Bolt-Loosening Detection Using 1D and 2D Input Data Based on Two-Stream Convolutional Neural Networks

Xiaoli Hou 1, Weichao Guo 1,*, Shengjie Ren 1, Yan Li 1, Yue Si 1 and Lizheng Su 2

1 School of Mechanical and Precision Instrument Engineering, Xi’an University of Technology, Xi’an 710048, China
2 Xi’an Institute of Electronic Engineering, Xi’an 710100, China
* Correspondence: weichaoguo@xaut.edu.cn; Tel.: +86-(0)29-823-12672

Abstract: At present, the detection accuracy of bolt-loosening diagnoses is still not high. In order to improve the detection accuracy, this paper proposes a fault diagnosis model based on the TSCNN model, which can simultaneously extract fault features from vibration signals and time-frequency images and can precisely detect the bolt-loosening states. In this paper, the LeNet-5 network is improved by adjusting the size and number of the convolution kernels, introducing the dropout operation, and building a two-dimensional convolutional neural network (2DCNN) model. Combining the advantages of a one-dimensional convolutional neural network (1DCNN) with wide first-layer kernels to suppress high-frequency noise, a two-stream convolutional neural network (TSCNN) is proposed based on 1D and 2D input data. The proposed model uses raw vibration signals and time-frequency images as input and automatically extracts sensitive features and representative information. Finally, the effectiveness and superiority of the proposed approach are verified by practical experiments that are carried out on a machine tool guideway. The experimental results show that the proposed approach can effectively achieve end-to-end bolt-loosening fault diagnoses, with an average recognition accuracy of 99.58%. In addition, the method can easily achieve over 93% accuracy when the SNR is over 0 dB without any denoising preprocessing. The results show that the proposed approach not only achieves high classification accuracy but also has good noise immunity.

Keywords: bolt connection; fault diagnosis; two-stream convolutional neural networks; anti-noise

1. Introduction

Bolted connections are frequently and widely applied in mechanical equipment because a bolted connection has remarkable advantages, e.g., low cost, high reliability, and easy installation and replacement. However, bolt loosening often occurs due to the fact that the mechanical equipment usually operates under time-varying loads, and bolt loosening normally has a negative impact on the normal operation of equipment such as poor equipment accuracy, significant economic losses, and even accidents. Therefore, developing an effective method for bolt-loosening level recognition is an important task for preventing any accidents in the normal operation of mechanical equipment [1,2]. In the past few decades, the detection of bolt loosening has usually been performed using vibration signals [2,3]. In order to establish the relationship between the mechanical structure and loosening vibration signal, some researchers have extracted time-domain and frequency-domain features and then used various classifiers to detect bolt loosening according to the extracted fault features. The common classifiers for fault diagnosis include k-nearest neighbor (KNN), support vector machine (SVM), k-means clustering, and so on. However, during this kind of detection process, it has been noted that hand-crafted features may miss some distinguishing features about bolt loosening, which means that the classification results are not accurate or reliable [4,5]. For this reason, a method that can automatically extract features is highly sought after.
Fortunately, in recent years, deep learning has developed rapidly and has been successfully applied in speech recognition [6], face recognition [7], computer vision [8], fault diagnosis [9], and other fields because the advantage of deep learning is that it uses unsupervised or semi-supervised feature learning and hierarchical feature extraction algorithms instead of manual feature extraction. Furthermore, another remarkable advantage of deep learning is that it uses multiple-layer artificial neural networks with representation learning, which can extract higher-level features from the raw input data. Simonyan et al. [10] proposed a two-stream convolutional network architecture that incorporates spatial and temporal networks. Guo et al. [11] proposed a novel diagnosis method using a convolutional neural network (CNN) to directly classify the continuous wavelet transform scalogram (CWTS). Wen et al. [12] adopted a new CNN based on LeNet-5 for fault diagnosis, which converts one-dimensional (1D) signals into two-dimensional (2D) images. Although every artificial intelligent method can miss features, deep learning usually automatically extracts features from 2D images and avoids extracting features that heavily rely on human intervention. In the above researches, it was found that scholars mostly converted the original signal into grayscale images or time–frequency images, and then deep learning was used for the fault diagnosis. However, the vibration, pressure, and sound signals measured by the sensors were all one-dimensional time signals in the fault diagnosis. Time-series signals usually contain important features about equipment conditions, such as periodic and short-term pulse features, which may not exist in two-dimensional images. Therefore, one-dimensional time signals are widely used in many fields and produce some remarkable results. For example, in real engineering fields, Yan et al. [13] applied a one-dimensional convolutional neural network (1DCNN) to detect chiller faults and Xiang et al. [14] carried out a study on wind turbine fault detection using 1DCNN and SCADA data analysis. Moreover, Acharya et al. [15] applied 1DCNN to help clinicians detect ECG signals for distinguishing between healthy individuals and those with myocardial infarction. Abdeljaber et al. [16] presented a novel, fast, and accurate structural damage detection system using 1DCNN that had an inherent adaptive design to detect the structural damage of the stand simulator. Peng et al. [17] used 1DCNN to diagnose the vibrations of wheel bearings of high-speed trains and achieved reasonable results. Zhang et al. [18] proposed a novel method called a deep convolutional neural network with wide first-layer kernels (WDCNN), which used raw vibration signals as input and had wide kernels in the first convolutional layer for suppressing high-frequency noise. This novel method showed a good classification ability under different working loads and noise environments. From the above applications, it can be seen that 1DCNN has a strong classification ability in real-time fault diagnosis.

