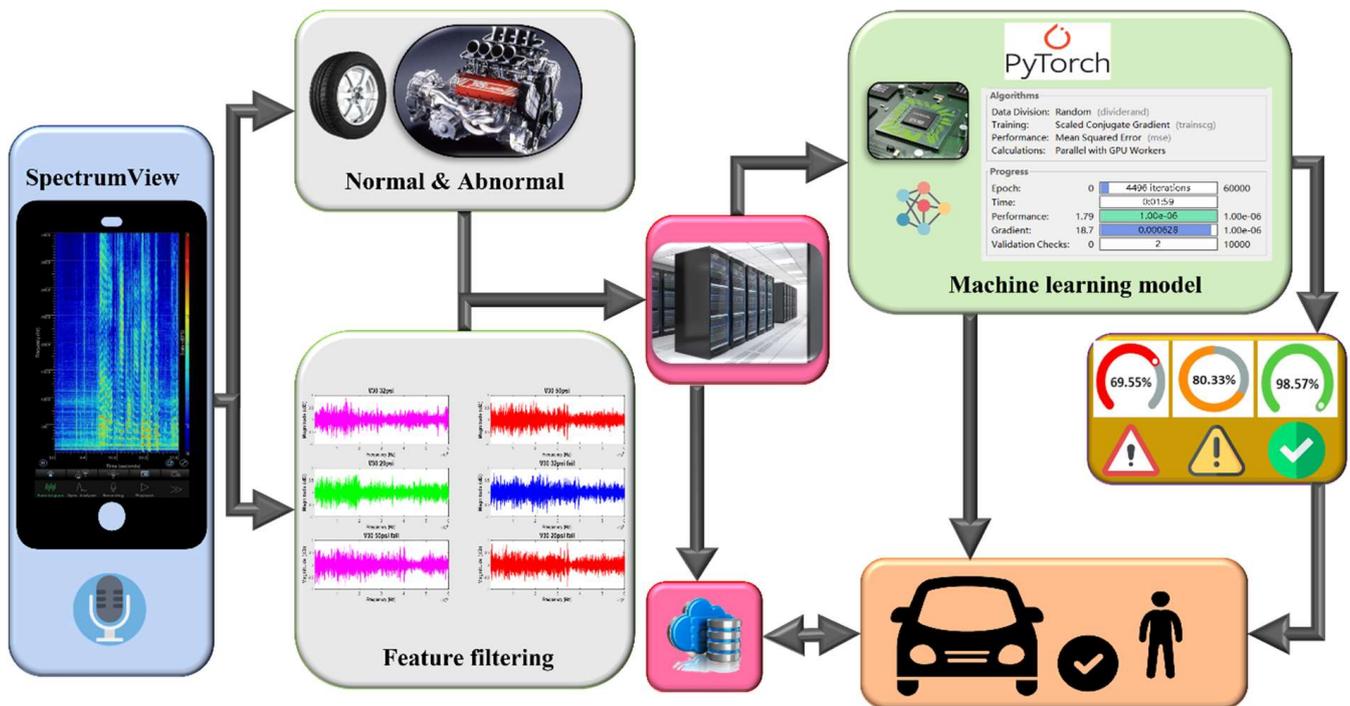
for timely remedial intervention, thus preventing potentially dangerous malfunctions [6]. Traditional fault diagnosis requires considerable personal expertise and experience, making it inefficient and costly. Expert systems present a potential solution [7]. Recently, the popularization of Internet of Things technologies has driven interest in the “Internet of Vehicles (IoV)”, in which multiple sensors are deployed on cars or other vehicles to collect timely status data [8]. Driverless vehicle applications simultaneously show to be increasingly reliable, with Tesla’s assisted driving system allowing for automatic lane changing based on real-time road conditions [9].

However, these developments are still subject to non-human-caused hazards, such as transmission failures and faulty tire conditions. In the past, such problems required passive prevention. Vehicles can fail without warning in any driving situation. Today’s vehicle diagnosis artificial intelligence technology adopts integrated signal feature extraction with various algorithm prediction models such as machine or deep learning and then uses a computer to perform rapid big data calculation and automatically classify its characteristic attributes to achieve fault prediction, identification, and diagnosis. The use of a variety of sensors and integration with the 5G communication V2X specification greatly improves the predictive ability of auto self-diagnosis. For example, in regular vehicle maintenance, the technician visually inspects the condition of the tires and transmission system using measurement devices. If the car self-diagnosis system is introduced, when various sensors of the vehicle collect the danger signal of failure or impending failure, it can immediately send a warning to the driver and the surrounding vehicles. In this way, the human resources required for car maintenance and repair will be greatly reduced, and road traffic safety will also be greatly improved.

However, Zhang’s proposed car networking structure emphasizes fault prediction and maintenance [10]. Such vehicle systems rely heavily on the use of sensors to collect current fault status data and provide a timely determination of the fault location. Such data can be transmitted to a nearby service station through the cloud for initial diagnosis, thus greatly reducing repair times, as shown in Figure 1. Moreover, advances in car sharing technology [11] now allow for warnings to be transmitted to neighboring vehicles. To date, many fault identification solutions have been proposed, such as the Distributed Fiber-Optic Acoustic Sensing (DAS) system proposed by Li et al. to detect vehicle vibrations on the road in order to identify vehicle model and know its amount [12], or to identify specific distortions and locate faulty turn phases [13]. The performance of AI approaches for diagnostics [4] relies on data quality and quantity. Currently, the two main directions for data sources are audio and video. In 2018, the BMW began to introduce AI-based image recognition techniques to inspect automotive sheet metal quality. Meanwhile, voice recognition technology has been widely deployed, with applications such as Siri and Google voice assistants continuously collecting longitudinal data to achieve high degrees of recognition accuracy.

Currently, no mature technology exists for vehicle voiceprint recognition, and the present research focuses on the development of voice recognition applications for use in vehicles. Feature extraction is the key aspect in machine learning, and good features selection can significantly improve model training accuracy. When an object vibrates freely, it has a fixed specific frequency and mode. As long as the rigidity, structure, and shape of these objects are fixed, the natural frequency of the vibration is fixed. As a result, it will not change with time and the external force environment. In this study, sensors are used to collect signals and analyze the early fault characteristics of rotating machinery to identify the signals. Typically, signals are obtained from non-stationary and non-linear machines with a certain degree of noise [14], which may obscure important fault features. Such signals, such as cancer, cannot be clearly identified initially and are often obscured by complex background environmental noise. If the weak abnormal signal in the early stage of the fault cannot be captured, the opportunity for early maintenance will be missed, resulting in vehicle accidents. In this study, a more accurate sensor is used to obtain the abnormal signal generated by the initial failure of the vehicle. The non-contact measurement method of a

MEMS microphone array is adopted, because the acoustic microphone has high resolution in the middle and high frequency bands. The fault characteristics can be presented by the mid-to-high frequency signal in the early stage of the fault [15,16], and consequently, the acoustic sensing technology is very suitable for the early abnormal detection of the rotating mechanical system.



**Figure 1.** The basic procedure of an early driving fault diagnosis system and machine learning training process.

Instead of using the MEMS microphone array, addressing these problems requires the use of appropriate filtering methods, and methods such as the Mel-scale frequency cepstral coefficient (MFCC) [17,18], Fast Fourier transform (FFT) [19,20], order-tracking technology [16] and wavelet transform [21] have been applied to machinery fault diagnosis. Linear prediction coefficients (LPC) have been applied to many modern speech processing systems for applications including coding, synthesis, analysis and recognition [22]; the initial model is constructed using historical data, and new data testing and verification can be used to predict the associated outcomes of audio signal data. Previous studies have used LPC to achieve perfect fault diagnosis performance [23], and compared with the above-mentioned filtering algorithm, LPC uses less resources to achieve high-resolution spectra. The development of artificial intelligence (AI) and the Internet of things (IoT) [4,24] has raised new possibilities. For example, failure detection of vehicle suspension systems [25] uses AI to achieve early prediction, thus reducing the occurrence of vehicle accidents.

Machine learning algorithms can be used to perform large-scale data analysis. Support vector machine (SVM) was first used for fault diagnosis in the late 1990s [26]. Artificial neural networks (ANN) are among the most widely used methods for fault diagnosis and have been used for mechanical fault prediction [27] and the later development of multilayer neural networks [28]. The improvement of hardware capabilities further drove the application of neural networks. Increasing the number of neurons deepens the hidden layer, thus improving the recognition rate. Deep neural networks (DNN) are still in the development stage for fault identification applications, and many challenges still need to be addressed. For example, very large amounts of data can significantly impair processing efficiency [29]. Even with strong hardware support, data processing presents a major challenge and is difficult to apply in practice. This article also discusses the challenges of finding a suitable

activation function to accelerate neuron convergence. Previously, CNNs were mainly used for image and facial recognition [30,31] and have rarely been used for classification in speech processing. In this experiment, the spectrum dataset is classified using a CNN model for comparison with different types of classifiers, such as DNNs. The remainder of this article is as follows. Section 2 discusses the theoretical background, including LPC, Wavelet, DNN, CNN, and LSTM algorithms. Section 3 introduces the experimental framework, collects sound signals, and converts them into voiceprint characteristic spectra to build a dataset. Section 4 introduces the test results of ML methods on the dataset. Section 5 draws conclusions and presents directions for future work.

## 2. Method Theory

### 2.1. Linear Predictive Coding Method

Linear prediction coefficient (LPC) is one of the most effective speech analysis techniques and is widely used in speech recognition and audio compression [32], as illustrated in Figure 2, and  $G$  is gain value. LPC can provide accurate speech parameter prediction, making it well suited for modeling the transfer characteristics of sound sources. Good analytical performance is also observed in the extraction of noise characteristics of the mechanical transmission system when using LPC. The main theory is that the input  $x(k)$  of a linear discrete-time system is a linear weighted combination of the input samples and the output of previous samples. The following function can be written:

$$x(k) = \sum_{i=1}^p a_i \cdot x(k-i) + e(k) \quad (1)$$

where integer  $k$  is the time index,  $a_i$  is defined as the linear prediction coefficient, and  $p$  represents the past coefficient. Given a prediction signal  $x(k)$ , the number of prediction errors  $e(k)$  is given by:

$$e(k) = x(k) - \hat{x}(k) = x(k) - \sum_{i=1}^p a_i \cdot x(k-i) \quad (2)$$

$\hat{x}(k)$  is the prediction sample, and  $a_i$  is determined by  $e(k)$  to minimize the mean square error (MSE). The equation is:

$$a_i = E[e^2(k)] \quad (3)$$

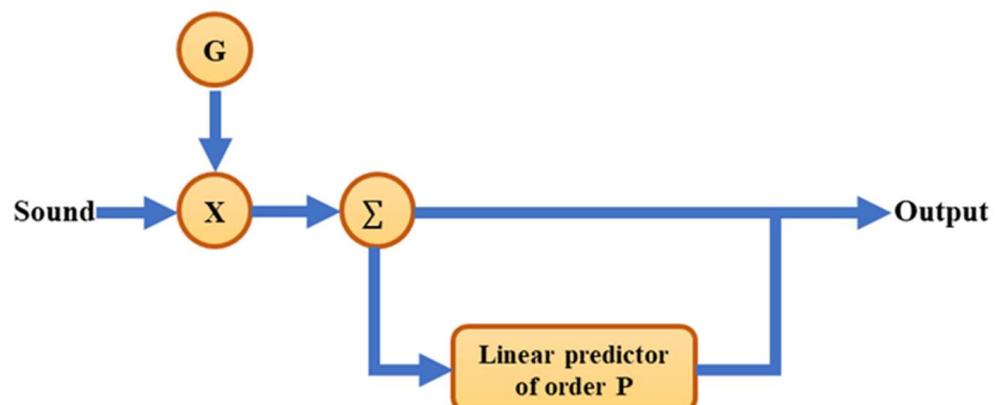


Figure 2. The spectral features filtered by LPC method.

## 2.2. Wavelet Transform (WT)

The Shannon recovery formula can assist in restoring the original analog function  $y(t)$ , where the relationship can be as follows

$$y(t) = \sum_{nh \in \mathbb{Z}} x(nh) \frac{\sin \pi(t - nh)}{\pi(t - nh)} \quad (4)$$

The continuous wavelet transform (CWT) of the time-domain signal  $y(t)$  can be expressed by the following transformation formula:

$$W_{\varnothing}x(b, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} y(t) \overline{\varnothing\left(\frac{t-b}{a}\right)} dt \quad (5)$$

If  $a = \frac{1}{2^\sigma}$  is used as the scale,  $b = \frac{k}{2^\sigma}$  is used as the translation, and both  $s$  and  $k$  are integers, in the time-scale plane, the CWT of  $y(t)$  is a value in  $(\frac{k}{2^\sigma}, \frac{1}{2^\sigma})$ , which represents the relationship between  $x(t)$  and  $\overline{\varnothing}(t)$  at that time-scale point, and is called discrete wavelet transform (DTW). This method generates a set of sparse values on the time-scale plane. The expression is as follows:

$$w_{k,s} = W_{\varnothing}x\left(\frac{k}{2^\sigma}, \frac{1}{2^\sigma}\right) = \int_{-\infty}^{\infty} y(t) \overline{\varnothing\left(\frac{t - \frac{k}{2^\sigma}}{\frac{1}{2^\sigma}}\right)} dt \quad (6)$$

With this expression, the wavelet coefficient can be represented in  $(b = \frac{k}{2^\sigma}, a = \frac{1}{2^\sigma})$ . That is the mapping to the time-domain signal  $y(t)$  under the discrete-time scale [33].

## 2.3. Deep Neural Network (DNN)

DNN is an artificial neural network used for supervised learning. Neural networks make “judgments” by simulating the operation of neurons in human brain cells. Such networks contain many computational layers, using the input layer and output layer as perceptrons, and with one or more hidden layers between them. Networks with multiple hidden layers are called deep neural networks (DNNs). Using a large number of hidden layer training data can help improve the accuracy of the weight value classification. The activation function plays an important role in neural networks. It can make neurons improve gradient descent performance through nonlinear conversion. Different activation functions can be used to improve the MLP performance [34]. Figure 3 shows MLP structure with two hidden layers.  $W^{(m)}$  is defined as weighted, which connects between the hidden layers.  $b^{(m)}$  is basis of the  $m$ th layers ( $m > 0$ ).  $a^{(m)}(x)$  indicates the previous level  $h^{(m-1)}$  and  $W^{(m)}$  are multiplied and added to  $b^{(m)}$ . The value of  $a^{(k)}(x)$  is inserted into the activation function  $g(x)$ , and the result is the output  $y_n$  [17].

