buildings

MDPI

Article

Multimodal Deep Neural Network-Based Sensor Data Anomaly Diagnosis Method for Structural Health Monitoring

Xingzhong Nong [1], Xu Luo [1,*], Shan Lin [1], Yanmei Ruan [1] and Xijun Ye [2]

[1] Guangzhou Metro Design & Research Institute Co., Ltd., Guangzhou 510006, China; nongxingzhong@gmdi.cn (X.N.); linshan@gmdi.cn (S.L.); ruanyanmei@gmdi.cn (Y.R.)
[2] School of Civil Engineering, Guangzhou University, Guangzhou 510006, China; xijun_ye@gzhu.edu.cn
* Correspondence: luolicn2003@163.com

Abstract: Due to sensor failure, noise interference and other factors, the data collected in the structural health monitoring (SHM) system will show a variety of abnormal patterns, which will bring great uncertainty to the structural safety assessment. This paper proposes an automatic data anomaly diagnosis method for SHM based on a multimodal deep neural network. In order to improve the detection accuracy, both two-dimensional and one-dimensional features of the sensor data are fused in the multimodal deep neural network. The network consists of two convolutional neural network (CNN) channels, one a 2D-CNN channel for extracting time–frequency features of sensor data and the other a 1D-CNN channel for extracting raw one-dimensional features of sensor data. After convolution and pooling operations for the sensor data by the 2D channel and 1D channel separately, the two types of extracted features are flattened into one-dimensional vectors and concatenated at the concatenation layer. The concatenated vector is then fed into fully connected layers for final SHM data anomaly classification. In order to evaluate the reliability of the proposed method, the monitored data lasting for one month of a long-span cable-stayed bridge were used for training, validation, and testing. Six types of training conditions (missing, minor, outlier, over-range oscillation, trend, and drift) are studied and analyzed to address the issue of imbalanced training data. With an accuracy rate of 95.10%, the optimal model demonstrates the effectiveness and capability of the proposed method. The proposed method shows a promising future as a reliable AI-assisted digital tool for safety assessment in structural health monitoring systems.

Keywords: deep learning; data anomaly diagnosis; structural health monitoring; multimodal deep neural network

check for updates

Citation: Nong, X.; Luo, X.; Lin, S.; Ruan, Y.; Ye, X. Multimodal Deep Neural Network-Based Sensor Data Anomaly Diagnosis Method for Structural Health Monitoring. *Buildings* **2023**, *13*, 1976. https://doi.org/10.3390/buildings13081976

Academic Editors: Shiping Huang, Mohammad Fotouhi, Majid Ilchi Ghazaan, Yuxin Pan and Armin Dadras Eslamlou

Received: 29 June 2023
Revised: 26 July 2023
Accepted: 30 July 2023
Published: 2 August 2023

Copyright: © 2023 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

Structural health monitoring (SHM) involves assessing structural loads, responses, and real-time performance to predict the future behavior of structures. Currently, SHM systems have gained extensive utilization in the field of building structures, and numerous scholars have undertaken research in this area. Yang et al. [1] directed their focus towards the design and implementation of a long-term SHM system specifically tailored for heritage timber buildings. Their approach encompassed the deployment of 104 sensors, comprising 6 different types, to effectively monitor diverse environmental factors and structural responses. Li et al. [2] employed a fusion of convex–concave hull and support vector machine methodologies to accurately identify various data stream types within the context of SHM. They conducted experiments using a laboratory-scale prototype building as the testbed. On a related note, Aytulun et al. [3] dedicated their research efforts towards the comprehensive assessment of tall buildings in Istanbul by means of long-term monitoring and modal identification, with a particular focus on seismic events. This investigation necessitated the analysis of data acquired from pre-existing SHM systems.

SHM systems comprise diverse types of sensors that facilitate various monitoring functions, including measurements of acceleration, strain, displacement, humidity, temperature, and wind speed. Among these, accelerometers are widely used in SHM systems. However, due to the substantial amount of data generated by SHM systems, data anomalies are prone to occur. These anomalies can stem from sensor or system failures, as well as environmental factors. Consequently, there is a pressing need for effective algorithms that can cleanse or identify such anomalies, enabling more reliable monitoring and subsequent analysis of SHM data.

Extensive research has been conducted by scholars worldwide to address the issue of data anomalies in SHM systems. The approaches employed primarily revolve around model-based prediction methods and data-driven methods. Model-based prediction methods involve using statistical and mechanical models to predict measurement results. By comparing the measured values with the predicted values, any data points that deviate significantly from the predictions are considered anomalies. For instance, Maher Abdelghani and Michael lan Friswell [4,5] proposed a method that utilizes residuals between measured and predicted responses, obtained from state-space equations, to detect and isolate sensor anomalies, and its effectiveness was successfully validated. Yuen et al. [6] introduced a probabilistic approach based on Bayesian inference to quantify the probability of data points being outliers, thereby detecting abnormal values. Thiyagarajan [7] employed the autoregressive integrated moving average (ARIMA) model to predict sensor data based on sparse historical time-series data. Wan et al. [8] utilized a Gaussian process-based Bayesian method to establish a model for detecting anomalies in structural health monitoring systems. Zhang et al. [9] proposed a real-time detection method for data anomalies based on Bayesian dynamic linear models.