Moreover, a few experts have also applied machine learning methods to diagnose bolt-loosening faults. Kong et al. [3] provided a percussion method based on the power spectral density (PSD) of the sound signal and the decision tree (DT) to identify the bolt-loosening level. Zhang et al. [19] extracted the Mel-frequency cepstral coefficients (MFCCs) of the sound signal as frequency-domain feature parameters, and a support vector machine classifier was trained to effectively detect bolt loosening.

However, at present, there are still many difficulties in the diagnosis of bolt loosening. For instance, the extraction of time-domain and frequency-domain features are still highly dependent on manual intervention. Considering that the manual intervention of feature extraction significantly affects the identification accuracy, it is difficult to extract the bolt-loosening features and construct the loosening feature set. Therefore, the entire process is usually time-consuming for bolt fault diagnoses, to say nothing of automatic detection. Furthermore, in bolt-loosening fault diagnoses, problems such as low model recognition accuracy and effectivity in noisy environments cannot be ignored.

In order to address these problems, a new fault diagnosis method for bolt loosening is proposed based on two-stream convolutional neural networks (TSCNN) simultaneously using one-dimensional signals and two-dimensional images. The feasibility and effectiveness of the method are verified via experimental data. The remainder of this paper is
organized as follows: Section 2 introduces the basic theories of convolution, activation, and pooling, with particular attention to the difference between 1DCNN and 2DCNN. Then, the specific network structure and diagnostic procedure of the TSCNN model are built in Section 3. Section 4 describes the experimental setup, and the effectiveness of the proposed method is demonstrated by comparing it with other diagnostic methods in different noise environments. Finally, some conclusions are drawn in Section 5.

2. Theoretical Background

Convolutional neural networks have fewer parameters and directly use original image data as input data. Thus, manual participation is not required in preprocessing, which means that CNN can be widely applied in image processing. A CNN usually consists of convolution layers, pooling layers, and fully connected layers. According to model structures and performance algorithms, LeNet, ResNet, AlexNet, and GoogleNet are the most popular CNN networks. Compared to the other types of networks, LeNet is applied in many real engineering fields because it has a simple model structure and fewer setting parameters. For instance, Table 1 shows the structure settings for the LeNet-5 network, which is a typical CNN. The classical LeNet-5 network consists of two convolution layers (i.e., Conv1 and Conv2), two pooling layers (i.e., Pooling1 and Pooling2), and three fully connected layers (i.e., FC1, FC2, and FC3) [18].

Table 1. Detailed settings of LeNet-5 network.

| Layer Type | Kernel Size/Stride | Kernel Number | Output Size | Activation Function |
|------------|--------------------|---------------|-------------|--------------------|
| Input      | 32 x 32            | 6             | 6@28 x 28  | Sigmoid            |
| Pooling1   | 2 x 2/2            | 6             | 6@14 x 14  |                    |
| Conv2      | 5 x 5/1            | 16            | 16@10 x 10 | Sigmoid            |
| Pooling2   | 2 x 2/2            | 16            | 16@5 x 5   |                    |
| FC1        | 120                | 1             | 120 x 1    | Sigmoid            |
| FC2        | 84                 | 1             | 84 x 1     | Sigmoid            |
| FC3        | 10                 | 1             | 10         |                    |

2.1. Convolutional Layer

Convolution was applied to check the input local area for convolution operations and generate corresponding features. The most important feature of this layer is weight sharing, which greatly reduces the network parameters of the convolutional layer. If the \( l \)-th layer is assumed as a convolutional layer, its convolution process can be described as follows [18]:

\[
y_{ij}^{l+1} = f \left( \sum_{i=1}^{M} (x_{i}^{l} \otimes k_{ij}^{l}) + b_{ij}^{l} \right),
\]

where the notation \( \otimes \) represents the convolution operation; \( y_{ij}^{l+1} \) is denoted as the input of the \( j \)-th neuron at layer \( l + 1 \); \( f \) represents the activation functions; \( x_{i}^{l} \) is the output of the \( i \)-th neuron at layer \( l \); \( k_{ij}^{l} \) is defined as the kernel from the \( i \)-th neuron at layer \( l \) to the \( j \)-th neuron at layer \( l + 1 \); \( M \) denotes the number of feature maps, which is same as the number of the kernel at layer \( l \); and \( b_{ij}^{l} \) is the bias of the \( j \)-th neuron at layer \( l + 1 \).