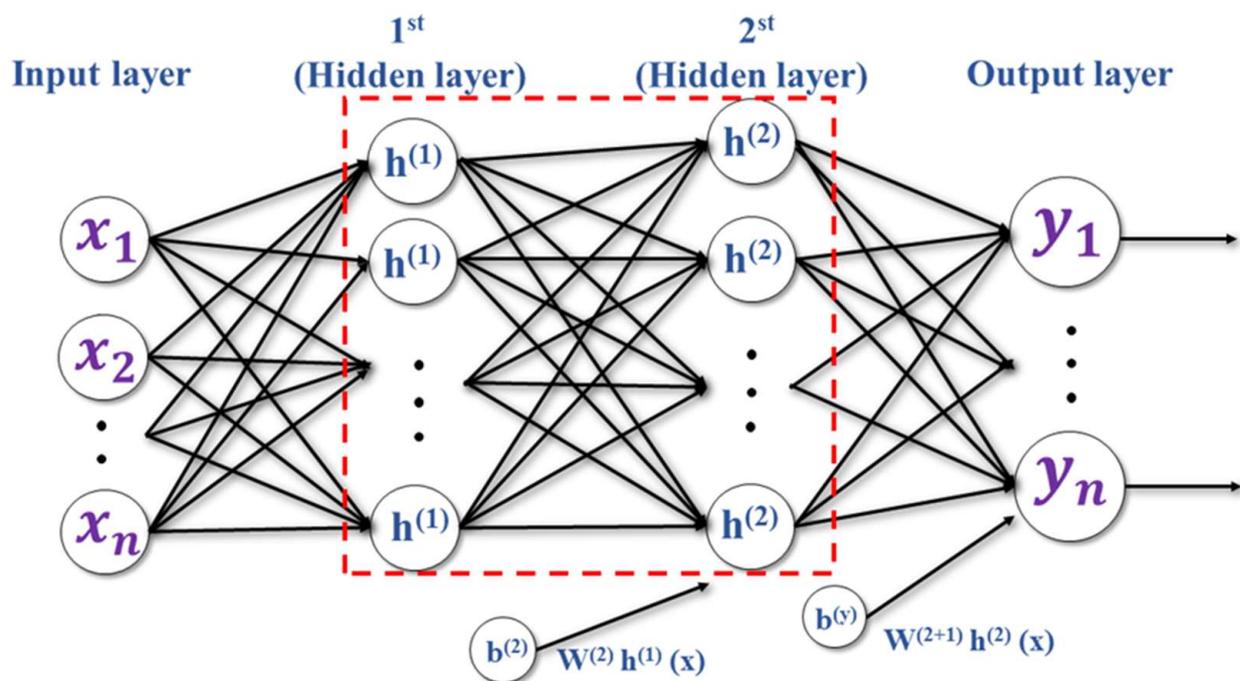
$$a^{(m)}(x) = b^{(m)} + W^{(m)}h^{(m-1)}(x) \quad (7)$$

For the  $i$ th neuron in the  $m$ th hidden layer, the concept is equated as below:

$$h^{(m)}(x)_i = g(a^{(m)}(x)_i) \quad (8)$$

The equation of the desired output layer is formulated as:

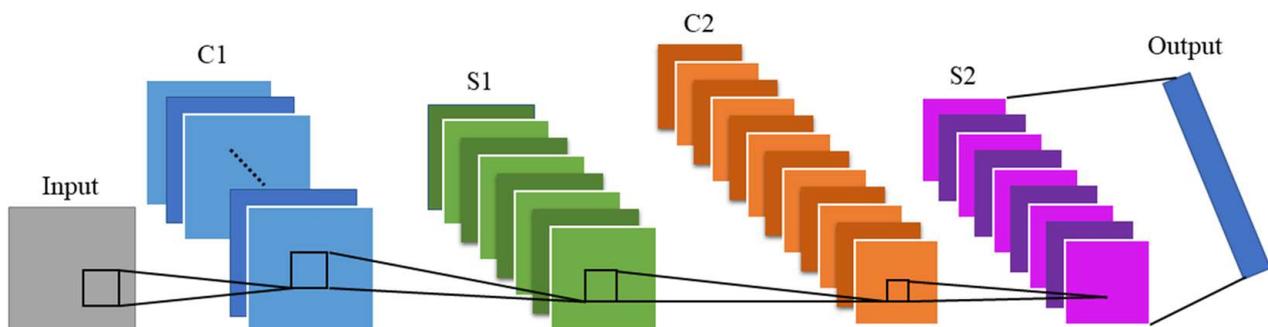
$$y_n = g(a^{(m+1)}(x)) = f(x) \quad (9)$$



**Figure 3.** General MLP neural network architecture for the automotive.

#### 2.4. Convolutional Neural Network (CNN)

The basic CNN architecture was first proposed in 1980 by Kunihiko Fukushima [35]. Its structure was inspired by the concept of simple and complex cells in the brain's visual cortex [36], as an extension of the ANN architecture. A CNN is composed of a convolution layer, a pooling layer, and a fully connected layer. The convolutional layer is a feature map obtained by applying the summing of the product of the input pixels. The pooling layer is used to reduce the feature dimensionality of the input, thus preventing overfitting. Finally, the fully connected layer flattens the features into a one-dimensional vector for classification. Some well-known CNN models, such as AlexNet [30], GoogLeNet [37], VGGNet [38], LeNet-5 [39], etc., have been widely used in image recognition. Among these, a CNN block diagram has been successfully applied for image-based fault diagnosis [40]. Figure 4 shows the CNN architecture, which is applied here to identify audio signal features in vehicles.

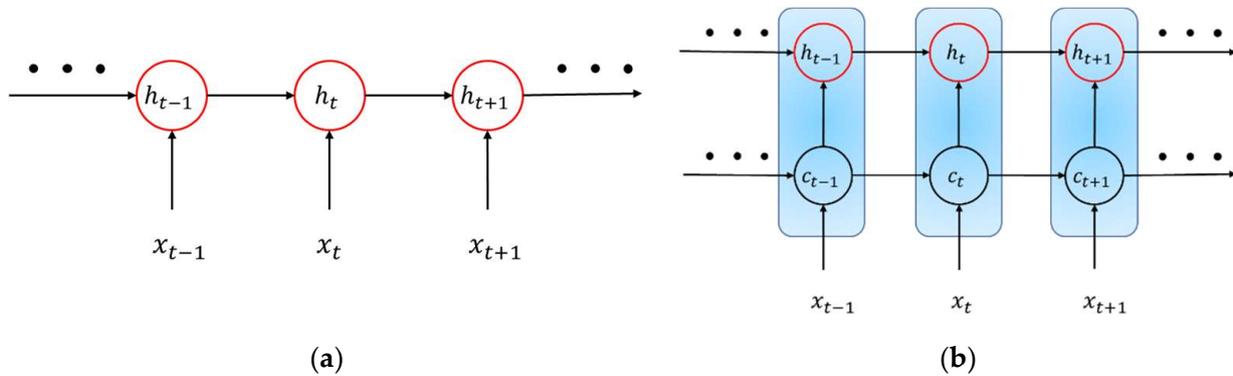


**Figure 4.** Architecture of the CNN.

#### 2.5. Long Short-Term Memory (LSTM)

In a general recurrent neural network, there is only one hidden state unit  $h_t$ , and the parameters of the hidden state unit at different times are the same, as shown in Figure 5a. This makes the recurrent neural network a long-term dependence problem that can only be sensitive to short-term input. LSTM adds a cell state unit  $c_t$  on the basis of an ordinary

recurrent neural network, which has variable connection weights at different times to solve the problem of gradient disappearance or gradient explosion in an ordinary recurrent neural network, as shown in Figure 5b. In Figure 5,  $h_t$  is the hidden state unit (short-term state unit), and  $c_t$  is the unit state unit (long-term state unit), which together constitute the LSTM architecture [41].



**Figure 5.** Architecture of RNN. (a) Normal architecture of RNN. (b) LSTM network.

Unlike general recurrent neural networks, LSTMs reference gating units. Gating is the unit learned by the neural network to control the storage, utilization, and discarding of signals. For each time  $t$ , LSTM has three gating units: input gate  $i_t$ , forget gate  $f_t$  and output gate  $o_t$ . The input of each gating unit contains the sequence information  $x_t$  at the current moment and the hidden state unit  $h_{t-1}$  at the previous moment. The actual calculation formula is:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (10)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (11)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (12)$$

Among them,  $W$  and  $U$  are the weight matrices,  $b$  is the bias vector, and  $\sigma(\cdot)$  is the startup function. It can be found that the calculation methods of the above three gating units are the same (all equivalent to a fully connected hierarchy), and only the weight matrix and the bias vector are different. The setting value range of the startup function  $\sigma(\cdot)$  is generally  $[0, 1]$ , and the commonly used startup function is the sigmoid function.

By multiplying the gating unit and the signal data element by element, the amount of information to be retained after the signal passes through the gating can be controlled. For example, when the state of the gate unit is 0, the signal will be completely discarded; when the state is 1, the signal will be fully retained; and when the state is between 0 and 1, the signal will be partially reserved.

LSTM operates by using three gating units and cell state units. Figure 6 is a schematic diagram of the gating unit and state unit in LSTM. It can be seen that the transmission of the cell state unit from  $c_{t-1}$  at the previous moment to  $c_t$  at the current moment is jointly controlled by the input gate and the forgetting gate. The input gate determines how much of the input information  $\tilde{c}_t$  is absorbed at the current moment. The forget gate determines how much of the cell state unit  $c_{t-1}$  is not forgotten at the previous moment, and the final cell state unit  $c_t$  is generated by the sum of the two gated signals. The actual formula is:

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (13)$$

$$c_t = (f_t \odot c_{t-1}) \oplus (i_t \odot \tilde{c}_t) \quad (14)$$

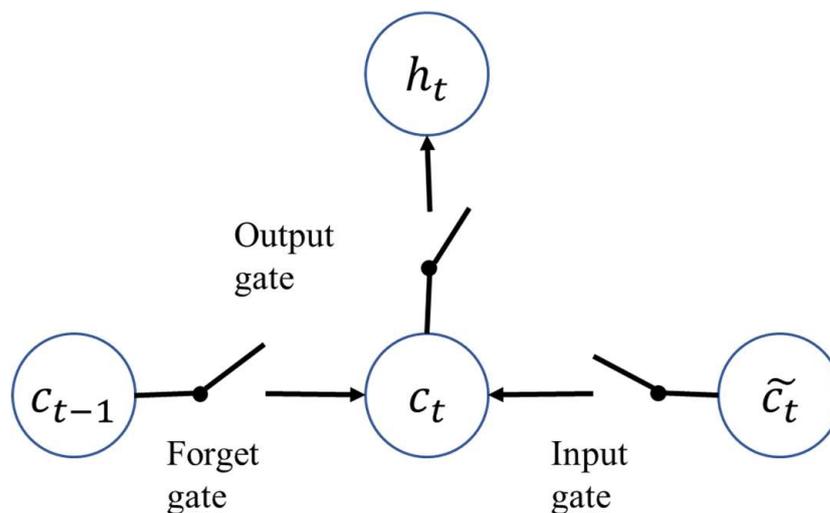


Figure 6. Schematic of LSTM cell.

Among them,  $\odot$  is the element-wise dot product operation. The hidden state unit  $h_t$  of LSTM is determined by the output gate and  $c_t$ :

$$h_t = o_t \odot \tanh(c_t) \tag{15}$$

It can be seen that in LSTM, not only the hidden state unit  $h_{t-1}$  and  $h_t$  have a relatively complex cyclic connection, but also the internal unit state unit  $c_{t-1}$  and  $c_t$ . There is also a linear self-circulating relationship between them. The linear self-loop between cell state units can be seen as sliding to process information at different times. When the gated unit is on, the past information is remembered; when the gated unit is off, the past information is discarded. On the whole, LSTM provides a path for the long-distance continuous circulation of gradients through the linear self-circulation of the gating unit and the cell state unit, which changes the propagation mode of information and gradients in the previous recurrent neural network and solves the long-term dependency problem. The complete LSTM architecture is shown in Figure 7.