However, these model-based prediction methods may not be applicable for handling data from typical, large-scale structural health monitoring systems. As the complexity and uncertainty of the systems increase, finding explicit models and their corresponding parameters becomes increasingly challenging. Moreover, these methods cannot handle differences among anomalous values and can only label them as either normal or abnormal. In contrast, data-driven methods have the advantage of directly analyzing the data for anomaly detection. For example, Kerschen et al. [10] utilized principal component analysis to compute the principal angle between reference data (normal data) and the test data. When the principal angle exceeds the control upper limit, defined as the average angle of the reference data plus three times its standard deviation, an alarm is triggered, and the sensor associated with the anomaly is flagged and isolated. Goebel et al. [11] employed fuzzy sensor technology and automatic parameter optimization to identify and correct sensor drift and intermittent failure issues. Kramer [12] utilized self-organizing neural networks for anomaly detection. By compressing the data information through self-organizing neural networks, a correlation model of the input data is obtained and used for data filtering, eliminating random errors and erroneous data. Tamilselvan et al. [13] implemented automatic detection and isolation of abnormal data using deep belief networks, and validated their approach using aircraft engine and power transformer data. Monica Arul et al. [14] proposed an abnormal data detection method based on Shapelet transform. It captures the unique time-series shapes of each type of abnormal data using the Shapelet transform, and trains a random forest classifier with these unique time-series shapes. Finally, the trained classifier is used to identify various types of abnormal data.

Since the groundbreaking victory of AlexNet [15] in the 2012 ImageNet competition, deep learning technology has captured the attention of researchers worldwide. Excellent networks like VGG-Net and ResNet have been introduced as a result. The widespread adoption of deep learning techniques can be observed in various engineering fields. Gibert et al. [16] successfully combined computer vision techniques with convolutional neural networks to automate regular inspections of railway tracks. Kim et al. [17] applied transfer learning techniques to the AlexNet convolutional neural network, enabling intelligent detection of concrete cracks. Their trained model can identify various types of

cracks. Li et al. [18] employed convolutional neural networks to detect structural cracks in civilian buildings, offering an alternative to certain manual on-site inspection tasks. In the field of data anomaly detection, Maya et al. [19] utilized long short-term memory (LSTM) networks to detect data anomalies. Their approach involved training a prediction model on normal data and detecting anomalies based on the prediction errors of measured data. Chen et al. [20] proposed a Bayesian deep learning-based anomaly detection model for satellite telemetry data. Inspired by computer vision techniques, Bao et al. [21] introduced a method that transforms raw time-series data into image vectors and utilizes deep neural networks to classify various anomalies in structural health monitoring systems. Tang et al. [22] proposed using convolutional neural networks to learn anomaly detection from multiple image information.

Nevertheless, classifiers for deep learning-based data anomaly detection still face challenges such as imbalanced training data classes and incomplete representation of abnormal patterns. To address these challenges, Mao et al. [23] proposed using generative adversarial networks (GAN) combined with deep autoencoders to identify abnormal data. Zhang et al. [24] developed a one-dimensional CNN-based method for data anomaly diagnosis, and they tackled the data imbalance issue by employing data augmentation techniques. They demonstrated the effectiveness of their method by detecting anomalies in accelerometer data from a large-span bridge. For real-time data anomaly detection, Maleki et al. [25] devised an enhanced training algorithm using deep long short-term memory (LSTM) autoencoders to differentiate anomalies in unlabeled time-series data.

Overall, considerable progress has been made by both domestic and international scholars in the field of data anomaly detection and repair. However, several pressing issues still require attention. Firstly, structural health monitoring (SHM) systems often generate a vast volume of data. Dealing with such a large number of data poses significant challenges in terms of identifying relevant parameters and establishing appropriate models. Furthermore, determining the specific types of data anomalies becomes an intricate task. Secondly, current research on data anomaly diagnosis using deep learning techniques often approaches the problem by transforming it into a computer vision classification task. However, there is a noticeable dearth of studies focusing on the utilization of one-dimensional convolutional neural networks for classification based on vibration signals. Therefore, it is of utmost importance to address the current research gaps in structural health monitoring. Specifically, the key areas of concern involve developing automated techniques for data analysis and feature extraction from large-scale data, enabling effective data anomaly detection. Additionally, further investigation into data anomaly diagnosis methods that integrate vibration signals with deep learning techniques holds significant promise and merits comprehensive exploration.

This paper proposes a data anomaly detection algorithm based on a dual-channel convolutional neural network. The algorithm is designed to analyze and process a vast amount of data in a structural health monitoring system, aiming to identify normal and abnormal data and conduct in-depth analysis of the anomalies. In Section 2, the data anomaly detection framework is introduced. Section 3 depicts the dataset and corresponding data preprocessing technique. The analysis results of the framework are displayed in Section 4. Section 5 is the conclusions.

2. Proposed Framework

As discussed in the introduction, the effectiveness of different sorts of CNNs has been proven [26–29]. Nevertheless, the data collected during SHM often contain not only information about structural damage but also environmental and electromagnetic noises. As a result, the accuracy of damage detection is compromised, as the noise interferes with the detection of damage-related features in the signals. To overcome this challenge and enhance damage detection, the information fusion technique was applied in this research for its ability to extract features even from noisy data [24,30]. This technique has proven to be more effective than simply increasing the depth of the neural network [31,32].

Therefore, a two-channel form of CNN was proposed, which combines features from both the time domain (1D channel) and the time–frequency domain (2D channel), as displayed in Figure 1. The framework involves visualizing the labeled original time-series data using wavelet transform to gain insights into their characteristics. Subsequently, a dual-channel input CNN is designed and trained for data anomaly classification, aiming to learn the description and performance of the data. By leveraging the well-trained model, various types of anomalies can be automatically detected in test data sets, enabling effective identification based on the learned patterns and features.