Furthermore, in this paper, the Rectified Linear Unit (ReLU) was employed as the activation layer to enhance the divisibility of the extracted features via a nonlinear operation. ReLU has good sparsity compared to the traditional activation functions, Sigmoid and Tanh functions, and more than 50% of neurons in the ReLU are activated. Moreover, the ReLU only needs to add, multiply, and compare operations, which makes calculations efficient and fast. In terms of optimization, the ReLU is a left-saturated function, which
can effectively alleviate the problem of gradient disappearance [20]. The ReLU can be presented as

$$ a_{l+1}^j = f(y_{l+1}^j) = \max(0, y_{l+1}^j) = \begin{cases} y_{l+1}^j, & \text{if } y_{l+1}^j \geq 0 \\ 0, & \text{else} \end{cases} $$

(2)

where $a_{l+1}^j$ is the activation of $y_{l+1}^j$.

2.2. Pooling Layer

The pooling layer in the CNN compresses the feature map. This means that the pooling layer not only extracts the main features but also reduces the parameters of the network. The pooling layer can be divided into max pooling, average pooling, and weighted pooling. The most commonly used pooling layer is the max pooling layer, which is defined as

$$ P_{l+1}^j(i) = \max_{(i-1)W+1 \leq t \leq iW} \{ a_{l}^j(t) \}, $$

(3)

where $P_{l+1}^j(i)$ denotes the corresponding value of the neuron in layer $l+1$ of the pooling operation and $W$ denotes the width of the pooling region.

2.3. Fully Connected Layer

The function of a fully connected (FC) layer is to fully connect all the neurons. Normally, the FC layer is also linked to the output layer and performs as a classifier [21]. The classifier employs a supervised learning algorithm to solve multi-classification problems. The output value is described as follows:

$$ z_{l+1}^j = \sum_{i=1}^{n} w_{ij}^l P_{l}^i + b_j^l, $$

(4)

where $z_{l+1}^j$ represents the output value of the $j$-th neuron at layer $l+1$ and $w_{ij}^l$ is defined as the weight between the $i$-th neuron at layer $l$ and the $j$-th neuron at layer $l+1$.

In the output layer, the softmax function is applied to transform the logits of the neuron into the form of the probability distribution. The normalized probability $q(z_j)$ obtained by the softmax function is described as

$$ q(z_j) = \text{softmax}(z_j) = \frac{\exp(z_j)}{\sum_{n=1}^{N} \exp(z_n)}, $$

(5)

where $z_j$ denotes the logits of the $j$-th output neuron and $N$ is the total number of classes.

2.4. Filter Sliding for 1DCNN and 2DCNN

The input of the 1DCNN is a one-dimensional time series and the data are the vibration signals recorded by the dynamical sensors without any treatment. This guarantees the authenticity of the signals. Moreover, only one-dimensional convolution is employed with fewer parameters, which can significantly reduce computing resources and time. A 1DCNN is frequently applied in natural language processing, sequence models, etc. Remarkably, a 2DCNN performs very well for extracting local features of images, thus it frequently performs in image processing, computer vision, and the like.

The difference between a 1DCNN and a 2DCNN is that the input data dimensions and filter sliding methods are different. As shown in Figure 1, the row of a 1DCNN represents the vibration signal of the measuring point and the column represents the sensor or channel such as the original signal in the $x$, $y$, or $z$ direction. The filter has only 1 degree of freedom to slide along the time series. A one-dimensional filter requires that the width and number of the channels of the filter and data are equal, covering all columns at any time, and its
height determines the length of each convolution operation. However, 2DCNN filters need to slide horizontally, and then vertically move to the next location to continue a horizontal sliding through the entire image. Thus, there are two degrees of freedom, which requires that the number of channels of the filter and data are equal, and its height and width determine the range of each convolution operation. In addition, a 1DCNN and a 2DCNN almost have the same architecture and working principles [22].

![Figure 1](image-url)  
**Figure 1.** The sliding direction of the filter on a 1DCNN and 2DCNN, respectively.

### 3. The Proposed WDCNN Bolt-Loosening Fault Diagnosis Model

Usually, one kind of CNN only permits one kind of input data. However, as mentioned previously, simultaneously using 1D time-varying signals and 2D image data could potentially improve the accuracy of diagnosing bolt-loosening faults. Thus, the frameworks of the 1DCNN and 2DCNN are improved in this section. Afterward, an improved diagnosis fault model for bolt loosening is proposed using the TSCNN.

#### 3.1. The Structure and Parameter Settings Based on an Improved 1DCNN

As shown in Table 2, the improved 1DCNN included five convolutional layers and five pooling layers. Wide kernels in the first convolutional layer (64 × 1) can better suppress high-frequency noise [18], and the multi-layer small convolution kernels (3 × 1) make the network deeper, which helps multi-layer nonlinear mapping and improves network performance [23]. Moreover, the size and number of kernels significantly affect the output accuracy of the network so researchers have carried out many studies by adjusting the kernel settings [24,25].