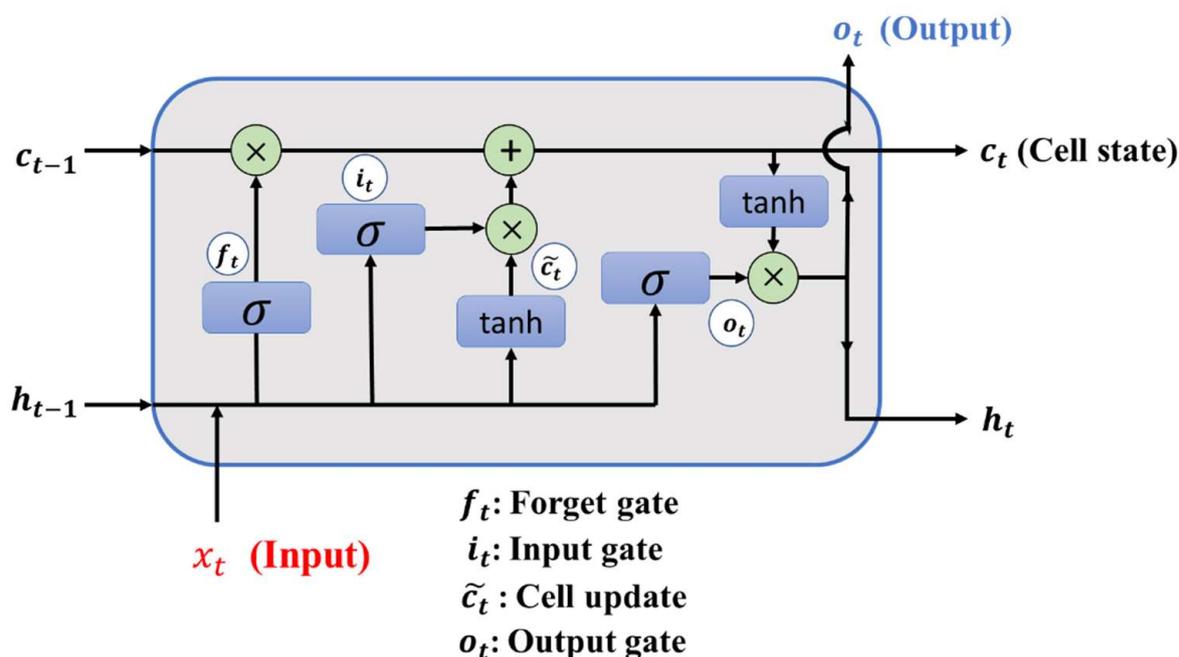
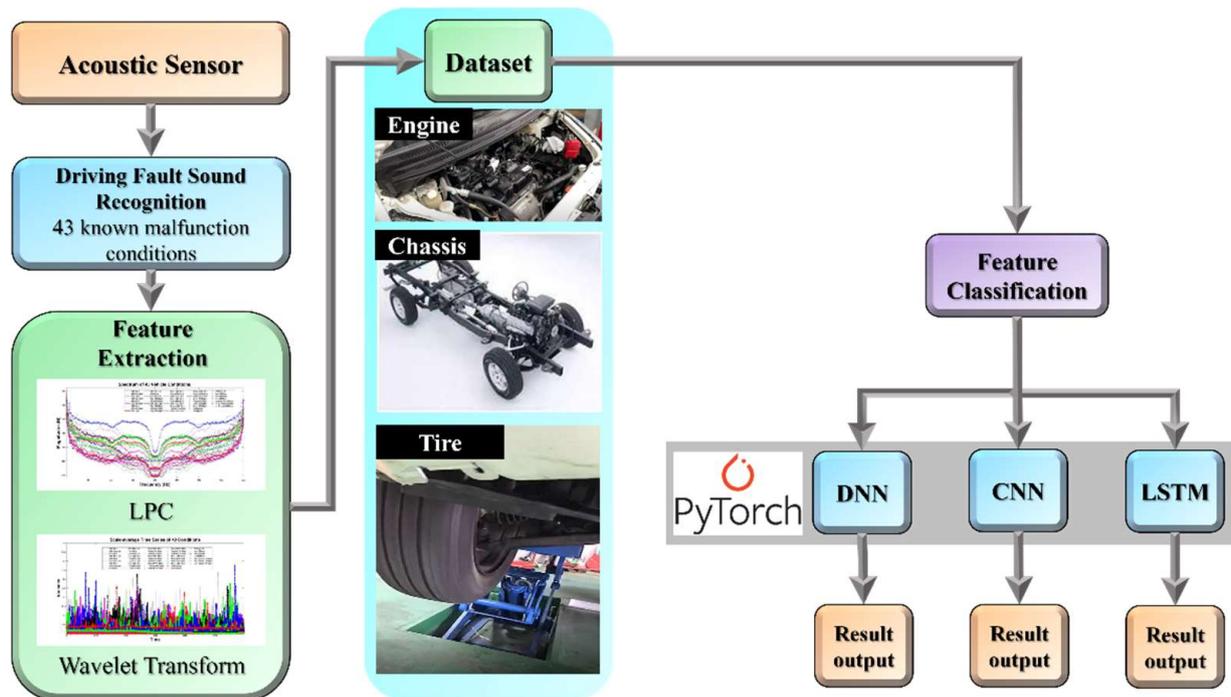


Figure 7. Complete architecture of LSTM.

### 3. Experimental Structure

This research is divided into three parts. Figure 8 shows the structure of the vehicular audio signal diagnosis experiment. The first part focuses on signal characteristic filtering. We use acoustic sensors to collect 43 vehicle fault signals. Table 1 shows the 43 fault conditions, including 18 different types for the tires, 6 types for the belt, 16 types for the chassis, and 3 types for the engine.



**Figure 8.** Systematically experimental structure with the use of classifying methodology for a driving fault acoustic signal system.

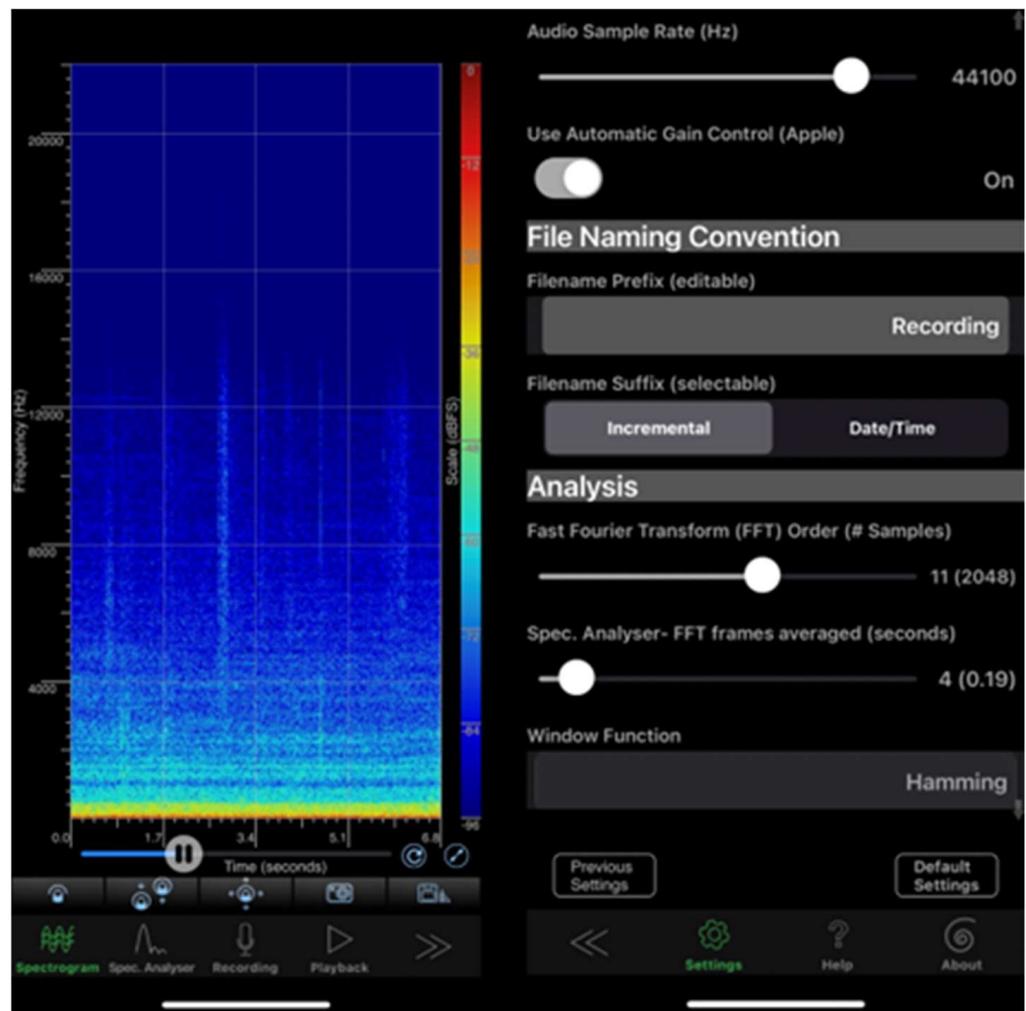
To obtain a considerable amount of the fault signals, the IoT device architecture plays an important role in ensuring that a large number of faulty signal samples are obtained. The combined equipment and network characteristics enable us to obtain a dynamic time signal and convert the energy spectrum on the experimental equipment. The Hamming window function is used to obtain the spectrum signal. Figure 9 shows the settings of the spectrum signal.

The sampling frequency is 44,100 Hz, and the acquisition time of each data sample is 40 s. In converting the frequency spectrum, signal preprocessing is first performed to eliminate background noise. According to the “Nyquist–Shannon” sampling theorem, the sampling frequency must be greater than twice the maximum frequency required for reproduction [42]. Since the human hearing range is approximately 20–20,000 Hz, the sampling frequency must be greater than 40 kHz. The audio codec of our smartphone has a standard sampling frequency of 44.1 kHz, which corresponds to a sampling rate of 20–20 kHz in the audible range of the human ear [43]. This analysis uses the LPC and wavelet algorithms for filtering, and the sound signal is converted from a continuous time domain signal to a sound spectrum frequency domain. Sound features are filtered through MATLAB to create a training dataset. Figure 10 shows the 43 normal and fault conditions of LPC and wavelet on the MATLAB platform with audio signal spectrum characteristics.

**Table 1.** The definition of 43 driving fault signals (engine, tire, and chassis).

Item	Feature Signal Condition	Statement
1	Tire V30 32 psi	Normal pressure in 30 km/h.
2	Tire V30 50 psi	High pressure in 30 km/h.
3	Tire V30 20 psi	Low pressure in 30 km/h.
4	Tire V30 32 psi fail	Tire wear and normal pressure in 30 km/h.
5	Tire V30 50 psi fail	Tire wear and high pressure in 30 km/h.
6	Tire V30 20 psi fail	Tire wear and low pressure in 30 km/h.
7	Tire V20 32 psi	Normal pressure in 20 km/h.
8	Tire V20 50 psi	High pressure in 20 km/h.
9	Tire V20 20 psi	Low pressure in 20 km/h.
10	Tire V20 32 psi fail	Tire wear and normal pressure in 20 km/h.
11	Tire V20 50 psi fail	Tire wear and high pressure in 20 km/h.
12	Tire V20 20 psi fail	Tire wear and low pressure in 20 km/h.
13	Tap V30 32 psi	Tire studs and normal pressure in 30 km/h.
14	Tap V30 50 psi	Tire studs and high pressure in 30 km/h.
15	Tap V30 20 psi	Tire studs and low pressure in 30 km/h.
16	Tap V20 32 psi	Tire studs and normal pressure in 20 km/h.
17	Tap V20 50 psi	Tire studs and high pressure in 20 km/h.
18	Tap V20 20 psi	Tire studs and low pressure in 20 km/h.
19	Belt 1000 rpm normal	Belt normal at 1000 rpm.
20	Belt 1500 rpm normal	Belt normal at 1500 rpm.
21	Belt 2000 rpm normal	Belt normal at 2000 rpm.
22	Belt 1000 rpm idle speed	Belt idle speed at 1000 rpm.
23	Belt 1500 rpm idle speed	Belt idle speed at 1500 rpm.
24	Belt 2000 rpm idle speed	Belt idle speed at 2000 rpm.
25	Toe-in V30 32 psi	Chassis toe-in and normal pressure in 30 km/h.
26	Toe-in V30 50 psi	Chassis toe-in and high pressure in 30 km/h.
27	Toe-in V30 20 psi	Chassis toe-in and low pressure in 30 km/h.
28	Toe-in V20 32 psi	Chassis toe-in and normal pressure in 20 km/h.
29	Toe-in V20 50 psi	Chassis toe-in and high pressure in 20 km/h.
30	Toe-in V20 20 psi	Chassis toe-in and low pressure in 20 km/h.
31	Toe-out V30 32 psi	Chassis toe-out and normal pressure in 30 km/h.
32	Toe-out V30 50 psi	Chassis toe-out and high pressure in 30 km/h.
33	Toe-out V30 20 psi	Chassis toe-out and low pressure in 30 km/h.
34	Toe-out V20 32 psi	Chassis toe-out and normal pressure in 20 km/h.
35	Toe-out V20 50 psi	Chassis toe-out and high pressure in 20 km/h.
36	Toe-out V20 20 psi	Chassis toe-out and low pressure in 20 km/h.
37	Drive V10 32 psi	Broken drive shaft boot and normal pressure in 10 km/h.
38	Drive V10 50 psi	Broken drive shaft boot and high pressure in 10 km/h.
39	Drive V10 20 psi	Broken drive shaft boot and low pressure in 10 km/h.
40	Drive V20 50 psi	Broken drive shaft boot and high pressure in 20 km/h.
41	No.1 nozzle 1000 rpm	Engine single cylinder misfire, idle speed at 1000 rpm.
42	No.1 nozzle 1500 rpm	Engine single cylinder misfire, idle speed at 1500 rpm.
43	No.1 nozzle 2000 rpm	Engine single cylinder misfire, idle speed at 2000 rpm.

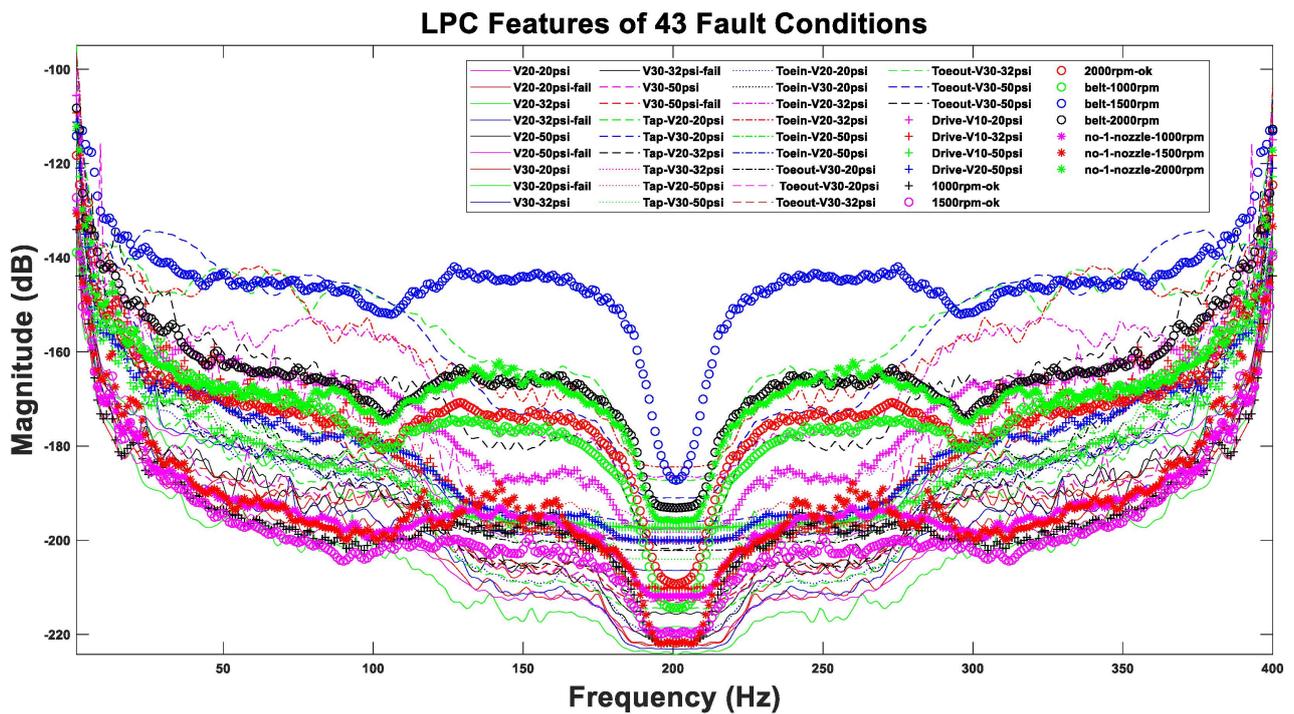
The second part establishes the spectral characteristic signal dataset. To effectively identify the characteristics of various types of faults, the original spectral characteristics are reduced to 10,000 characteristic lengths as filtered spectral characteristics. Then, 30 and 40 sets of characteristic coefficients were set for 43 fault conditions. A total of 1290 and 1720 voiceprint characteristic data were constructed, respectively. The voiceprint feature on the horizontal axis is expected to be set to 10000 points, that is, 10000 pieces of feature data to train the model. The dimensions of the total dataset are 1290\*10000 and 1720\*10000, respectively. Figure 11 shows the settings diagram of the dataset.