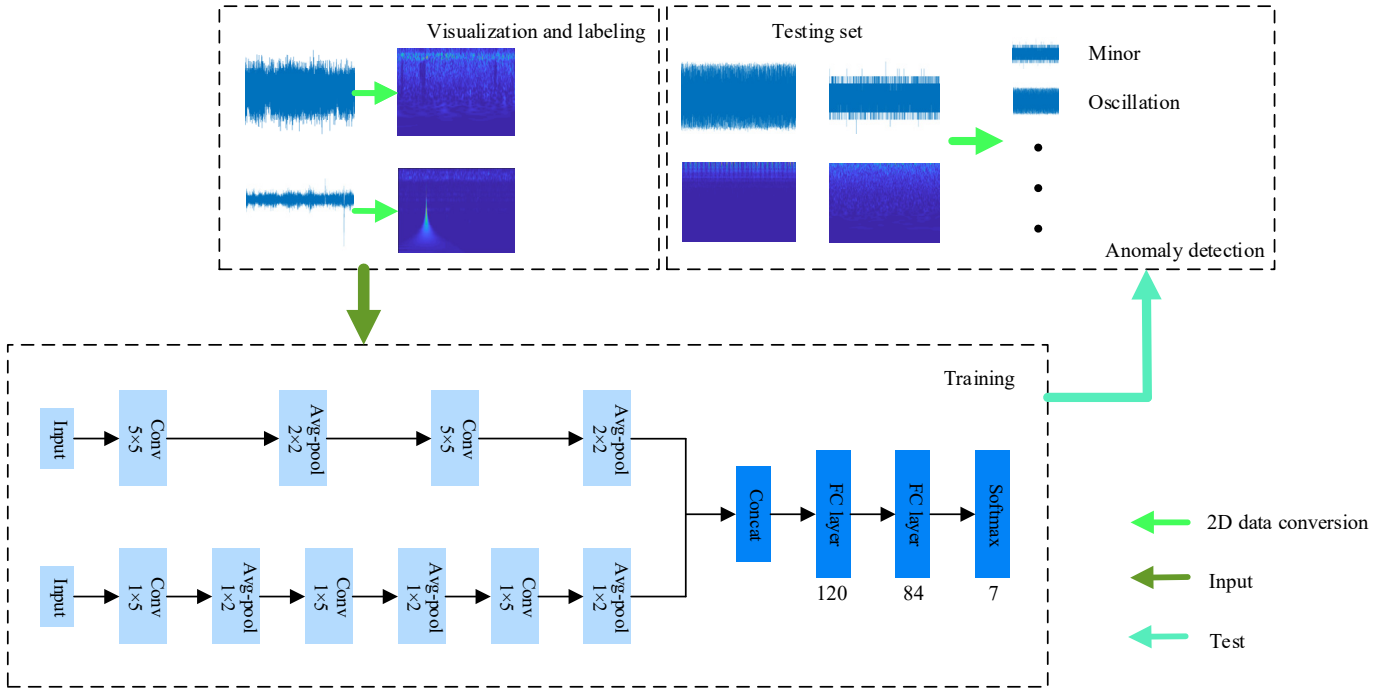


Figure 1. The proposed multimodal fusion deep learning network for data anomaly detection.

To be more intuitive, the specific steps of the proposed framework are shown in Figure 2. In essence, this study involves the acquisition of signals from a bridge-structure health monitoring system, to establish a comprehensive database through data preprocessing techniques. Subsequently, the obtained database is partitioned into training, validation, and testing subsets. A dual-channel convolutional neural network architecture is constructed, and initial network parameters are determined. The classification model is trained, and the error between the predicted values and the ground-truth values is computed. The convergence of the network is evaluated by assessing the error, and in the case of non-convergence, a backpropagation algorithm is executed. This algorithm progressively updates the weight parameters at each layer and recalculates the error between the predicted and actual values until convergence is achieved. Once convergence is confirmed, the optimal network model is selected and preserved. The selected model, evaluated using the testing subset, serves to generate data anomaly detection outcomes.