| Layer Type | Kernel Size/Stride | Kernel Number | Output Size | Activation Function |
|------------|--------------------|---------------|-------------|---------------------|
| Input      | 64 × 1/16          | 8             | 4096 × 1    | ReLU                |
| Conv1      | 2 × 1/2            | 8             | 80253 × 1   | ReLU                |
| Pooling1   | 3 × 1/2            | 16            | 160124 × 1  | ReLU                |
| Conv2      | 2 × 1/2            | 16            | 160126 × 1  | ReLU                |
| Pooling2   | 3 × 1/1            | 32            | 320860 × 1  | ReLU                |
| Conv3      | 2 × 1/2            | 32            | 320830 × 1  | ReLU                |
| Pooling3   | 3 × 1/1            | 64            | 640828 × 1  | ReLU                |
| Conv4      | 2 × 1/2            | 64            | 640814 × 1  | ReLU                |
| Pooling4   | 3 × 1/1            | 64            | 640812 × 1  | ReLU                |
| Conv5      | 2 × 1/2            | 64            | 64086 × 1   | ReLU                |
| Pooling5   | 120                | 1             | 120 × 1     | ReLU                |
| FC1        | 84                 | 1             | 84 × 1      | ReLU                |
| FC2        | 6                  | 1             | 6           |                     |

The pooling layers performed a 2 × 1 max pooling operation after using the ReLU instead of the Sigmoid activation function. Afterward, a two-stream fusion of the fault
information extracted by the 1DCNN and 2DCNN was performed. The FC1 layer was a full-connection layer with 120 neurons, which was fully connected to the output of the two-stream fusion layer and produced 120 feature maps of $1 \times 1$ pixels. The FC2 layer was a full-connection layer with 84 neurons, which calculated the dot-product between the input vector and weight vector and added the bias value, and the results were outputted by the ReLU function. The FC3 layer was the output layer, which had six neurons and divided all the input images into six different categories corresponding to the bolts under six different loosening conditions; the loosening conditions can be seen in the table in Section 4.1. The dropout operation was performed after the FC1 and FC2 layers, and its value was set to 0.2 [26], which effectively prevented overfitting during learning the fault diagnosis model. In addition, using the ReLU function achieved a full backpropagation calculation without causing gradient disappearance.

3.2. The Structure and Parameter Settings Based on an Improved 2DCNN

Before fault diagnosis, the vibration signals needed to be converted into images to match the 2DCNN input. The short-time Fourier transform (STFT), also known as the window Fourier transform, is an extensive and effective time–frequency analysis method. The basic principle is to assume that the signal is stable for a short period, and the time-domain signal can be divided into a limited number of small segments. Afterward, the time-varying frequency spectrum of the entire signal is produced by continuously sliding the time window to the end of the signal via Fourier transform [27]. As shown in Figure 2, the original signal was divided into multiple data samples with oversampling, and STFT was used to obtain the time–frequency characterization of each sample, which was beneficial to the generalization ability of the model. The STFT can be presented as

$$STFT(t, \omega) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j\omega \tau}d\tau,$$

where $t$ is the shift factor and $\tau$ is the time position center of the window function $w(t)$.

![Figure 2. Signal preprocessing process.](image)

The network structure of the improved 2DCNN included four convolution layers and four pooling layers, also using the ReLU activation function and dropout operation. In the
training of the bolt-loosening fault diagnosis model, it was found that with the increase in the image size, the diagnosis accuracy increased but the training speed became slower. Considering the fault diagnosis accuracy and training speed, the final time–frequency image size was set to $64 \times 64$. The detailed settings are listed in Table 3.

Table 3. Detailed settings of the 2DCNN.

| Layer Type | Kernel Size/Stride | Kernel Number | Output Size | Activation Function |
|------------|--------------------|---------------|-------------|---------------------|
| Input      |                    |               | $64 \times 64$ |                     |
| Conv1      | $5 \times 5/1$     | 8             | $8@60 \times 60$ | ReLU                |
| Pooling1   | $2 \times 2/2$     | 8             | $8@30 \times 30$ | ReLU                |
| Conv2      | $3 \times 3/1$     | 16            | $16@28 \times 28$ | ReLU                |
| Pooling2   | $2 \times 2/2$     | 16            | $16@14 \times 14$ | ReLU                |
| Conv3      | $3 \times 3/1$     | 32            | $32@12 \times 12$ | ReLU                |
| Pooling3   | $2 \times 2/2$     | 32            | $32@6 \times 6$ | ReLU                |
| Conv4      | $3 \times 3/1$     | 64            | $64@4 \times 4$ | ReLU                |
| Pooling4   | $2 \times 2/2$     | 64            | $64@2 \times 2$ | ReLU                |
| FC1        | 120                | 1             | $120 \times 1$ | ReLU                |
| FC2        | 84                 | 1             | $84 \times 1$ | ReLU                |
| FC3        | 6                  | 1             | 6            |                     |

Compared with the traditional LeNet-5 network, the improved 2DCNN and 1DCNN mainly had the following five advantages.

1. The size of the input layer was changed. The improved 1DCNN inputs were the original vibration signals of $4096 \times 1$, and the improved 2DCNN inputs were the time–frequency images with a size of $64 \times 64$.