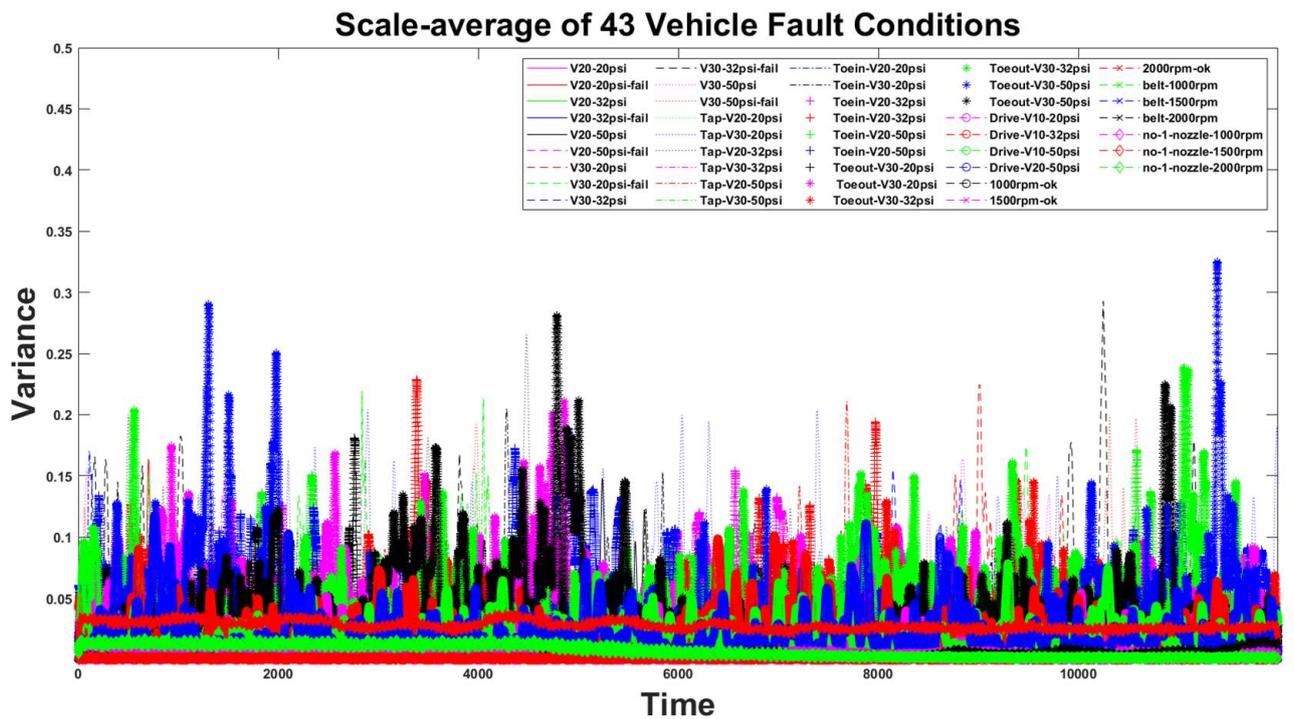
**Figure 9.** Hamming window function by the measurement industrial fault signals.

In the third part, after completing the construction and labeling of the dataset, the Pytorch architecture in Python is used to import the dataset into the three algorithms of DNN, CNN and LSTM for classification. The Pytorch architecture used in this study uses adaptive moment estimation (Adam) as the optimizer function. The Adam is an adaptive learning rate algorithm whose essence is an RMSprop optimization method with a momentum term and is currently the most widely used model training optimizer [44]. The experimental process is mainly used to cross-compare the datasets constructed by two different filtering algorithms LPC and wavelet and the differences between the recognition results and learning speeds caused by the three machine learning algorithms. The learning rate and batch size are based on the premise of using the least Epoch (the number of iterations) to adjust the parameters to achieve the best recognition rate.

The learning rate controls the learning rate of the model. The larger the learning rate, the faster the convergence rate and the less training time. After the extreme value is exceeded, the loss function stops decreasing and oscillates at a certain position. The smaller the learning rate, the slower the convergence speed, the more time it takes to train the model, and the easier the network to enter the local minimum, which makes the loss function converge poorly. Therefore, the appropriate learning rate can be adjusted by observing the change of the model loss parameters. In this study, we set the learning rate of the three algorithms to 0.00001 after many experiments, which not only achieves the best model convergence overall, but also facilitates cross-validation of different deep learning algorithms.

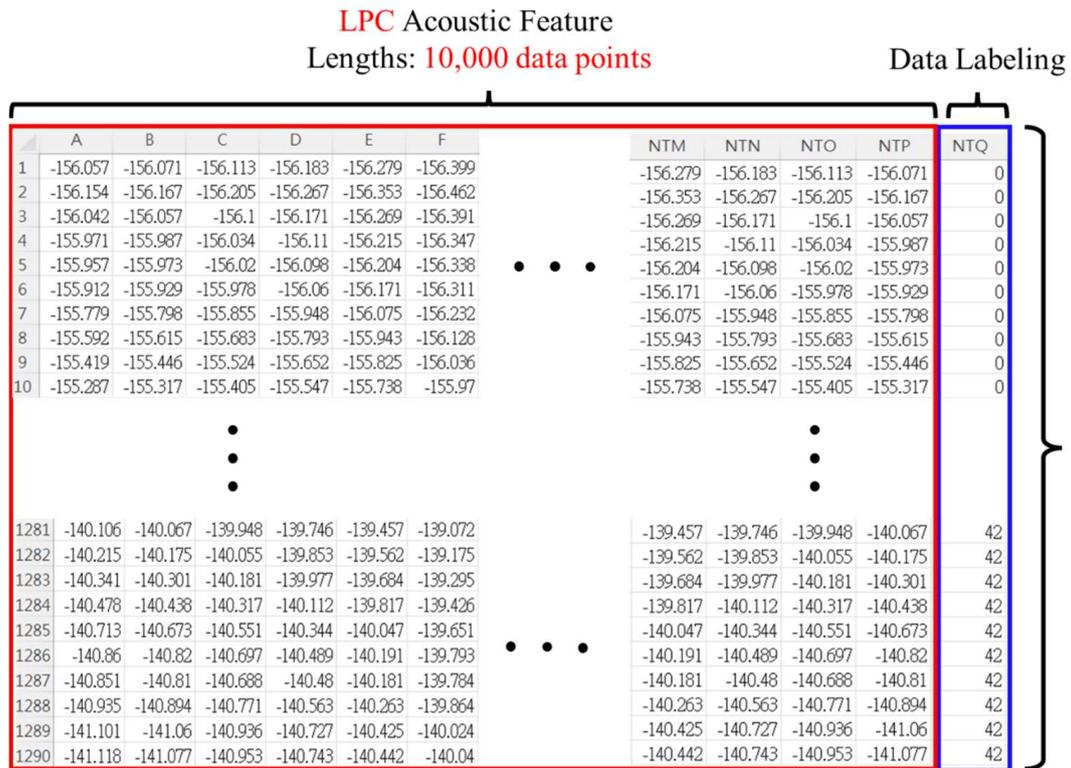


(a)

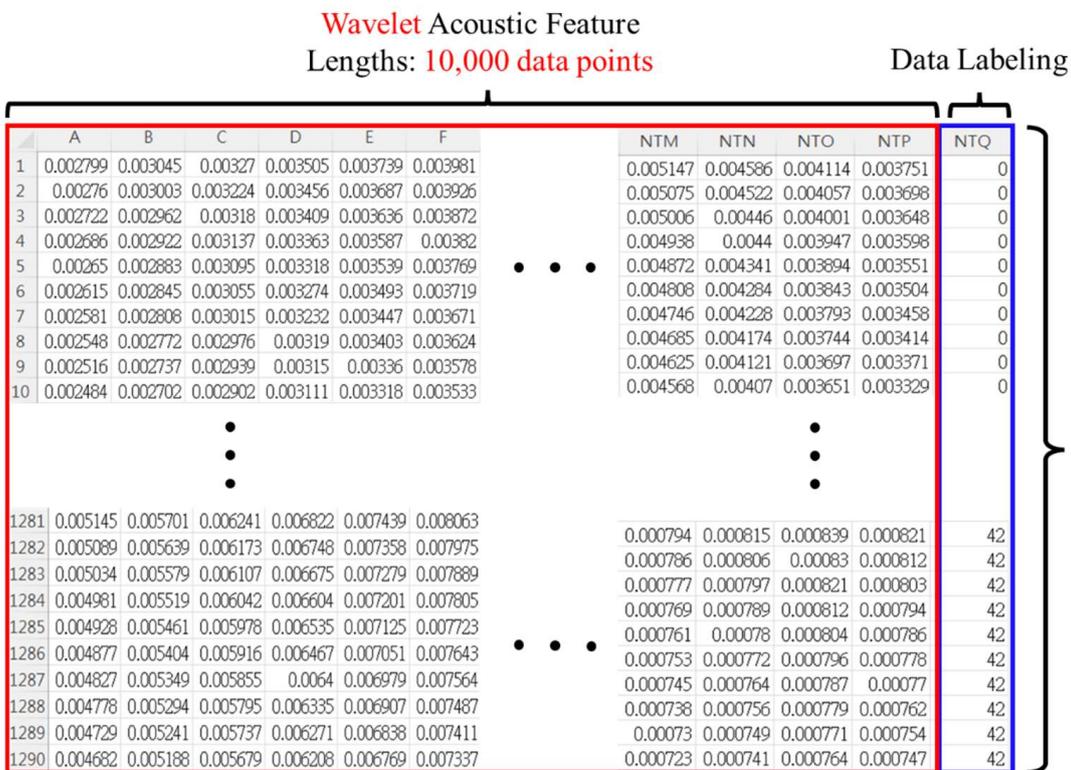


(b)

Figure 10. Features of the 43 conditions of the acoustic signals after (a) LPC and (b) wavelet extractions.



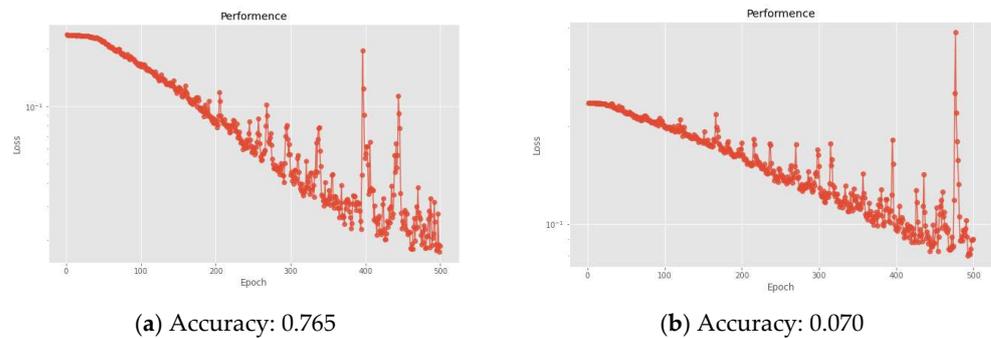
(a)



(b)

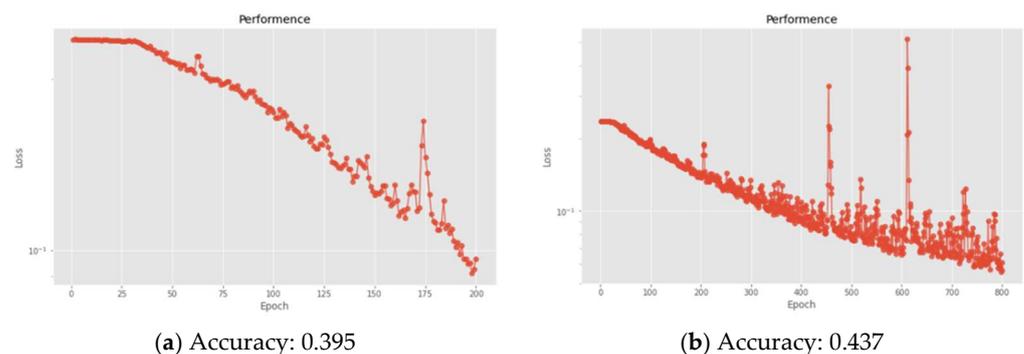
Figure 11. Dataset map. The horizontal axis is the data labeling of the audio signal spectrum, and the vertical axis is the predictive coefficient number sequence. (a) LPC and (b) wavelet.

In terms of the hidden layer setting of the DNN algorithm in this study, at first, we tried to use a DNN architecture with 12 hidden layers, but it was found that the training results of the model with too few hidden layers were not ideal, as shown in Figure 12a. After that, we used a DNN architecture with 20 hidden layers and found that there was an overfitting problem. In addition, the model training and identification results were also extremely poor, as shown in Figure 12b. Therefore, it was finally decided to adopt a DNN architecture with 15 hidden layers as the hidden layer parameter setting in this study.