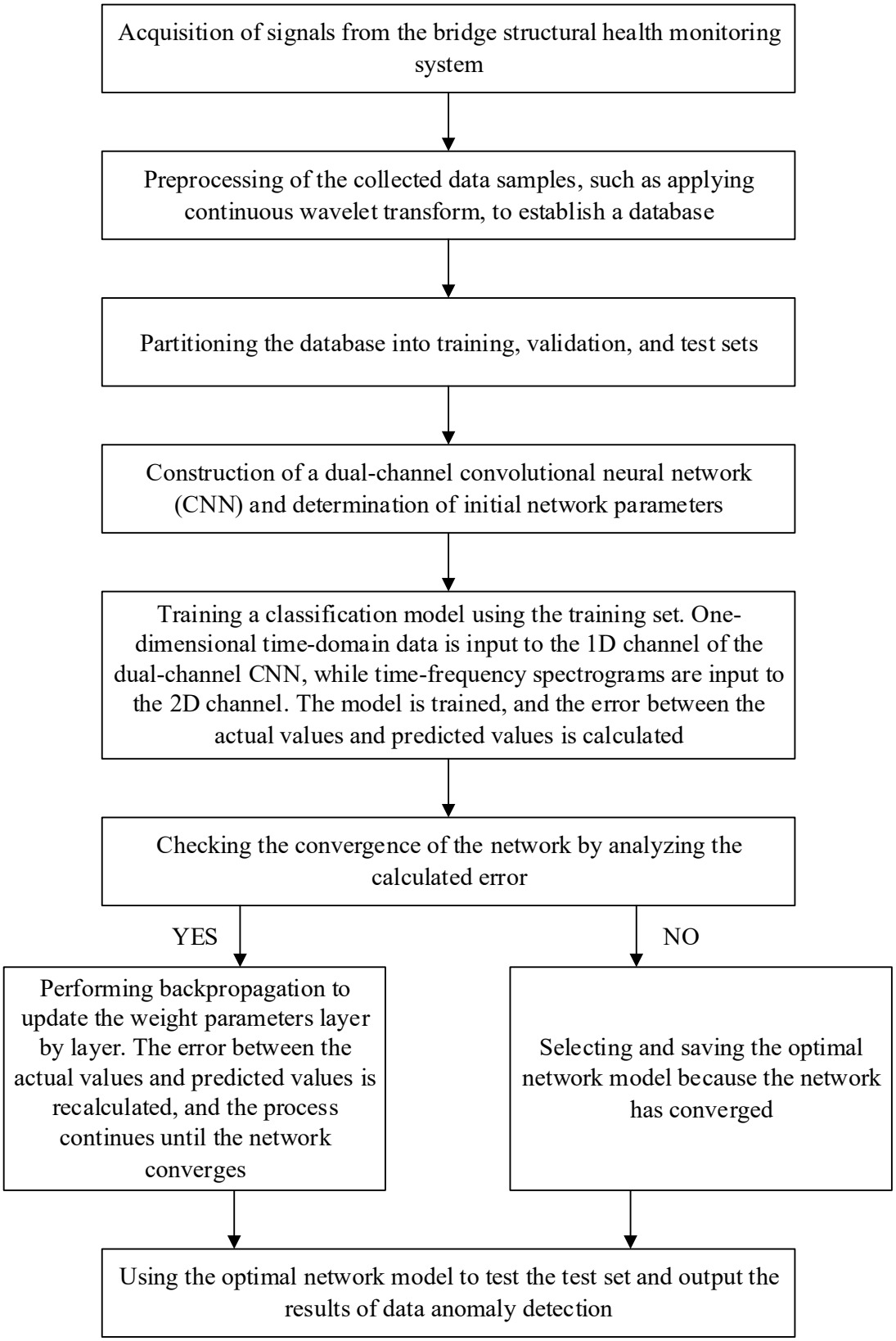


Figure 2. The specific steps of the proposed framework.

2.1. Introduction of Wavelet Transform

The fundamental concept of wavelet transform is based on the observation that high-frequency and low-frequency signals exhibit different time-varying characteristics in a

given signal. Studies have revealed that the frequency spectrum of high-frequency signals changes rapidly over time, whereas the frequency spectrum of low-frequency signals changes slowly over time. Exploiting this property, wavelet transform utilizes scaling and translation factors to enable efficient multi-resolution analysis of signals.

2.1.1. Continuous Wavelet Transform

Define $x(t)$ as a finite-energy function and $x(t) \in L^2(R)$. The continuous wavelet transforms of $x(t)$ are defined as the inner product of the L^2 norm in Hilbert space, as expressed in Equation (1)

$$W_x(a,b;\psi) = \langle x(t), \psi_{a,b}(t) \rangle = a^{-1/2} \int_{-\infty}^{+\infty} x(t)\psi^*_{a,b}(t)\mathrm{d}t \ (a > 0) \tag{1}$$

where $*$ stands for the complex conjugate, and $\psi_{a,b}(t)$ is expressed as:

$$\psi_{a,b}(t) = a^{-1/2}\psi\left(\frac{t-b}{a}\right) \tag{2}$$

$\psi()$ is a continuous function in both the time domain and the frequency domain, called the mother wavelet; a is a parameter characterizing frequency; b is a parameter characterizing time or spatial location. The time–frequency resolution of wavelet analysis is influenced by the frequency content of the signal. In the high-frequency range, wavelets provide high time resolution but low frequency resolution. Conversely, in the low-frequency range, wavelets offer high frequency resolution but low time resolution. This behavior arises from the nature of wavelet functions, which exhibit different oscillatory properties at different scales. By adapting the scale of the wavelet, one can achieve either fine time resolution or fine frequency resolution, allowing for localized analysis of signals in different frequency bands.

2.1.2. Wavelet Scalogram

The time–frequency mapping of the CWT is generally represented by the wavelet scalogram, which can be described as follows:

$$Sc_x(a,b) = |W_w(a,b;\psi)|^2 = \frac{1}{|a|}\left|\int_{-\infty}^{\infty} x(t)\psi_{a,b}(t)dt\right|^2 \tag{3}$$

the visualized result represents the square of the absolute value of the wavelet coefficient. This paper employs Morlet wavelet [33] basis functions to perform signal transformation and generate the corresponding wavelet scalogram. This Morlet wavelet is a single-frequency complex harmonic function with Gaussian amplitude, which can be expressed as:

$$y(t) = e^{iw_0 t}e^{\frac{-t^2}{2}} \tag{4}$$

where ω_0 refers to the frequency of a complex harmonic function. Details of this wavelet can be found in [34].

2.2. Introduction of CNN

CNN is a type of deep learning model specifically designed for processing and analyzing visual data such as images. It consists of multiple layers, including convolutional layers that extract meaningful features from input data, pooling layers that downsample the feature maps to reduce dimensionality, and fully connected layers that perform classification based on the extracted features. Some specific characteristics of CNN are introduced in the following sections.

2.2.1. Convolution and Rectified Linear Unit

The objective of convolution is to capture distinctive features from the input data. In a CNN, each convolutional layer consists of multiple convolutional neurons, and the parameters of each convolutional unit are optimized through the backpropagation algorithm. Within a convolutional layer, the input layer undergoes convolution with trainable filters, resulting in intermediate feature maps that are obtained as shown in Equation (5).

$$x_j^i = f(\sum_{i=1}^{I} x_i^{l-1} \cdot F_{ij}^l + b_j^l), j \in [1, J] \tag{5}$$

where x_j^i is the jth neuron of layer l; F_{ij}^l is the ith neuron of filter j in layer l; and I and J are the neuron amounts of layer $l-1$ and l, respectively. $f(\cdot)$ denotes the activation function. Convolution is performed by applying a filter to the input data. The filter conducts element-wise multiplication using the receptive field of the input, sums the products in each channel, and moves with a designated stride. This process is repeated to cover the entire input. The filter is represented by a matrix, and different filters can extract different features. The number of filters determines the number of output channels, as each filter produces a separate channel in the output feature maps. The input plane size of the output feature maps is calculated based on the size of the input, the receptive field size, and the stride length. The input plane size of the output feature maps is calculated by

$$D_{out} = \frac{D_{in} - C}{S_{conv}} + 1 \tag{6}$$

where D_{in} is the side length of input layer $l-1$; C is the side length of the filter; D_{out} is the side length of output layer l; and S_{conv} is the stride, both in height and width.