2. The number of convolutional layers and pooling layers was increased. Theoretically, the deeper the neural network, the stronger the feature expression ability. However, more convolutional layers and pooling layers easily lead to network degradation. Finally, five convolutional layers and five pooling layers were used in the improved 1DCNN, and four convolution layers and four pooling layers were used in the improved 2DCNN.

3. The size and number of the convolution kernels were changed. Due to the complexity of the vibration signals, the size and number of the convolution kernels needed to be adjusted to enhance the network’s classification ability. The adjusted convolution kernel settings are shown in Tables 2 and 3.

4. The dropout operation was used, which effectively reduced overfitting during the training of the fault diagnosis model.

5. The ReLU activation function was adopted. The ReLU activation function prevented gradient disappearance and the training speed was faster than the Sigmoid activation function.

3.3. The Bolt Diagnosis Model Based on the TSCNN

Figure 3 shows the fault diagnosis process for bolt loosening based on the TSCNN model. As shown in Figure 3, the inputs of this network were a vibration signal and time–frequency image, the features extracted by the 1DCNN and 2DCNN were fused with the Softmax classification, and then the bolt-loosening fault diagnosis was performed. The diagnosis process is described as follows:

Step 1: The vibration signals of bolt loosening were collected by the sensors deployed on the guideway of the machine tool. The training samples were dealt with by oversampling in order to reduce the effect of the overfitting problem.

Step 2: The preprocessed vibration signals in Step 1 were converted into time–frequency images with STFT, and the sizes of the images were set to $64 \times 64$.

Step 3: The original vibration signal was divided into training set 1 and test set 1 with a ratio of 8:2. At the same time, the time–frequency images were also divided into training set 2 and test set 2 with the same ratio.
Step 4: The bolt-loosening fault diagnosis model was trained using training sets 1 and 2.

Step 5: The bolt-loosening fault diagnosis model was employed to identify the test set online. The validity of the diagnosis model was evaluated via the identified results, and the network parameters were optimized in order to obtain a reliable and effective bolt-loosening fault diagnosis model.

Step 6: Real bolt conditions were recognized by the optimized bolt-loosening fault diagnosis model and some equipment maintenance advice was provided according to the recognition results.

![Figure 3. The structure illustration of the TSCNN model.](image)

3.4. Model Evaluation Index

After the bolt loosening, the fault diagnosis model was well trained, and the recognition accuracies and loss value indexes were observed to evaluate the quality of the diagnosis model. The accuracy rate represents the ratio of the number of correct predictions to the total number. The loss function of the model is the cross-entropy between the estimated softmax output probability distribution and the target class probability distribution. The Loss value was smaller and the accuracy of the diagnostic model was higher. The formula is as follows:

\[
\text{Loss} = -\sum_x p(x) \log q(x),
\]

where \(p(x)\) denotes the target distribution and \(q(x)\) denotes the estimated distribution.

4. Experiment and Analysis

The TSCNN model was implemented in MATLAB 2020a and Pytorch 1.5.1. The TSCNN model was trained on a Windows 10 computer with a Hexa-core Intel i5-8400H CPU at 2.8 GHz and 16 GB RAM. All experiments were conducted using adaptive moment estimation (Adam), where the initial learning rate was set to 0.008 by comprehensively considering the classification accuracy and convergence speed. It was important to select the right batch size during the training and testing. According to our previous work, both batch sizes were set to 128.
4.1. Experimental Setup

The experiment for recognizing bolt loosening under the same working conditions was carried out on a machine tool guideway (size: 1500 mm × 130 mm × 50 mm), which is illustrated in Figure 4. Twelve Q235 T-bolts were arranged on both sides of the guideway. Different loosening conditions in the six bolts were analyzed due to the symmetry of the bolt layout on the guideway. For convenience, the six bolts were numbered and the details can be seen in Figure 4. Different torques of bolts were set via a digital torque wrench (SATA 96525) to simulate different loosening failures. One acceleration sensor (PCB 333B30) was glued to the guideway and connected to the data acquisition instrument (LMS SCADAS Mobile) through connection line to collect vibration data of different cases with a sampling rate of 12.8 kHz. The sensitivity of the acceleration sensor was 103.4 mv/g. The LMS data acquisition instrument was connected to the laptop by the network cable. The standard torque for the grade 8.8 M12 steel bolt was 80 Nm. Therefore, the bolts were torqued to 80 Nm by a digital torque wrench. The bolts in this condition were considered to be in a healthy state without any loosening.

![Experimental setup for bolt loosening.](image)

Table 4 shows that six kinds of bolt looseness were set in this experiment, such as T1 (all bolts tightened with torques of 80 Nm), T2 (only bolt 1 was severely loose with a torque of 0 Nm), T3 (bolt 1 and 2 were severely loose with torques of 0 Nm), T4 (all bolts were slightly loose with torques of 60 Nm), T5 (all bolts were moderately loose with torques of 40 Nm), and T6 (all bolts were severely loose with torques of 0 Nm). At each torque level, five assemble–disassemble–repeat tests were performed to validate the robustness and repeatability of the proposed method. Each sample contained 4096 points and the details of all the datasets are also depicted in Table 4.