**Figure 12.** Loss function and accuracy of various hidden layer settings of DNN: (a) 12 layers. (b) 20 layers.

Taking the LPC 30 feature as an example, if epoch = 200, the number of iterations of the model training is insufficient, and underfitting occurs. Although the model training time can be the shortest, the recognition accuracy is low (as shown in Figure 13a). If we lengthen the epoch to 800, the model training stability is extremely low, and overfitting occurs (as shown in Figure 13b). Therefore, it is more appropriate to use epoch = 500, thereby avoiding overfitting and underfitting. In addition, in order to facilitate the comparison of the learning effects of the three deep learning algorithms DNN, CNN and LSTM on the same dataset, we use the same epoch to facilitate cross-comparison of the time consumed by the training model.



**Figure 13.** Loss function and accuracy of various epoch settings of DNN. (a) Epoch = 200, underfitting. (b) Epoch = 800, overfitting.

The DNN parameters of this experiment are set as: the learning rate = 0.00001, the iteration time (Epoch time) = 500, the batch size (Batch) = 128, the test size = 0.3. We adopted a 15-layer deep neural network architecture, and the number of neurons in each hidden layer is shown in Figure 14. The connection between the hidden layers is fully connected. In order to solve the problem of gradient disappearance at saturation, Xavier Glorot et al. proposed a linear rectified function (Rectified Linear Unit, Relu) [45]. The disadvantage is that when the variable is updated too fast and when the function has not found the optimal value, the neuron will become less than 0 and the neuron will die. Therefore, the activation

function we selected in the experiment is Selu (Scaled Exponential Linear Unit), and Selu is a variant of Relu [46], as shown in Figure 15, where its function is:

$$\text{Selu}(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x < 0 \end{cases} \quad (16)$$

where  $\lambda$  is a fixed value of 1.05070098736, and  $\alpha$  is 1.67326324235.

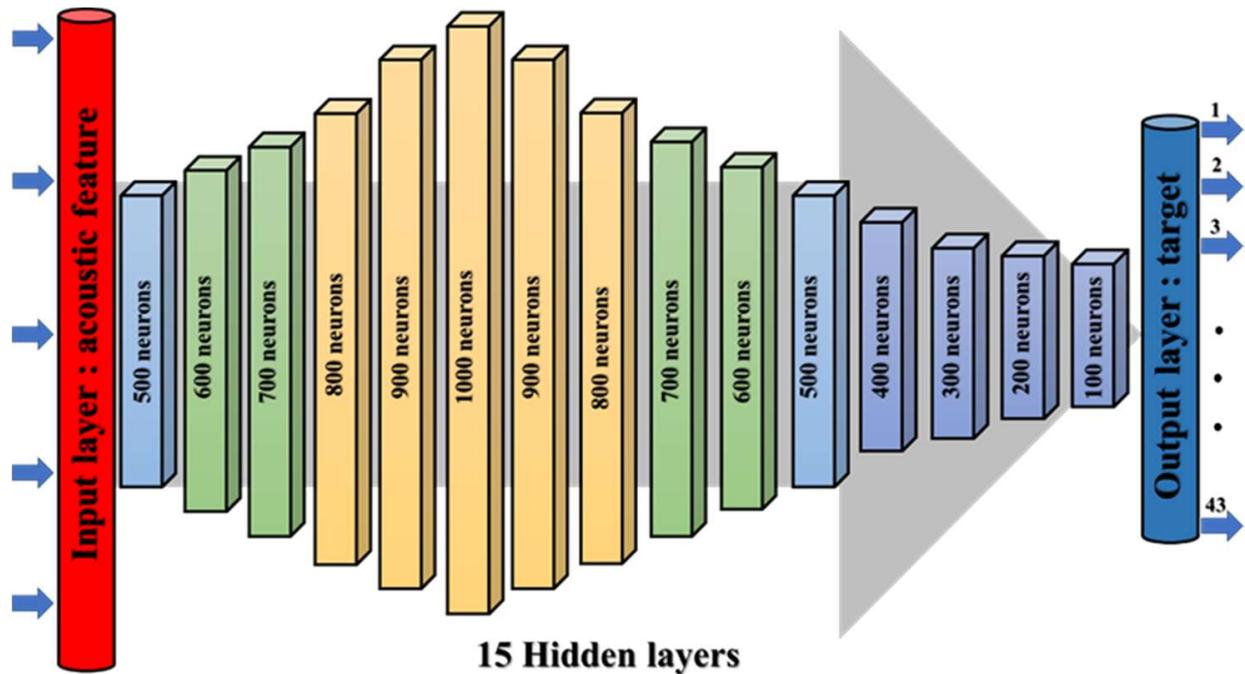


Figure 14. DNN architecture.

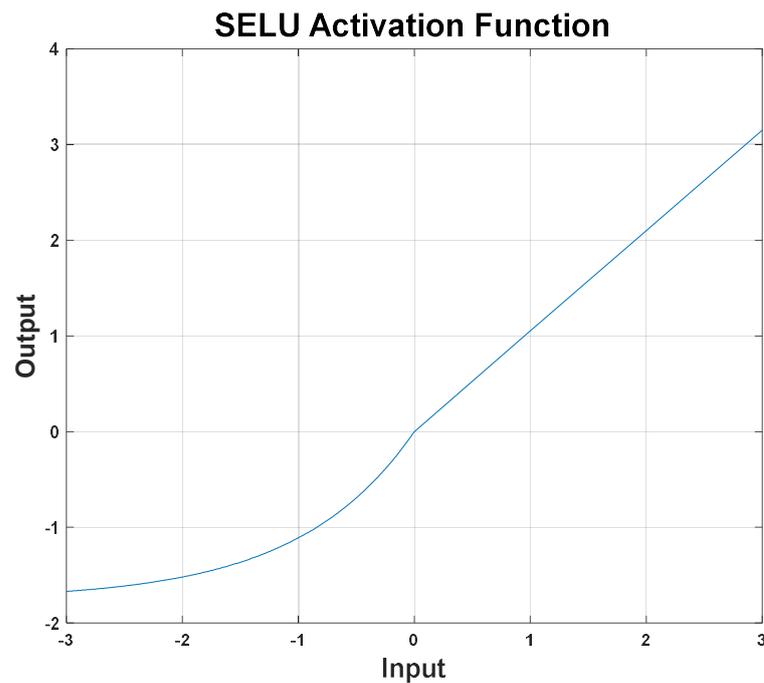


Figure 15. SELU activation function.

Furthermore, in this experimental CNN model, three convolutional layers and three pooling layers are used to reduce the size of audio signal features. The sizes of the three convolutional layers are  $(5 \times 5 \times 3)$ ,  $(5 \times 5 \times 5)$ ,  $(5 \times 5 \times 5)$  in sequence, and the pooling layers are all  $2 \times 2$ . That is, the filter measures is  $2 \times 2$ , as shown in Figure 16. The network architecture parameter settings include a learning rate of 0.00001, a batch size of 128, and a test set scale of 0.3.

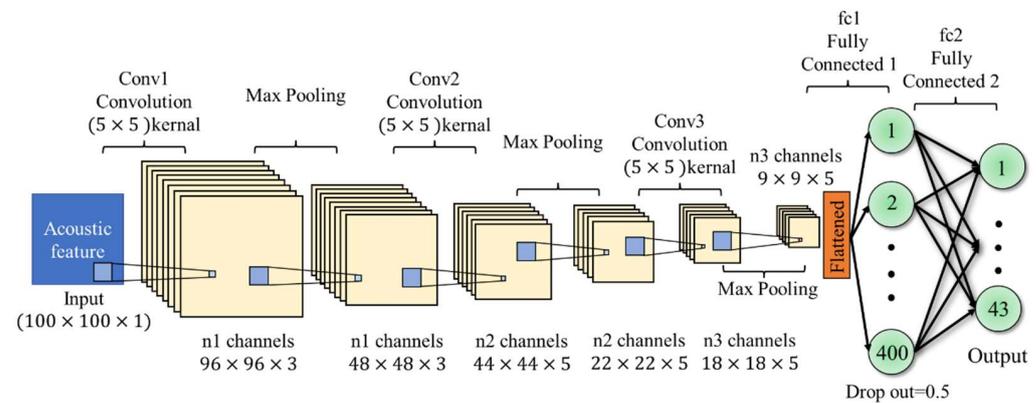


Figure 16. CNN architecture.

The third neural network approach used in this experiment is the Long Short-Term Memory (LSTM) algorithm. The LSTM architecture we used in the experimental process is based on two hidden layers of the network, and 300 hidden neurons are used in each fault condition with 30 and 40 features, where the LSTM training model network label layer is  $2 \times 300$ , indicating that there are two hidden layers containing 300 hidden neurons, as shown in Figure 17. Here, we compare the training performance of the LPC dataset and the wavelet dataset. Two different spectral datasets use a batch size of 128, the learning rate is set to 0.00001, and the number of iterations is set to 500.

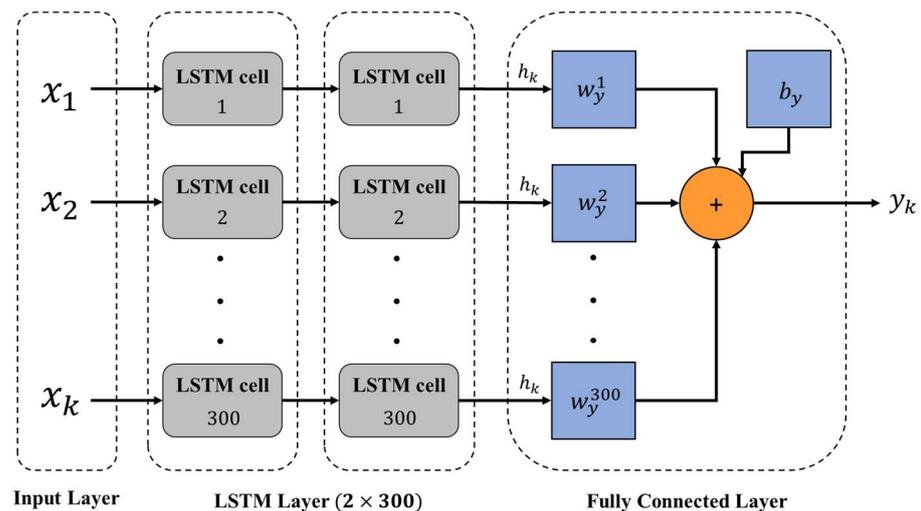


Figure 17. LSTM architecture.

#### 4. Results and Discussion

This part uses the DNN algorithm to model the training dataset in the Python Pytorch framework. We compare the results for DNNs with more than 10 hidden layers. Generally, in deep learning, deeper hidden layers are more accurate than shallow hidden layers.

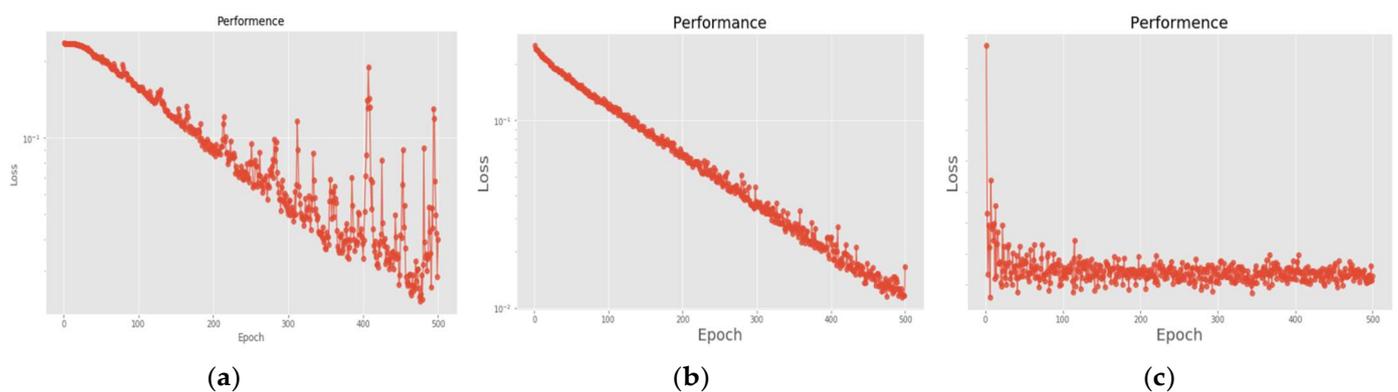
As showed in Table 2, the LPC dataset uses the CNN algorithm to achieve better identification results, and the LSTM algorithm is extremely poor, followed by DNN. The wavelet dataset using LSTM and DNN has a good identification effect, while the effect

of the CNN algorithm is slightly worse. In terms of wavelet dataset, the identification accuracy of the DNN algorithm is as high as 1.00, which is 16.57% higher than 0.86 of the CNN algorithm and 13.82% higher than 0.88 of the LSTM algorithm. As far as the LPC dataset is concerned, the accuracy rate of the CNN algorithm reaches 1.00, which is 72.77% higher than that of the DNN algorithm. In terms of model training time, the training time of the two datasets imported into the CNN algorithm is the longest, followed by DNN, and the shortest by LSTM. Compared with the LPC dataset, the wavelet dataset takes a longer time to import the three machine learning algorithms for the model training, and the difference is the largest in the CNN algorithm, which is 3.13% longer than the LPC dataset training time.

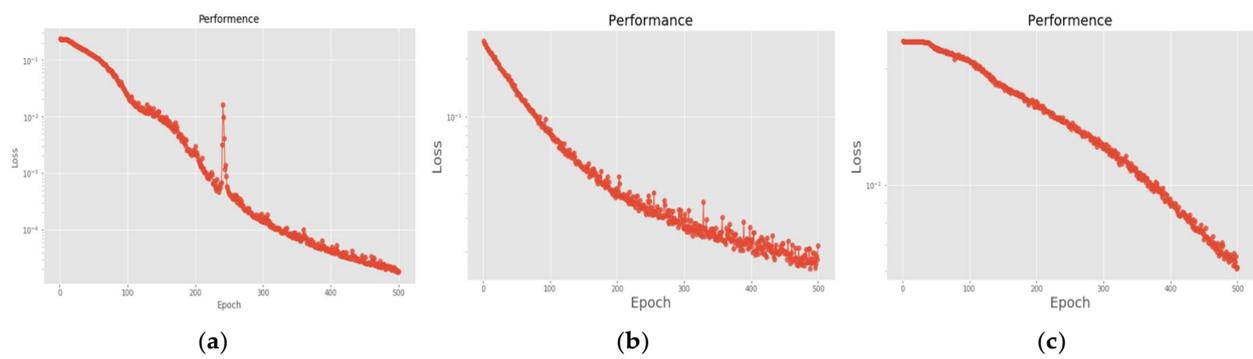
**Table 2.** Results with three deep learning structures of 30-feature datasets.