Figure 3 is an example of convolution between image and filter. As mentioned above, convolution is a linear operation. In order to introduce nonlinearity and enhance the model's learning capability for complex nonlinear behaviors, it is necessary to apply an activation function to the feature maps generated by the convolutional layers to obtain non-linearized feature maps. Commonly used activation functions include the logistic function and the hyperbolic tangent function. In this study, we utilize the rectified linear unit (ReLU) as the activation function, which effectively addresses the vanishing gradient problem. The mathematical expression of ReLU is as follows:

$$\mathrm{ReLU}(x) = \begin{cases} 0, x < 0 \\ x, x > 0 \end{cases} \tag{7}$$

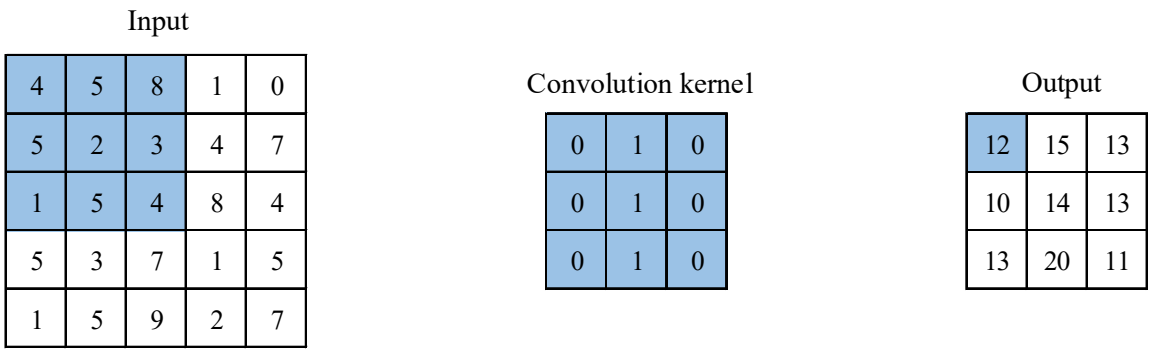


Figure 3. Example of convolution between image and filter.

2.2.2. Pooling

Pooling is a technique used to decrease the dimensionality of features. Its primary purpose is to reduce computational complexity by reducing network parameters and to prevent overfitting, to some extent. Figure 4 is a pooling sample. During pooling, feature

maps are independently subsampled within each channel by selecting representative values from receptive fields, such as:

$$x_j^l = x_i^{l-1} * P_i^l, i \in [1, I] \tag{8}$$

where x_j^l is the is the jth neuron of layer l; P_i^l is the ith neuron of the pooling operator in layer l; and I is the neuron amount of layer. Moreover, the size of the output feature maps in the spatial dimension is determined in Equation (9).

$$D_{out} = \frac{D_{in} - P_{pool}}{S_{pool}} + 1 \tag{9}$$

where P_{pool} is the side length of the pooling operator and S_{pool} is the stride, both in height and width.

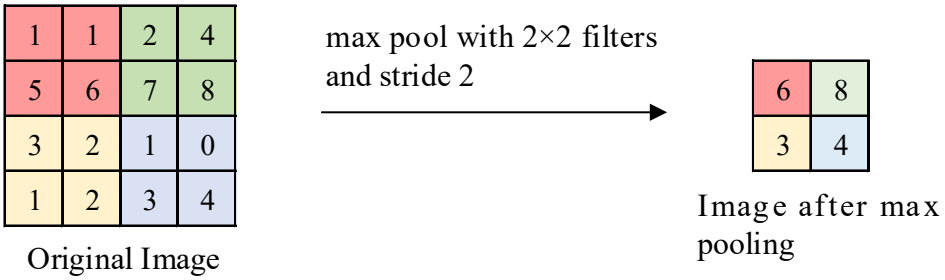


Figure 4. Pooling sample.

2.2.3. Classification

After multiple rounds of convolution, pooling, feature extraction, and dimension reduction, the input image or data is processed to obtain feature maps, which are then fed into a fully connected layer. As the name implies, each neuron in the fully connected layer is connected to every neuron in the previous layer. The purpose of the fully connected layer is to recombine the local features obtained from the convolutional layers, activation functions, and pooling layers into a complete image or data representation, using a weighted matrix. In the fully connected layer, the multidimensional neurons from the previous layer are reshaped into a one-dimensional form. The mathematical formula for this transformation is as follows:

$$x_j^l = f(\sum_{i_1=1}^{Height} \sum_{i_2=1}^{Width} \sum_{i_3=1}^{Channel} x_{i_1 i_2 i_3}^{l-1} W_{i_1 i_2 i_3}^l + b_j^l), j \in [1, J] \tag{10}$$

where x_j^l represents the output of the j-th neuron in the l-th fully connected layer. J denotes the number of neurons in the layer. $x_{i_1 i_2 i_3}^{l-1}$ represents the output of the multidimensional neurons in the previous layer, where i_1, i_2, i_3 represent the length, width, and height of the feature map, respectively. $W_{i_1 i_2 i_3}^l$ represents the weights between the two layers of neurons. b_j^l represents the bias vector, and $f(\cdot)$ represents the activation function.

The layer in a convolutional neural network that is responsible for image classification is referred to as the output layer, and is also a fully connected layer. This layer typically serves as a classifier function, producing an output vector where each value corresponds to the probability of a particular class. The classifier function used in this study is the Softmax function. When the output from the previous fully connected layer is fed into the Softmax classifier function, denoted as the l-th layer, its computation can be expressed as displayed in Equation (11).

$$y_a = [P(prediction = a | x^{l-1}; W_t^l)] = \frac{e^{W_A^l x^{l-1}}}{\sum_{m-1}^{A} e^{W_m^l x^{l-1}}}), a \in [1, A] \tag{11}$$

where y_a represents the probability of class a, and A denotes the total number of classes. x^{l-1}, W_a^l and W^l are shown in Equations (12)–(14):

$$x^{l-1} = [x_1^{l-1}, \cdots, x_i^{l-1}]^T \tag{12}$$

$$W_a^l = [W_{a1}{}^l, \cdots, W_{ai}{}^l] \tag{13}$$

$$W^l = [W_1{}^l, \cdots, W_A^l]^T \tag{14}$$

where x^{l-1} represents the feature vector of the neural network outputs from the $l-1$ layer with a size of $I \times 1$, and W_a^l corresponds to the weight connecting the feature vector from the $l-1$ layer to class a, with a size of $1 \times i$. Figure 5 illustrates a simplified diagram of the computation between the Softmax function and the neural network outputs from the second-last to the last layer.