Table 4. Detailed arrangement of experimental cases.

| Case | Looseness Extent | Torque (Nm) | Training/Test Dataset |
|------|------------------|-------------|-----------------------|
| T1   | All bolts tightened | 80          | 160/40                |
| T2   | Bolt 1 severely loose | 0           | 160/40                |
| T3   | Bolt 1 and 2 severely loose | 0   | 160/40                |
| T4   | All bolts slightly loose | 60         | 160/40                |
| T5   | All bolts moderately loose | 40   | 160/40                |
| T6   | All bolts severely loose | 0   | 160/40                |

4.2. Training and Verification of the Bolt Fault Diagnosis Model

In order to demonstrate the effectiveness of the proposed method in this paper, the fault diagnosis model based on the TSCNN model was trained and tested with the dataset listed in Table 4. The total computing time was about 900 s and ran on the laptop mentioned in the previous section. The accuracy and loss curves of the bolt fault diagnosis model were depicted.
Table 4. Detailed arrangement of experimental cases.

| Case | Looseness Extent                  | Torque (Nm) | Training/Test Dataset |
|------|----------------------------------|-------------|-----------------------|
| T1   | All bolts tightened              | 80          | 160/40                |
| T2   | Bolt 1 severely loose            | 0           | 160/40                |
| T3   | Bolt 1 and 2 severely loose      | 0           | 160/40                |
| T4   | All bolts slightly loose         | 60          | 160/40                |
| T5   | All bolts moderately loose       | 40          | 160/40                |
| T6   | All bolts severely loose         | 0           | 160/40                |

4.2. Training and Verification of the Bolt Fault Diagnosis Model

In order to demonstrate the effectiveness of the proposed method in this paper, the fault diagnosis model based on the TSCNN model was trained and tested with the dataset listed in Table 4. The total computing time was about 900 s and ran on the laptop mentioned in the previous section. The accuracy and loss curves of the bolt fault diagnosis model obtained during the training and test phases are shown in Figures 5 and 6, respectively.

Figure 5. Accuracy curve of the fault diagnosis model based on the TSCNN.

Figure 6. Loss function curve of the fault diagnosis model based on the TSCNN.

As shown in Figure 5, during the training and test phases of the TSCNN fault diagnosis model, it can be seen that the initial diagnosis accuracy was continuously improved, and the fault classification accuracy stabilized after about 70 iterations. When the iteration reached 100, the diagnosis accuracy could reach up to 99.58%. On the other hand, Figure 6 shows that the loss function value decreased rapidly in the first 70 iterations, but the loss function values of the test dataset were much high than the values for the training dataset.
This is because the extracted fault features for the test dataset and the training dataset were not exactly the same. However, with the evolution of the TSCNN model, the loss function curves for the test dataset and training dataset decreased rapidly and both curves tended to zero after 70 iterations. This indicates that the TSCNN model had high accuracy, fast convergence speed, and no overfitting.

Figure 7 clearly shows the classification effect of the fault diagnostic model on the test set. The x-axis represents 240 sets of test data and the y-axis represents six types of faults. It can be seen that most test sets were classified correctly except for T2. One of the test samples T2 was wrongly grouped with T4, i.e., the loosening of bolt 1 was incorrectly classified as the loosening of bolts 1 and 2. Thus, the diagnostic accuracy of the fault diagnostic model was 97.5% in recognizing T2 faults.

Figure 7. Prediction results on the test set.

In order to verify the diagnosis performance of the proposed method, we compared it with the CNN-SVM and AlexNet algorithms.

Table 5 describes the classification accuracy of bolt loosening under different diagnosis models. As seen in Table 5, for the CNN-SVM diagnosis model, the diagnosis accuracies of the six failure types were 92.5%, 95%, 97.5%, 75%, 100%, and 90%, respectively. For the fine-tuned AlexNet diagnosis model, the recognition accuracies were 100%, 90%, 92.5%, 90%, 95%, and 95%, respectively. For the TSCNN diagnosis model, the recognition accuracies reached 100%, 97.5%, 100%, 100%, 100%, and 100%, respectively.

Table 5. Results of different bolt-loosening faults based using the proposed TSCNN compared with CNN-SVM and AlexNet.

| Fault Diagnosis Model | Diagnosis Accuracy (%) |
|-----------------------|------------------------|
|                       | T1  | T2  | T3  | T4  | T5  | T6  | Average  |
| CNN-SVM               | 92.5| 95  | 97.5| 75  | 100 | 90  | 91.67    |
| AlexNet               | 100 | 90  | 92.5| 90  | 95  | 95  | 93.75    |
| TSCNN                 | 100 | 97.5| 100 | 100 | 100 | 100 | 99.58    |

Although the AlexNet and LeNet networks had similar network structures, compared to LeNet, AlexNet utilized the GPU to treat the matrix operation, which accelerated the CNN model training. Hence, AlexNet had obvious advantages in dealing with a large data problem. AlexNet also performed well in a large and deep network. This means that AlexNet could extract more kinds of features and had a good generalization ability. However, it could see that the average accuracy of the TSCNN diagnosis model was the highest because the average accuracies for the three fault diagnosis models were 91.67%,
93.75%, and 99.58%, respectively. Therefore, it can be said that the proposed TSCNN was more effective for diagnosing the different loosening fault states for bolt connections compared to the other two models.