Results of 30-Feature Datasets	DNN	CNN	LSTM
Accuracy for LPC	0.579	1.000	0.023
Training time for LPC (s)	97.515	115.200	70.031
Accuracy for Wavelet	1.000	0.858	0.879
Training time for Wavelet (s)	97.747	118.767	70.042

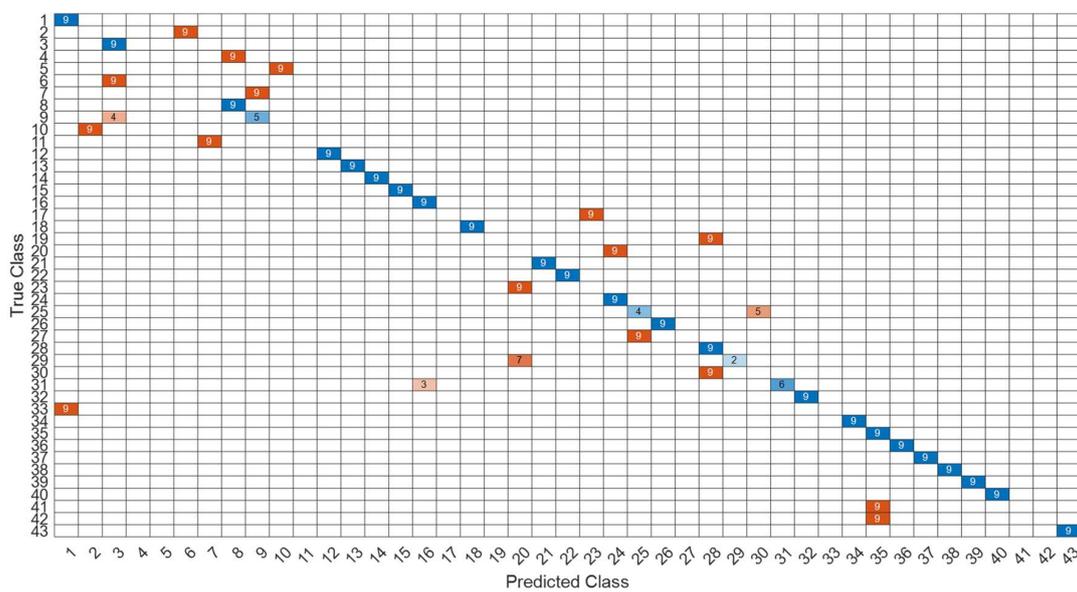
Figure 18 is a comparison of the loss functions of the LPC feature dataset (with 30 features). Figure 18a–c are the loss functions using DNN, CNN and LSTM algorithms, respectively. We found that the convergence speed of DNN and CNN is faster. Compared with the wavelet dataset, the CNN algorithm converges more stably, whereas LSTM has extremely poor convergence here. Figure 19 shows the comparison of the loss function of 30 kinds of features by wavelet. Figure 19a–c are the loss functions using DNN, CNN and LSTM algorithms, respectively. We found that DNN and CNN converge faster, but DNN is more stable overall. Furthermore, the gradient convergence of LSTM is slower, but the stability is higher than the previous two. In addition, Figures 20–22 are the confusion matrices of the LPC datasets with 30 features imported into DNN, CNN and LSTM algorithms for classification and identification; Figures 23–25 are the wavelet datasets of 30 features imported into DNN and CNN confusion matrix for classification and identification with LSTM algorithm. The results of our experiments can also be seen from the confusion matrix.



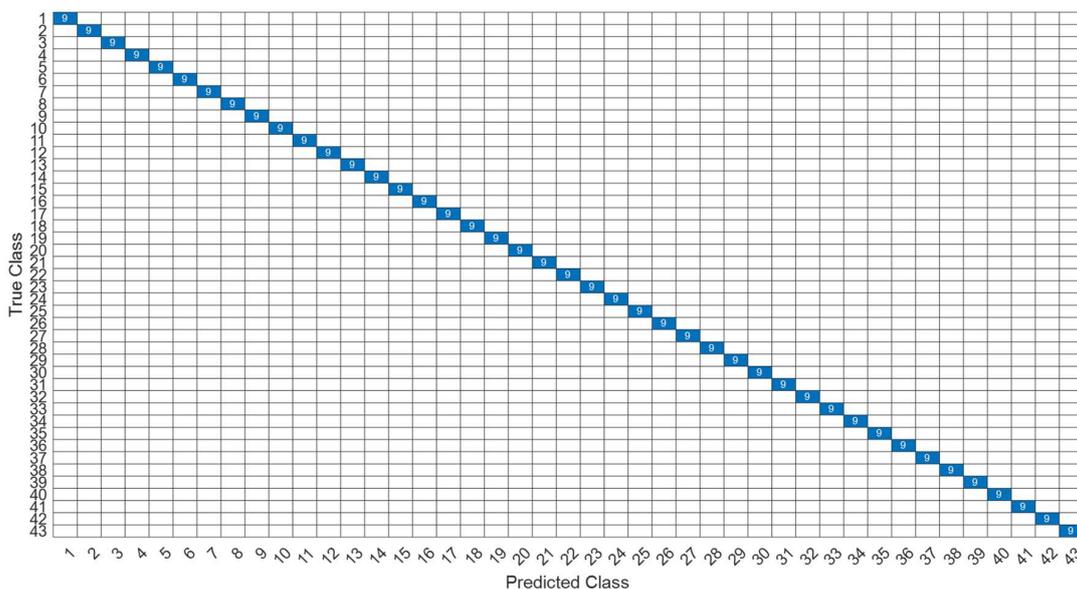
**Figure 18.** Loss function of LPC 30 feature dataset. (a) LPC + DNN. (b) LPC + CNN. (c) LPC + LSTM.



**Figure 19.** Loss function of wavelet 30 feature dataset. (a) Wavelet + DNN. (b) Wavelet + CNN. (c) Wavelet + LSTM.



**Figure 20.** The confusion matrix of the DNN model with LPC 30-features dataset.



**Figure 21.** The confusion matrix of the CNN model with LPC 30-features dataset.



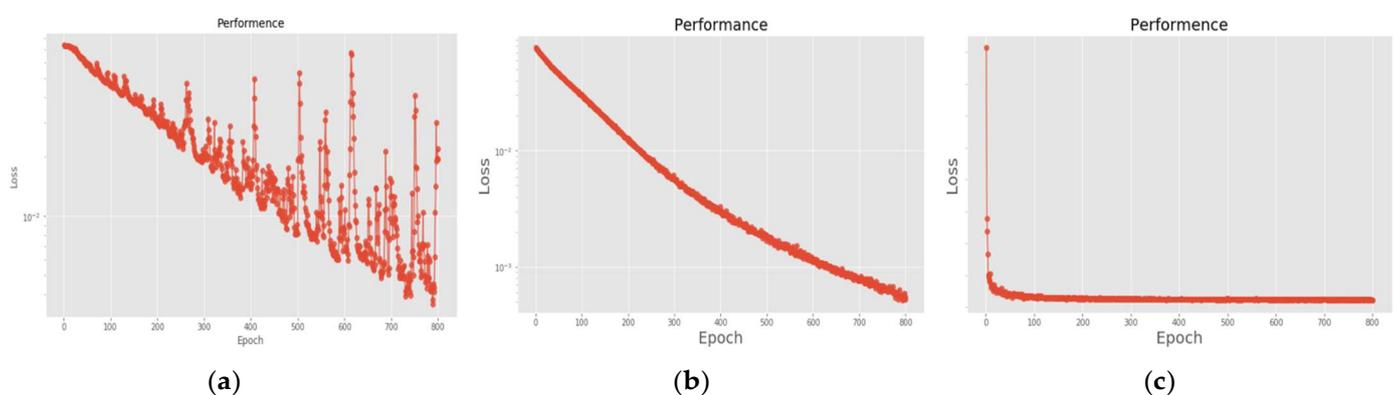


using LSTM and DNN has a good identification effect, while the effect of the CNN algorithm is slightly worse. For the wavelet dataset, the identification accuracy of the DNN algorithm and the LSTM algorithm is as high as 1.00, which is 9.55% higher than the 0.86 of the CNN algorithm. As far as the LPC dataset is concerned, the accuracy rate of the CNN algorithm reaches 1.00, which is 20.84% higher than that of the DNN algorithm. In terms of the model training time, the training time of the two datasets imported into the CNN algorithm is the longest, followed by LSTM. The DNN is the shortest in the training time. Unlike the 30 features, DNN is a calculus with a shorter training time between the two. Compared with the LPC dataset, the wavelet dataset has a shorter time for model training when the three machine learning algorithms are applied.

**Table 3.** Results with three deep learning structures of 40-feature datasets.

Results of 40-Feature Datasets	DNN	CNN	LSTM
Accuracy for LPC	0.828	1.000	0.023
Training time for LPC (s)	125.193	207.647	163.552
Accuracy for Wavelet	1.000	0.913	1.000
Training time for Wavelet (s)	124.090	207.727	162.962

Figure 26 is a comparison of the loss functions of the LPC feature dataset with 40 features. Figure 26a–c are the loss functions using DNN, CNN and LSTM algorithms, respectively. Similar to the previous results of taking 30 feature datasets: DNN and CNN converge faster. Moreover, compared with wavelet datasets, the CNN algorithm converges more stably, whereas LSTM has extremely poor convergence here. Figure 27 shows the comparison of the loss function of 40 kinds of features by wavelet. Figure 27a–c are the loss functions using DNN, CNN and LSTM algorithms, respectively. We find that the CNN loss function converges more smoothly, and the DNN and CNN loss functions are more stable than the experimental results with 30 features. In addition, Figures 28–30 are the confusion matrices of the LPC dataset with 40 features imported into DNN, CNN and LSTM algorithms, respectively, for classification and identification. Figures 31–33 are the confusion matrices of the 40-features wavelet dataset imported into DNN, CNN, and LSTM algorithms, respectively, for classification and identification. The results of our experiments can also be seen from the confusion matrix. Furthermore, LSTM has faster gradient convergence than the experimental results with 30 features per failure, but is less stable when the number of iterations is small.



**Figure 26.** Loss function of LPC 40-features dataset. (a) LPC + DNN. (b) LPC + CNN. (c) LPC + LSTM.

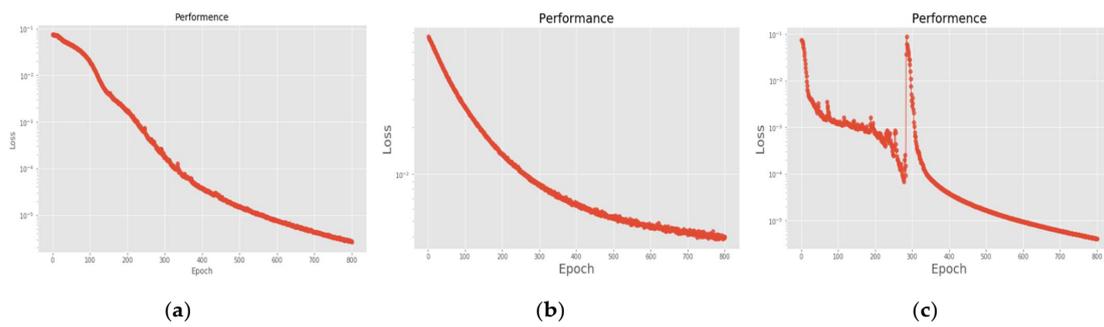


Figure 27. Loss function of wavelet 40-features dataset. (a) Wavelet + DNN. (b) Wavelet + CNN. (c) Wavelet + LSTM.

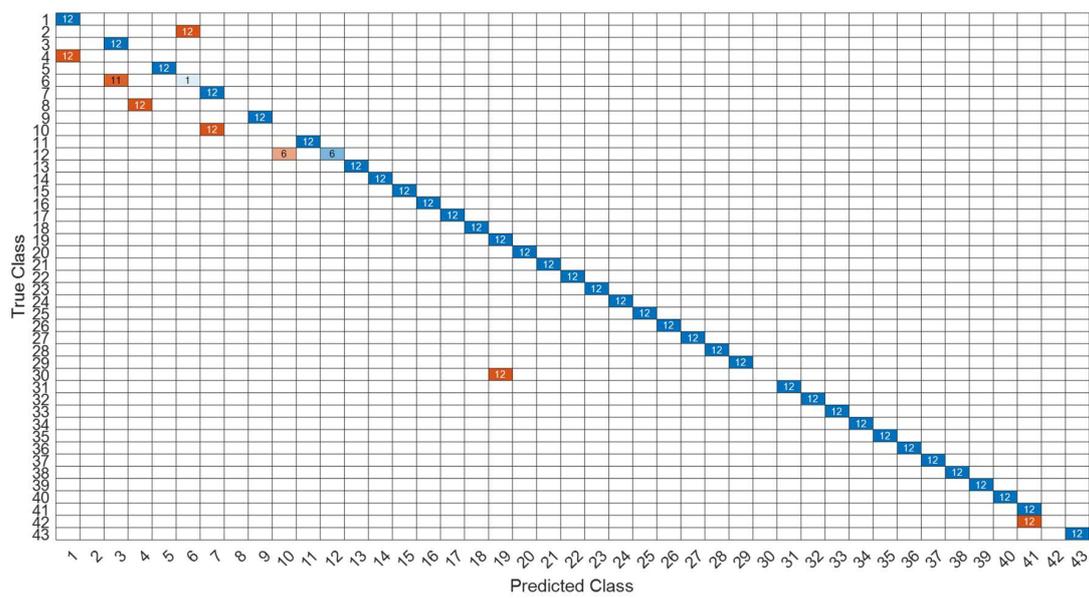


Figure 28. The confusion matrix of the DNN model with LPC 40-features dataset.