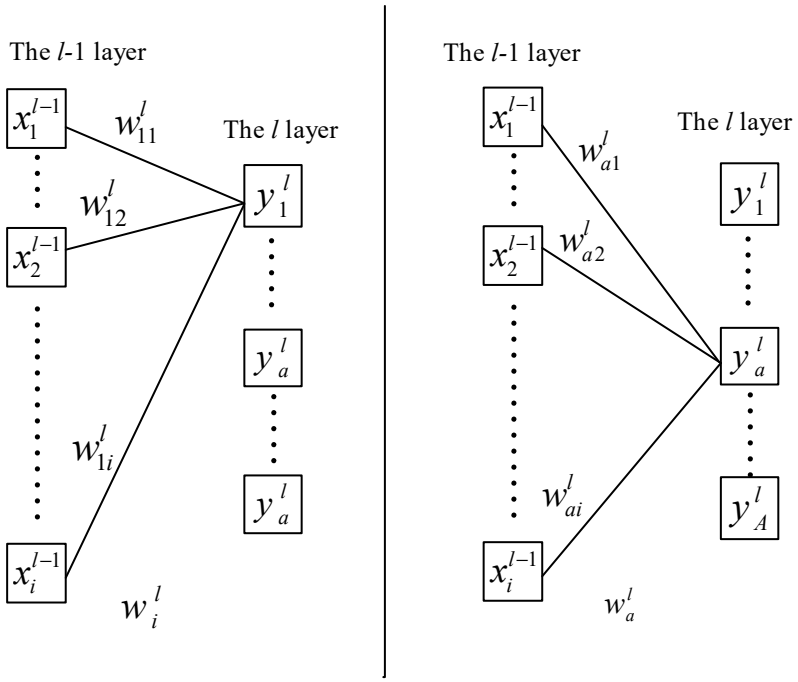


Figure 5. Example of the SoftMax layer.

In the image classification task of deep learning models, the Softmax classifier function is used to obtain the predicted results of the input data. The next step involves comparing the predicted results with the ground truth and adjusting and optimizing the model accordingly. In this process, to quantify the difference between the predicted results and the ground truth, a loss function is introduced to optimize the deep learning network. This study employs the commonly used loss function for multi-class classification in deep learning models, which is the cross-entropy loss. Its mathematical formula is as follows:

$$\begin{aligned} \mathrm{E(W)} &= -\frac{1}{N}[\sum_{n=1}^{N}\sum_{a=1}^{A} 1\left\{y^N = a\right\} \log(y_a^N)] \\ &= -\frac{1}{N}\left[\sum_{n=1}^{N}\sum_{a=1}^{A} 1\{y^N = a\} \log\left(\frac{e^{[W_a^l x^{l-1}]^N}}{\sum_{m=1}^{A} e^{[W_m^l x^{l-1}]^N}}\right)\right] \end{aligned} \tag{15}$$

where $\mathrm{E(W)}$ represents the loss function, N is the total number of samples; $1\{\cdot\}$ is the indicator function, which takes the value of 1 when the prediction is true and 0 when

the prediction is false. y_a^N represents the predicted probability of the model for the a-th class and the N-th sample, and y^N represents the true label of sample N. Details of the loss function can be found in [35].

2.3. Parameter of the Proposed Framework

The base CNN in this paper is LeNet-5 [36], and the specific parameters of the LeNet-5 network structure are presented in Table 1. It consists of an input layer, two convolutional layers, two pooling layers, two fully connected layers, and an output layer.

Table 1. Parameter of LeNet-5.

Layer	Kernel/Neuron Number	Step	Output Size
Input	-	-	32×32
Convolution	5×5, 6	1	$28 \times 28 \times 6$
Pooling	2×2, 6	2	$14 \times 14 \times 6$
Convolution	5×5, 16	1	$10 \times 10 \times 16$
Pooling	2×2, 16	2	$5 \times 5 \times 16$
Fully connected	-, 120	-	-
Fully connected	-, 84	-	-
Softmax (Output)	-, 10	-	-

Due to the simplicity of the LeNet-5 network structure and its small number of parameters, it is well suited for the task of recognizing handwritten digits. The input images used for this task are grayscale images of size 32×32. However, in the context of structural health monitoring (SHM) data anomaly detection, the original data has a dimension of $1 \times 72{,}000$, representing the acceleration data. To effectively utilize the SHM data and achieve accurate anomaly detection, this paper introduces an additional convolutional channel that takes the raw one-dimensional vibration time-series data as input. Therefore, the proposed dual-channel CNN architecture for SHM data anomaly detection is illustrated in Figure 6.