4.3. Performance under Noise Environment

The working conditions of the bolt structure are accompanied by varying degrees of noise in actual industrial applications. In order to further verify the applicability of the proposed TSCNN model, we added white Gaussian noise to the original vibration signals in the following cases and introduced a signal-to-noise ratio \( (SNR_{dB}) \) to evaluate the level of white Gaussian noise. The definition of \( SNR_{dB} \) is described as follows:

\[
SNR_{dB} = 10 \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) = 10 \log_{10}\left(\frac{A_{signal}}{A_{noise}}\right),
\]

where the subscript dB means that the signal-to-noise ratio is expressed in decibels. \( P_{signal} \) and \( A_{signal} \) are the power and amplitude of the signal, and \( P_{noise} \) and \( A_{noise} \) are the power and amplitude of the noise, respectively.

Figure 8 shows that the original signal of the bolt 2 loosening fault was added with the white Gaussian noise. The SNR for the composite noisy signal was 0 dB, which means that the power of the noise was equal to that of the original signal. To verify the anti-noise ability of the proposed model, we tested the CNN-SVM, AlexNet, and proposed TSCNN models with five different SNR levels, i.e., SNRs in the range of \(-4\) dB, \(-2\) dB, 0 dB, 2 dB, and 4 dB.

![Figure 8](https://example.com/figure8.png)

**Figure 8.** Illustration of the original signal of bolt 2 loosening fault with white Gaussian noise. The composite noise signal with SNR = 0 dB. (a) Original signal, Gaussian noise, and noise signal; (b) Zoom in the three signals to the range [1000, 1200].

The fault diagnosis results of the three diagnosis models in noisy environments with different SNR levels are shown in Figure 9. It can be seen that following the increase in the SNR, the recognition accuracies of the three diagnosis models also increased. It should be noted that the recognition accuracies of the TSCNN model were always above 90% for the different SNR levels. Even at the highest noise level, for instance, \( SNR = -4 \), the recognition accuracy of the TSCNN model was 92%. At the same SNR level, the diagnosis accuracy of the CNN-SVM and AlexNet models was less than 50%. Although the SNR was larger than 0 dB, the recognition accuracy of the TSCNN model reached 98.75%. In contrast, the fault diagnosis accuracies of the CNN-SVM and AlexNet models were still below 85%. Hence, it can be seen that the TSCNN model performed better than the CNN-SVM and AlexNet models for bolt-loosening diagnoses even with high noise.
Author Contributions: Conceptualization, methodology, W.G. and L.S.; writing—original draft, review, and editing, X.H. and S.R.; writing—review and editing, W.G.; validation, formal analysis, data curation, resources, investigation, S.R. and Y.S.; project administration, review and editing, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China under Grant No. 51905421 and Grant No. 52075437, by the Science and Technology Activities Selection of Shaanxi Overseas Students under Grant 302/253081605, and by the Key Scientific Research Project of Shaanxi Provincial under Grant 20JY047 and 2022GY-211.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Figure 9. Diagnosis accuracies of the three diagnosis models in different noise environments.

5. Conclusions

This paper proposes a new fault diagnosis model based on the TSCNN model to address the bolt-loosening fault diagnosis problem. The TSCNN model simultaneously extracts fault features from vibration signals and time–frequency images to detect the bolt connection state. The diagnosis ability of the model is verified by the experiment and is compared to the other two fault diagnosis models. The detailed conclusions are as follows:

1) The dynamic characteristics of the structure change with the loosening of the bolts on the machine tool guideways and the vibration signal and its frequency spectrum also change. In this paper, the TSCNN model is developed to simultaneously integrate the fault features extracted by the 1DCNN and 2DCNN, which can accurately represent bolt-loosening situations.

2) The CNN is improved by applying wide kernels in the first convolutional layer (64 × 1) and successive small convolution kernels (3 × 1). Wide kernels can suppress high-frequency noise and successive small convolution kernels lead to deeper multilayer nonlinear mapping, which enhances the fault classification capability.

3) Adopting the ReLU activation function and performing the dropout operation can significantly improve the convergence speed and generalization ability of the model.

4) The TSCNN model effectively avoids manual feature extraction and low-efficiency post-processing. The ability to recognize different degrees of bolt-loosening situations is assessed via real experiments and compared to the other two fault diagnosis models. The results show that the average recognition accuracies of the TSCNN model can reach 99.58% and easily achieve over 93% once the SNR is over 0 dB without any denoising preprocessing. Hence, it can be said that the TSCNN model can achieve high classification accuracy for bolt-loosening fault diagnoses and is also very effective and robust even in a high-noise environment.
Data Availability Statement: Data are contained within the article. In addition, the data presented are available on request from the corresponding author.