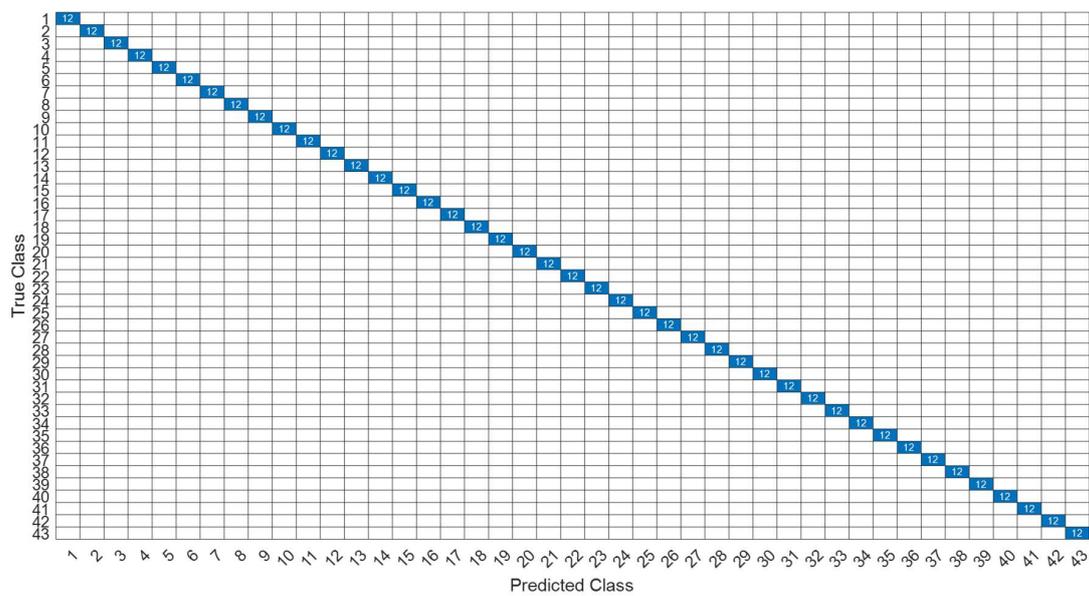


Figure 29. The confusion matrix of the CNN model with LPC 40-features dataset.

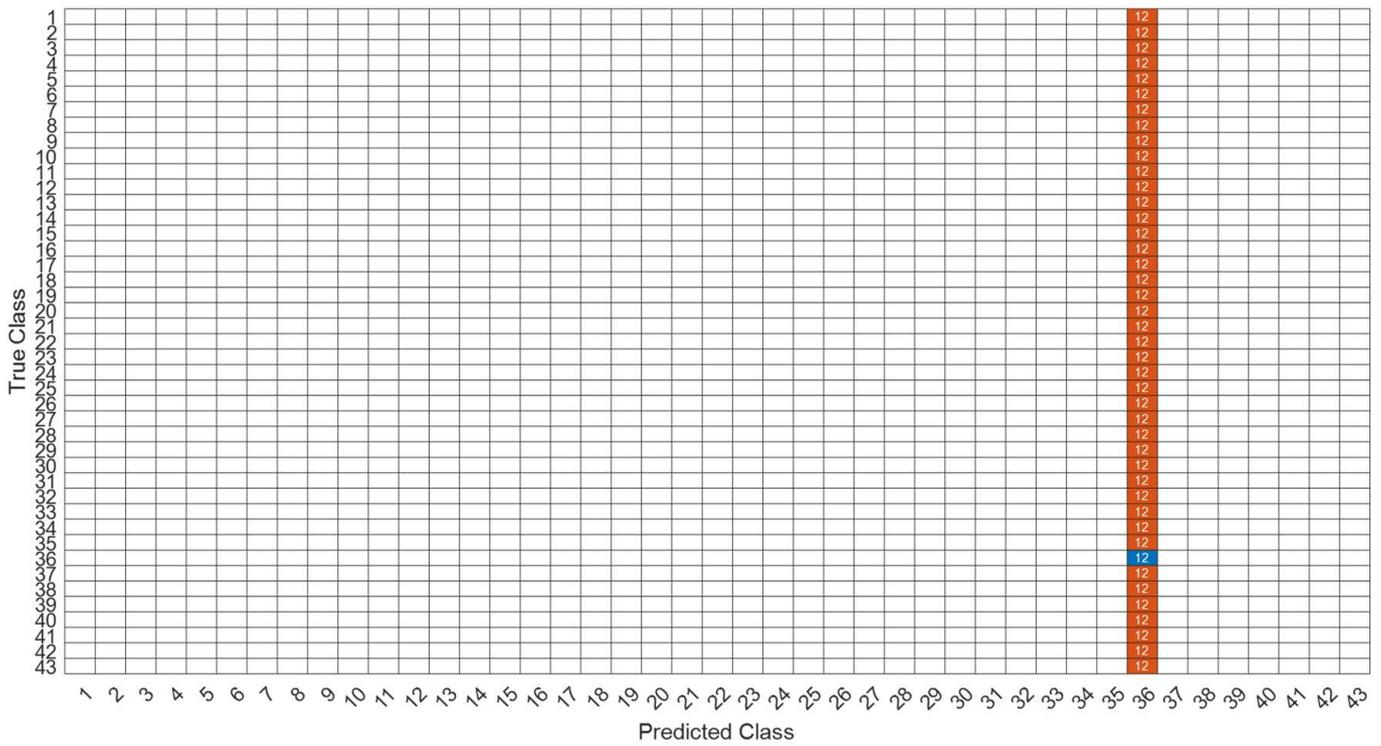


Figure 30. The confusion matrix of the LSTM model with LPC 40-features dataset.

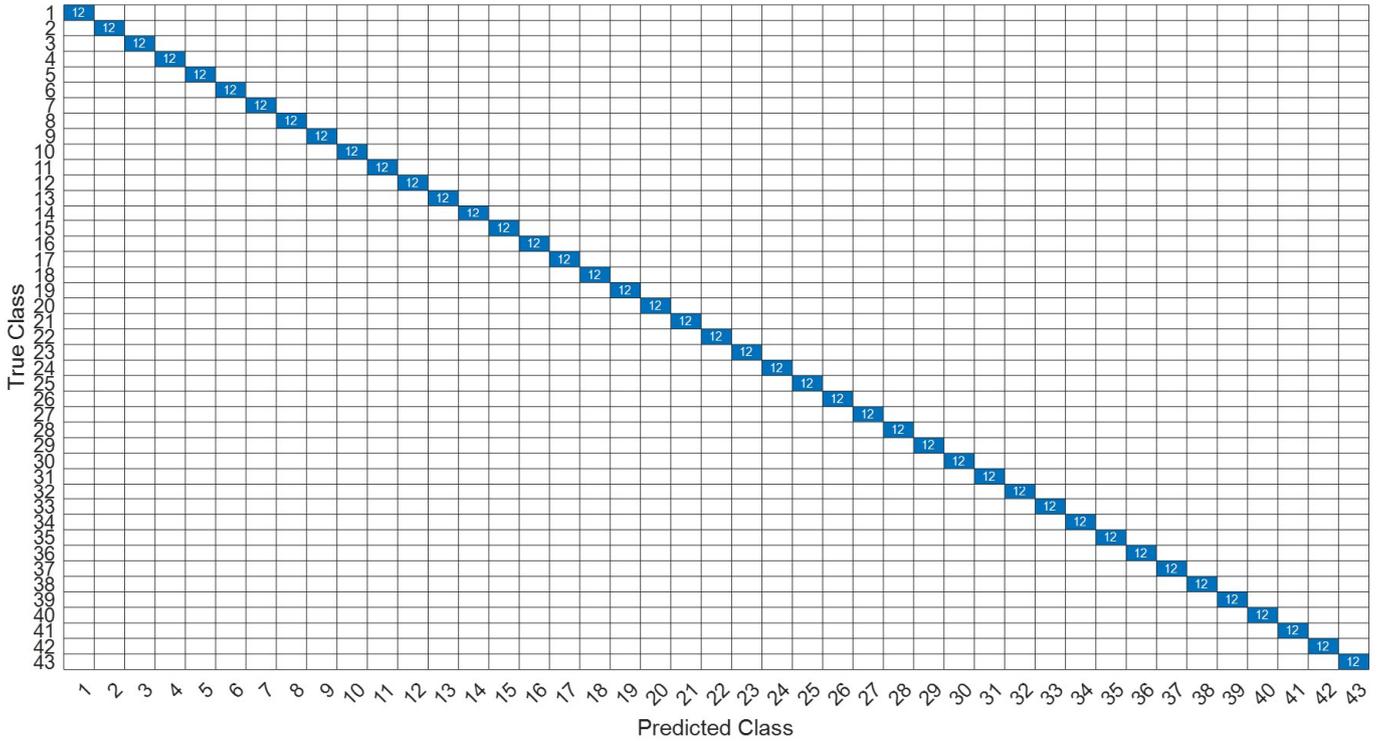


Figure 31. The confusion matrix of the DNN model with wavelet 40-features dataset.

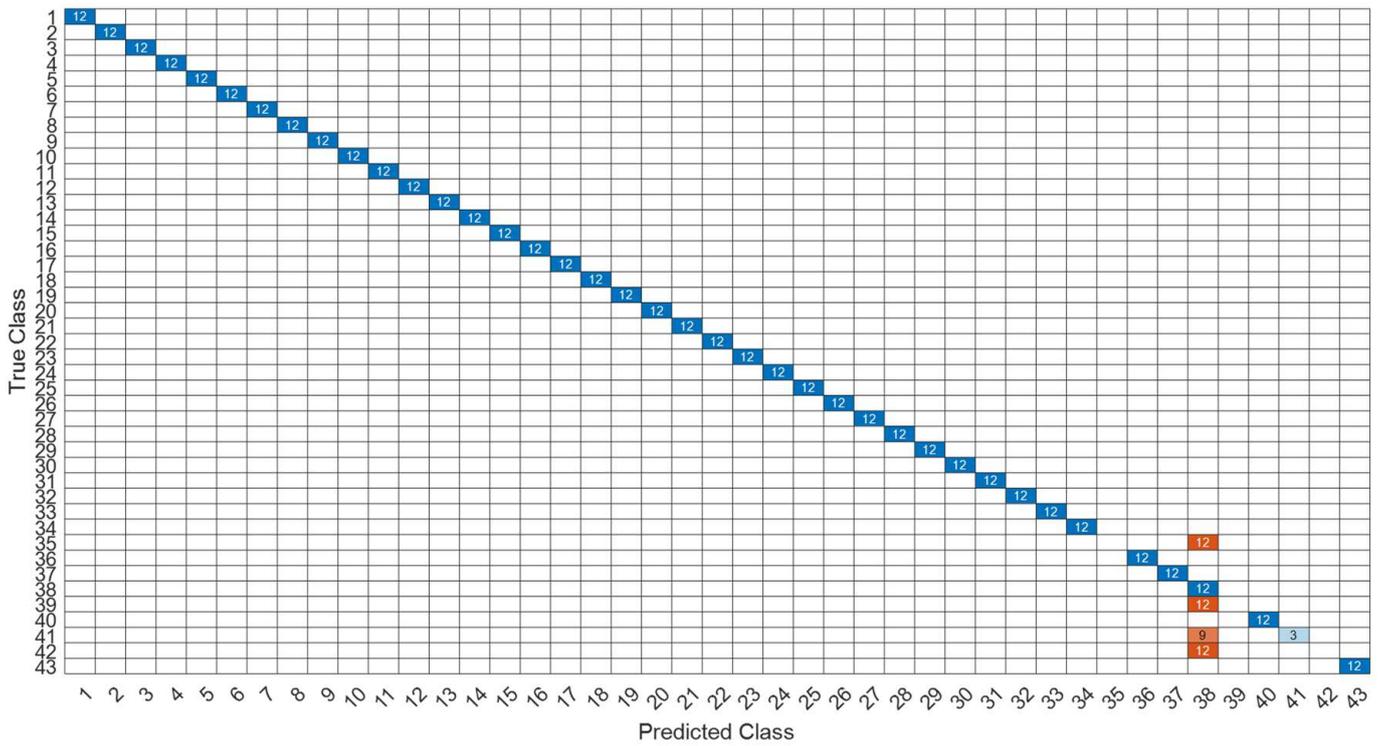


Figure 32. The confusion matrix of the CNN model with wavelet 40-features dataset.

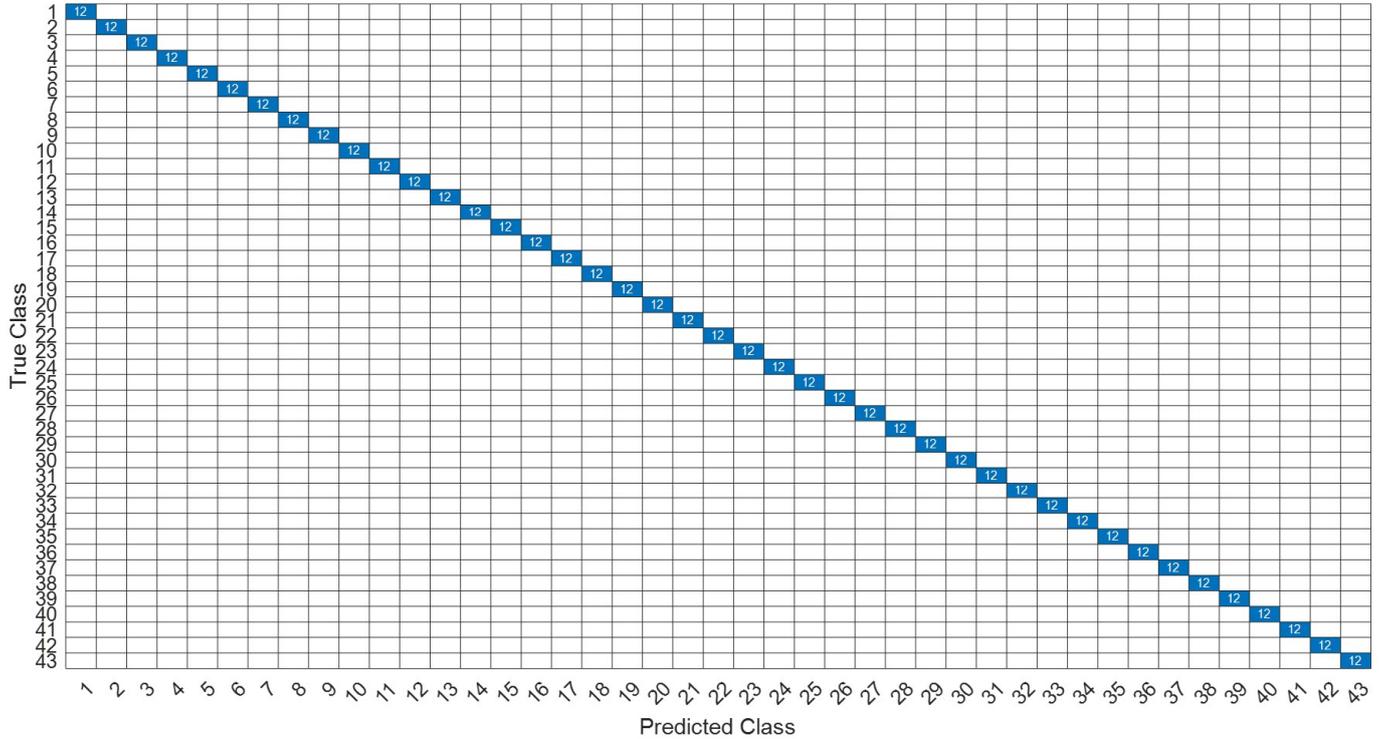


Figure 33. The confusion matrix of the LSTM model with wavelet 40-features dataset.