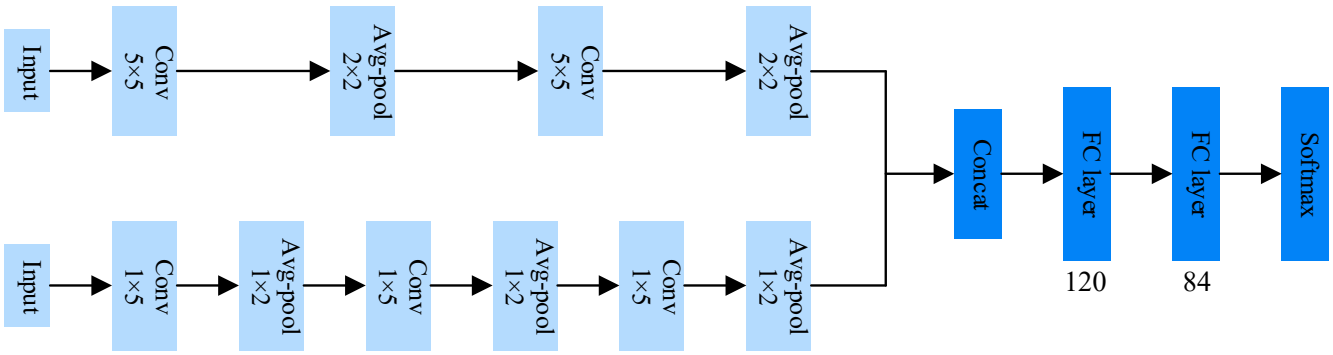


Figure 6. Dual-channel convolutional neural network mode.

The main feature of the proposed network architecture is the presence of two convolutional channels: a 2-dimensional convolutional neural network (2D-CNN) channel for processing time–frequency images and a 1-dimensional convolutional neural network (1D-CNN) channel for processing raw one-dimensional vibration time-series data. After convolution and pooling operations, the two types of data are flattened into 1-dimensional vectors and concatenated at the concatenation layer. The concatenated vector is then fed into fully connected layers for final SHM data anomaly classification. The specific parameters of this dual-channel convolutional neural network are detailed in Table 2.

Table 2. Key parameters in the proposed framework.

CNN Branch	**1D Branch**			**2D Branch**		
Input format	Time-series data (1 × 1024)			wavelet scalogram (120 × 160)		
Feature extraction	Layer	Kernel	Shape	Layer	Kernel	Shape
	CONV	6@1 × 5	1 × 71,996 × 6	CONV	6@5 × 5	220 × 220 × 6
	POOLING	6@1 × 2	1 × 35,998 × 6	POOLING	6@2 × 2	110 × 110 × 6
	CONV	16@1 × 5	1 × 35,994 × 16	CONV	16@5 × 5	106 × 106 × 16
	POOLING	16@1 × 2	1 × 17,997 × 16	POOLING	16@2 × 2	53 × 53 × 16
	CONV	16@1 × 5	1 × 17,792 × 16	/	/	/
	POOLING	16@1 × 2	1 × 8996 × 16	/	/	/
Feature fusion	Flatten/Connect	143,936 + 44,944 neurons				
Feature classification	Full connection1	120 neurons				
	Full connection2	84 neurons				
	Softmax	7 neurons				

3. Database Introduction and Data Preprocessing

3.1. Model-Data Introduction

3.1.1. Experiment Bridge and the SHM System

The data used for the experiments are obtained from the structural health monitoring system of a large-span cable-stayed bridge. The database is provided by the 1-st International Project Competition for Structural Health Monitoring (IPC-SHM-2020) [37]. The cable-stayed bridge has a main span length of 1088 m, side span length of 300 m, and tower height of 306 m. It was completed in 2008, and has been equipped with a structural monitoring system since then. The health monitoring system of the bridge includes accelerometers, anemometers, strain gauges, global positioning systems, thermometers, etc. Table 3 displays the used sensor with corresponding monitored parameters.

Table 3. Sensor selection.

Monitored Parameter	**Sensor Type**	**Indicator**
Wind	Wind speed and wind direction meter	Specify the wind speed in the direction of the wind (average wind speed of 3 s, 2 min, and 10 min)
Humidity	Air temperature and humidity meter	Ambient temperature and humidity
Temperature	Thermometer	Temperature and temperature gradient
Displacement	Level meter	Beam end displacement, bridge lateral offset
Stress and strain	Vibrating string strain gauges	Stress–strain distribution of the main beam and main tower
Cable tension	Accelerometer	Cable frequency and cable force
Seat movement	Displacement sensors	Seat displacement
Main tower displacement	GPS	Structural general offset
Traffic flow monitoring	Camera	Traffic flow
Vibration of the main beam	Accelerometer	Amplitude

These sensors are the front-end devices in the SHM system, and all data collected are processed by corresponding hardware/software. This data processing procedure is shown in Figure 7, which is composed of four sub-systems.

(1) The sensor subsystem

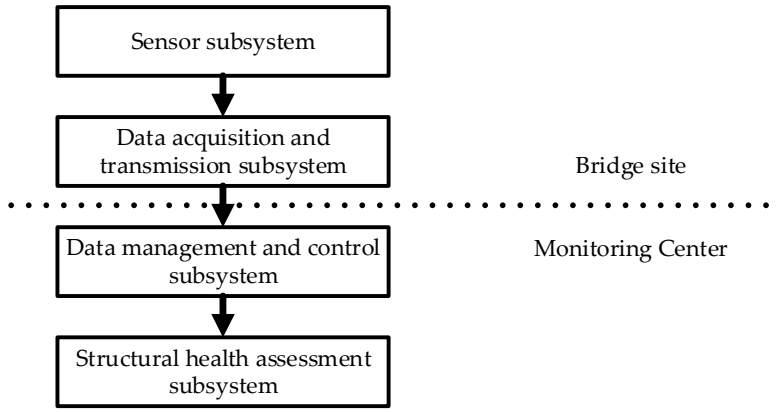


Figure 7. The implemented SHM system.

The sensor subsystem serves as the forefront of the monitoring system, where various metrics of the bridge are collected with the involvement of sensors. The key aspects of the sensor subsystem are the rational placement of monitoring points and the selection of appropriate sensors, enabling optimal monitoring of the bridge within a cost-effective framework.

(2) The data acquisition and transmission subsystem

The data acquisition and transmission subsystem primarily involves converting, pre-processing, storing, and transmitting real-time data from the bridge to the data center using signal conditioners and optical fiber cables. The subsystem comprises data collection units (workstations), data transmission networks, and data acquisition and transmission software.