Acknowledgments: The authors are grateful to the Collaborative Innovation Center for Green Manufacturing of Modern Equipment, which provided the machine tool guideway and the data acquisition instrument (LMS SCADAS Mobile) with all acceleration sensors.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References
1. Wang, F.; Ho, S.C.M.; Song, G. Modeling and analysis of an impact-acoustic method for bolt looseness identification. Mech. Syst. Signal Process. 2019, 133, 106249. [CrossRef]
2. Sun, W.; Guan, Z.; Zeng, Y.; Pan, J.; Gao, Z. Bolt Loosening Detection of Rocket Connection Structure Based on Variational Modal Decomposition and Support Vector Machines. Appl. Sci. 2022, 12, 6266. [CrossRef]
3. Kong, Q.; Zhu, J.; Ho, S.C.M.; Song, G. Tapping and listening: A new approach to bolt looseness monitoring. Smart Mater. Struct. 2018, 27, 07LT02. [CrossRef]
4. Shao, H.; Jiang, H.; Lin, Y.; Li, X. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. Mech. Syst. Signal Process. 2018, 102, 278–297. [CrossRef]
5. Gong, W.; Chen, H.; Zhang, Z.; Zhang, M.; Wang, R.; Guan, C.; Wang, Q. A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion. Sensors 2019, 19, 1693. [CrossRef]
6. Noda, K.; Yamaguchi, Y.; Nakadai, K.; Okuno, H.G.; Ogata, T. Audio-visual speech recognition using deep learning. Appl. Intell. 2014, 42, 722–737. [CrossRef]
7. Nagpal, S.; Singh, M.; Singh, R.; Vatsa, M. Regularized deep learning for face recognition with weight variations. IEEE Access 2015, 3, 3010–3018. [CrossRef]
8. Nie, S.; Zheng, M.; Ji, Q. The deep regression bayesian network and its applications: Probabilistic deep learning for computer vision. IEEE Signal Process. Mag. 2018, 35, 101–111. [CrossRef]
9. Jia, F.; Lei, Y.; Lu, N.; Xing, S. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. Mech. Syst. Signal Process. 2018, 110, 349–367. [CrossRef]
10. Simonyan, K.; Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
11. Guo, S.; Yang, T.; Gao, W.; Zhang, C. A novel fault diagnosis method for rotating machinery based on a convolutional neural network. Sensors 2018, 18, 1429. [CrossRef]
12. 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]
13. Yan, K.; Zhou, X. Chiller faults detection and diagnosis with sensor network and adaptive 1D CNN. Digit. Commun. Netw. 2022, 8, 531–539. [CrossRef]
14. Xiang, L.; Wang, P.; Yang, X.; Hu, A.; Su, H. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. Measurement 2021, 175, 109094. [CrossRef]
15. Acharya, U.R.; Fujita, H.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adam, M. Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. Inf. Sci. 2017, 415–416, 190–198. [CrossRef]
16. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. J. Sound Vib. 2017, 388, 154–170. [CrossRef]
17. Peng, D.; Liu, Z.; Wang, H.; Qin, Y.; Jia, L. A novel deeper one-dimensional cnn with residual learning for fault diagnosis of wheelset bearings in high-speed trains. IEEE Access 2019, 7, 10278–10293. [CrossRef]
18. 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 Sensors. Sensors 2017, 17, 425. [CrossRef]
19. Zhang, Y.; Zhao, X.; Sun, X.; Su, W.; Xue, Z. Bolt loosening detection based on audio classification. Adv. Struct. Eng. 2019, 22, 2882–2891. [CrossRef]
20. Min, X.; Clarence, W.D.S. Gear transmission fault classification using deep neural networks and classifier level sensor fusion. Instrumentation 2019, 9, 101–109. [CrossRef]
21. Swapna, P.; Jayasanka, R.; Lalith, B.G. An improved machine learning approach to classify sleep stages and apnea events. Instrumentation 2019, 6, 30–40. [CrossRef]
22. Khan, S.; Yairi, T. A review on the application of deep learning in system health management. Mech. Syst. Signal Process. 2018, 107, 241–265. [CrossRef]
23. Lopez-Martín, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Shallow neural network with kernel approximation for prediction problems in highly demanding data networks. Expert Syst. Appl. 2019, 124, 196–208. [CrossRef]
24. Najafzadeh, M.; Tafarzolouz, A. Evaluation of neuro-fuzzy GMDH-based particle swarm optimization to predict longitudinal dispersion coefficient in rivers. Environ. Earth Sci. 2016, 75, 1–12. [CrossRef]
25. Najafzadeh, M.; Rezaie-Balf, M.; Tafarojnoruz, A. Prediction of riprap stone size under overtopping flow using data-driven models. *Int. J. River Basin Manag.* 2018, *16*, 505–512. [CrossRef]

26. Lu, S.; Lu, Z.; Zhang, Y.-D. Pathological brain detection based on AlexNet and transfer learning. *J. Comput. Sci.* 2019, *30*, 41–47. [CrossRef]

27. Wang, J.; Mo, Z.; Zhang, H.; Miao, Q. A deep learning method for bearing fault diagnosis based on time-frequency image. *IEEE Access* 2019, *7*, 42373–42383. [CrossRef]