### 5. Conclusions

An early vehicle fault signal classification method is proposed based on voiceprint filtering combined with deep learning algorithms. We collected 43 different vehicle breakdown signals. LPC and wavelet were used to filter the original signal to obtain important

signal spectral characteristics used to define fault type. In addition, three machine learning algorithms, DNN, CNN and LSTM, were used to develop automatic diagnosis methods to classify complex fault features. Looking at the whole experiment, in terms of the LPC dataset, CNN has the best performance, followed by DNN, and finally LSTM. However, in terms of model training time, the order of the three is reversed.

The LPC dataset and the CNN algorithm can obtain the best identification results, but the training process is also the most time-consuming. With regard to LPC + LSTM, although the training time is the shortest, it is almost impossible to identify and classify. That is, the accuracy rate is extremely low. Furthermore, for the wavelet dataset, DNN has the best performance both in terms of identification performance and training time. For datasets with large dimensions, the accuracy of the wavelet algorithm combined with LSTM also has good identification performance. Based on our experimental results, we can infer that in this experiment, the wavelet algorithm combined with DNN can not only achieve the best identification performance, but also the shortest model training time when the dataset dimension is large.

All deep learning models are implemented on the Python Pytorch platform using NVIDIA GeForce GTX. In this research, early failure prediction in vehicles is of great significance to the emerging Internet of Vehicles and can help increase the production capacity of Internet of Things and Industry 4.0 applications. Future work will seek to combine two filtering methods, such as MFCC + LPC or MFCC + wavelet, and to apply machine learning methods suitable for natural language processing (NLP), such as long short-term memory (LSTM) work to produce an application that effectively achieves faster identification. Voice recognition methods are an area worthy of attention and have extensive applications in daily life; thus, combining artificial intelligence and voiceprint recognition can potentially produce significant and widespread benefits.

**Author Contributions:** Conceptualization, C.-S.A.G.; methodology, C.-S.A.G., C.-H.S.S. and Y.-E.L.; validation, C.-S.A.G., C.-H.S.S., Y.-E.L., Y.-H.C. and D.-Y.G.; resources, C.-S.A.G.; writing—original draft preparation, C.-H.S.S. and Y.-E.L.; writing—review and editing, C.-S.A.G., C.-H.S.S., Y.-E.L., Y.-H.C. and D.-Y.G.; supervision, C.-S.A.G. and C.-H.S.S.; project administration, C.-S.A.G.; funding acquisition, C.-S.A.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Chang Gung Memorial Hospital, grant number CM-RPD2M0111. The APC was funded by the National Science and Technology Council (NSTC), Taiwan (grant no. 110-2221-E-182-053-).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank Kuo-Wei Chao for assistance, and they appreciate the support from Linkou Chang Gung Memorial Hospital (CGMH) under contract CM-RPD2M0111. This work is also supported, in part, by the National Science and Technology Council (NSTC), Taiwan, under contract 110-2221-E-182-053-. Finally, the authors are thankful for technical assistance from the Department of Mechanical Engineering, National Yang Ming Chiao Tung University, Taiwan.

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

## References

1. Jaynes, C.; Seales, W.B.; Calvert, K.; Fei, Z.; Griffioen, J. The Metaverse: A networked collection of inexpensive, self-configuring, immersive environments. In Proceedings of the Workshop on Virtual Environments 2003 (EGVE'03), Zurich, Switzerland, 22–23 May 2003; Association for Computing Machinery: New York, NY, USA, 2003; pp. 115–124.
2. Strutynska, I.; Dmytrotsa, L.; Kozbur, H.; Hlado, O.; Dudkin, P.; Dudkina, O. Development of Digital Platform to Identify and Monitor the Digital Business Transformation Index. In Proceedings of the 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, 23–26 September 2020; pp. 171–175.

3. Jiang, Y. An Analysis of the Relationship between Mechanical and Electronic Engineering and Artificial Intelligence. In Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China, 14–15 September 2019; pp. 191–194.
4. Ghosh, A.; Chakraborty, D.; Law, A. Artificial intelligence in internet of things. *CAAI Trans. Intell. Technol.* **2018**, *3*, 208–218. [[CrossRef](#)]
5. Buyya, R. Cloud computing: The next revolution in information technology. In Proceedings of the 2010 First International Conference on Parallel, Distributed and Grid Computing (PDGC 2010), Solan, India, 28–30 October 2010; pp. 2–3.
6. Dai, X.; Gao, Z. From model signal to knowledge: A data-driven perspective of fault detection and diagnosis. *IEEE Trans. Ind. Inform.* **2013**, *9*, 2226–2238. [[CrossRef](#)]
7. Purkait, P.; Chakravorti, S. Time and frequency domain analyses based expert system for impulse fault diagnosis in transformers. *IEEE Trans. Dielectr. Electr. Insul.* **2002**, *9*, 433–445. [[CrossRef](#)]
8. Yang, F.; Wang, S.; Li, J.; Liu, Z.; Sun, Q. An overview of Internet of Vehicles. *China Commun.* **2014**, *11*, 1–15. [[CrossRef](#)]
9. Cummings, M.L.; Bauchwitz, B. Safety Implications of Variability in Autonomous Driving Assist Alerting. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 12039–12049. [[CrossRef](#)]
10. Zhang, H.; Zhang, Q.; Liu, J.; Guo, H. Fault detection and repairing for intelligent connected vehicles based on dynamic bayesian network model. *IEEE Internet Things J.* **2018**, *5*, 2431–2440. [[CrossRef](#)]
11. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4298–4311. [[CrossRef](#)]
12. Liu, H.; Ma, J.; Xu, T.; Yan, W.; Ma, L.; Zhang, X. Vehicle detection and classification using distributed fiber optic acoustic sensing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 1363–1374. [[CrossRef](#)]
13. Obeid, N.H.; Battiston, A.; Boileau, T.; Nahid-Mobarakeh, B. Early intermittent interturn fault detection and localization for a permanent magnet synchronous motor of electrical vehicles using wavelet transform. *IEEE Trans. Transp. Electrification.* **2017**, *3*, 694–702. [[CrossRef](#)]
14. Kemalkar, A.K.; Bairagi, V.K. Engine fault diagnosis using sound analysis. In Proceedings of the International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 9–10 September 2016; pp. 943–946.
15. Chao, K.W.; Chen, Y.H.; Ho, Y.Y.; Guu, D.Y.; Luo, L.B.; Tseng, C.W.; Su, C.S.; Gong, C.A.; Huang, Q.Y.; Lee, I.E.; et al. Feature-Driven Fault Classification for Vehicle Driving Health based on Supervised Learning. In Proceedings of the 26th National Conference on Vehicle Engineering, Taichung, Taiwan, 25–26 January 2021.
16. Gong, C.S.A.; Lee, H.C.; Chuang, Y.C.; Li, T.H.; Su, C.H.S.; Huang, L.H.; Hsu, C.W.; Hwang, Y.S.; Lee, J.D.; Chang, C.H. Design and Implementation of Acoustic Sensing System for Online Early Fault Detection in Industrial Fans. *Hindawi J. Sens.* **2018**, *2018*, 4105208. [[CrossRef](#)]
17. Gong, C.S.A.; Su, C.H.S.; Tseng, K.H. Implementation of machine learning for fault classification on vehicle power transmission system. *IEEE Sens. J.* **2020**, *20*, 15163–15176. [[CrossRef](#)]
18. Gong, C.S.A.; Su, C.-H.S.; Chao, K.-W.; Chao, Y.-C.; Su, C.-K.; Chiu, W.-H. Exploiting deep neural network and long short-term memory methodologies in bioacoustic classification of LPC-based features. *PLoS ONE* **2021**, *16*, e0259140. [[CrossRef](#)]
19. Ameid, T.; Menacer, A.; Talhaoui, H.; Harzelli, I. Broken rotor bar fault diagnosis using fast Fourier transform applied to field-oriented control induction machine: Simulation and experimental study. *Int. J. Adv. Manuf. Technol.* **2017**, *92*, 917–928. [[CrossRef](#)]
20. Ayhan, T.; Dehaene, W.; Verhelst, M. A 128:2048/1536 point FFT hardware implementation with output pruning. In Proceedings of the 2014 22nd European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, 1–5 September 2014; pp. 266–270.
21. Yan, R.; Gao, R.X.; Chen, X. Wavelets for fault diagnosis of rotary machines: A review with applications. *Signal Process.* **2014**, *96*, 1–15. [[CrossRef](#)]
22. Swedia, E.R.; Mutiara, A.B.; Subali, M.; Ernastuti. Deep Learning Long-Short Term Memory (LSTM) for Indonesian Speech Digit Recognition using LPC and MFCC Feature. In Proceedings of the 2018 Third International Conference on Informatics and Computing (ICIC), Palembang, Indonesia, 17–18 October 2018; pp. 1–5. [[CrossRef](#)]
23. Chong, U.P.; Lee, S.S.; Sohn, C.H. Fault diagnosis of the machines in power plant using LPC. In Proceedings of the 8th Russian-Korean International Symposium on Science and Technology (KORUS), Tomsk, Russia, 26 June–3 July 2004; pp. 170–174. [[CrossRef](#)]
24. Chao, K.W.; Hu, N.-Z.; Chao, Y.-C.; Su, C.-K.; Chiu, W.-H. Implementation of artificial intelligence for classification of frogs in bioacoustics. *Symmetry* **2019**, *11*, 1454. [[CrossRef](#)]
25. Yin, S.; Huang, Z. Performance monitoring for vehicle suspension system via fuzzy positivistic C-means clustering based on accelerometer measurements. *IEEE/ASME Trans. Mechatron.* **2014**, *20*, 2613–2620. [[CrossRef](#)]
26. Tax, D.M.J.; Ypma, A.; Duin, R.P.W. Pump failure determination using support vector data description. In *Advances in Intelligent Data Analysis. IDA*; Hand, D.J., Kok, J.N., Berthold, M.R., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Germany, 1999; Volume 1642, pp. 415–425. [[CrossRef](#)]
27. Shifat, T.A.; Hur, J.-W. ANN Assisted Multi Sensor Information Fusion for BLDC Motor Fault Diagnosis. *IEEE Access* **2021**, *9*, 9429–9441. [[CrossRef](#)]
28. Weatherspoon, M.H.; Langoni, D. Accurate and efficient modeling of FET cold noise sources using ANNs. *IEEE Trans. Instrum. Meas.* **2008**, *57*, 432–437. [[CrossRef](#)]

29. Zhang, Y.; Fu, Y.; Jiang, W.; Li, C.; You, H.; Li, M.; Chandra, V.; Lin, Y. DIAN: Differentiable Accelerator-Network Co-Search Towards Maximal DNN Efficiency. In Proceedings of the 2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Boston, MA, USA, 26–28 July 2021; pp. 1–6. [\[CrossRef\]](#)
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
31. Lei, X.; Pan, H.; Huang, X. A Dilated CNN Model for Image Classification. *IEEE Access* **2019**, *7*, 124087–124095. [\[CrossRef\]](#)
32. Markel, J.D.; Gray, A.H., Jr. *Linear Prediction of Speech*; Springer: New York, NY, USA, 1976.
33. Goswami, J.C.; Chan, A.K. *Fundamentals of Wavelets: Theory, Algorithms, and Applications*, 1st ed.; Wiley: Hoboken, NJ, USA, 2008.
34. Choi, J.Y.; Choi, C.H. Sensitivity analysis of multilayer perceptron with differentiable activation functions. *IEEE Trans. Neural Netw.* **1992**, *3*, 101–107. [\[CrossRef\]](#)
35. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **1980**, *36*, 193–202. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Olshausen, B.A.; Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **1996**, *381*, 607–609. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556v6.
39. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)
40. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5990–5998. [\[CrossRef\]](#)
41. Aggarwal, C.C. *Neural Networks and Deep Learning*; Springer: Cham, Switzerland, 2018.
42. Haykin, S.; Veen, B.V. *Signals and Systems*, 2nd ed.; Wiley: New York, NY, USA, 2003.
43. Hotho, G.; Villemoes, L.F.; Breebaart, J. A Backward-Compatible Multichannel Audio Codec. *IEEE Trans. Audio Speech Lang. Process.* **2008**, *16*, 83–93. [\[CrossRef\]](#)
44. Lin, C.H.; Lin, Y.C.; Tang, P.W. ADMM-ADAM: A New Inverse Imaging Framework Blending the Advantages of Convex Optimization and Deep Learning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5514616. [\[CrossRef\]](#)
45. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 11–13 April 2011; Volume 15, pp. 315–323.
46. Nguyen, A.; Pham, K.; Ngo, D.; Ngo, T.; Pham, L. An Analysis of State-of-the-art Activation Functions For Supervised Deep Neural Network. In Proceedings of the 2021 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam, 26–28 August 2021; pp. 215–220.