(3) The data management and control subsystem

The data management and control subsystem consist of three main functional modules: human–machine interface module, control and analysis module, and system management module. Firstly, the human–machine interface module provides real-time variations of bridge data, displaying curves related to bridge environment, deformation, internal forces, cable forces, etc. It also offers an interface to set various collection and transmission parameters and presents alerts for faults, anomalies, data exceedances, and abnormal data. Secondly, the control-and-analysis module allows real-time queries of collected data and analysis through large computational software like Matlab, Ansys, etc. Various analyses include temperature record calculation, vibration force calculations, variance calculations, etc. The module can also identify sensor failures and data anomalies, issuing the corresponding alarms. Furthermore, this module facilitates centralized data management and importation, performing statistical calculations (extremum, average, trend lines, etc.) on processed data, and storing the results in a database. Maintenance units can access these data from the database for real-time viewing. Finally, the system management module controls data collection and transmission through communication protocols. Database backup and archiving are customarily performed, and data storage involves compression during backup and storage in high-capacity storage devices.

(4) The structural health assessment subsystem

The structural health assessment subsystem accomplishes bridge pre-warning and evaluation through data collection. Its evaluation module consists of internal force state recognition, structural dynamic characteristic analysis, vibration characteristic analysis, wind field analysis, wind-induced vibration analysis, and other modules. Based on the results, regular evaluations of the bridge's structural performance are conducted, generating quantitative or qualitative assessment reports (monthly, quarterly, yearly, or for special events), guiding and providing a basis for bridge maintenance, repair, and management

decisions. The pre-warning module is responsible for real-time structural monitoring concerning variable loads (e.g., wind loads, heavy vehicles) and structural response indicators (e.g., main beam deformations) that may pose a threat to the bridge's structural safety. Additionally, in special weather conditions or abnormal operating situations, the pre-warning system triggers alerts to prompt the bridge management and maintenance department to pay attention to the operational safety of the structure.

The four subsystems operate independently while collaborating harmoniously, establishing the efficient structural health monitoring system for the structure. Additionally, the method proposed in this study if for identifying abnormal data that can be seamlessly integrated into the fourth module, prior to conducting structural assessment and pre-warning, which ensures the authenticity of the data sources.

3.1.2. Data Process System

A total of 38 accelerometers are installed on the deck and tower, and their specific locations are shown in Figure 8.

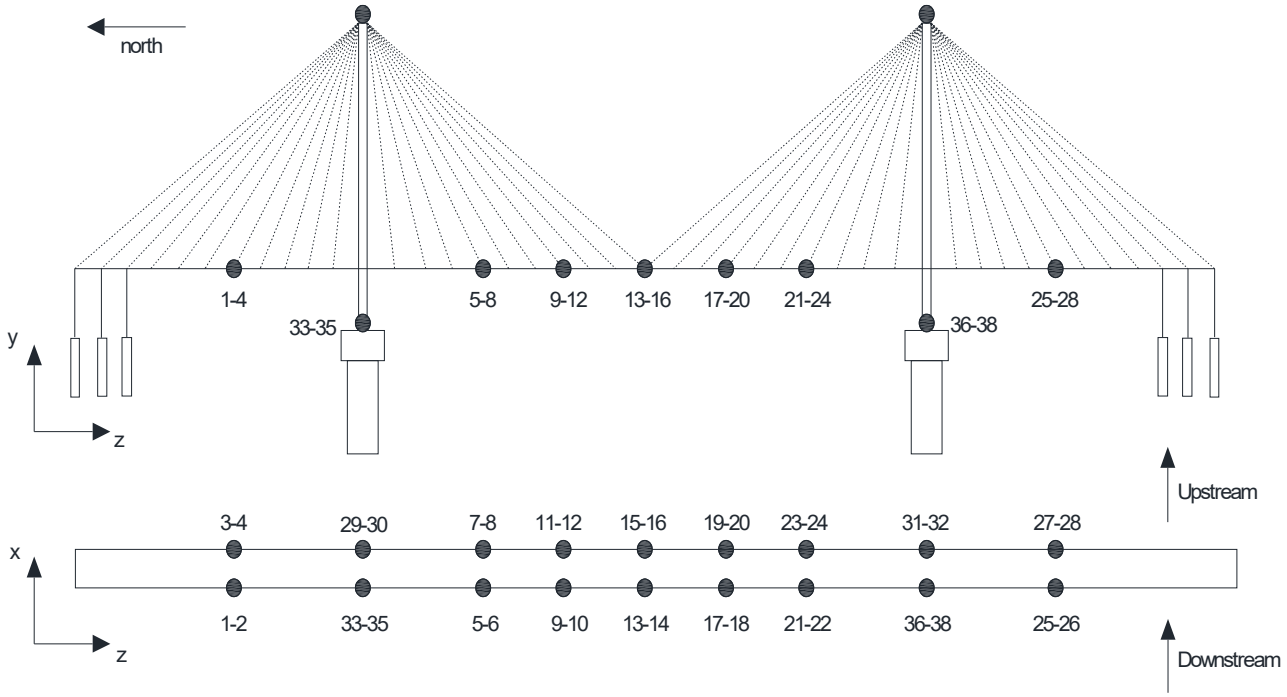


Figure 8. Sensor locations (The numbers in the figure represent the sensor number).

The database used in this study consists of accelerometer information collected from the aforementioned 38 sensors, with a sampling frequency of 20 Hz. The normally monitored data with six types of anomaly data, including missing data, minor data, outliers, over-range oscillation, trend data, and drift data, are shown in Figure 9. The distribution of the different data anomaly patterns is displayed in Table 4.

Table 4. Distribution of different data anomaly patterns.

Label	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Pattern	Normal	Missing	Minor	Outlier	Square	Trend	Drift
Quantity	13,575	2942	1775	527	527	5778	679
Proportion	52.60%	11.40%	6.90%	2.00%	2.00%	22.40%	2.60%

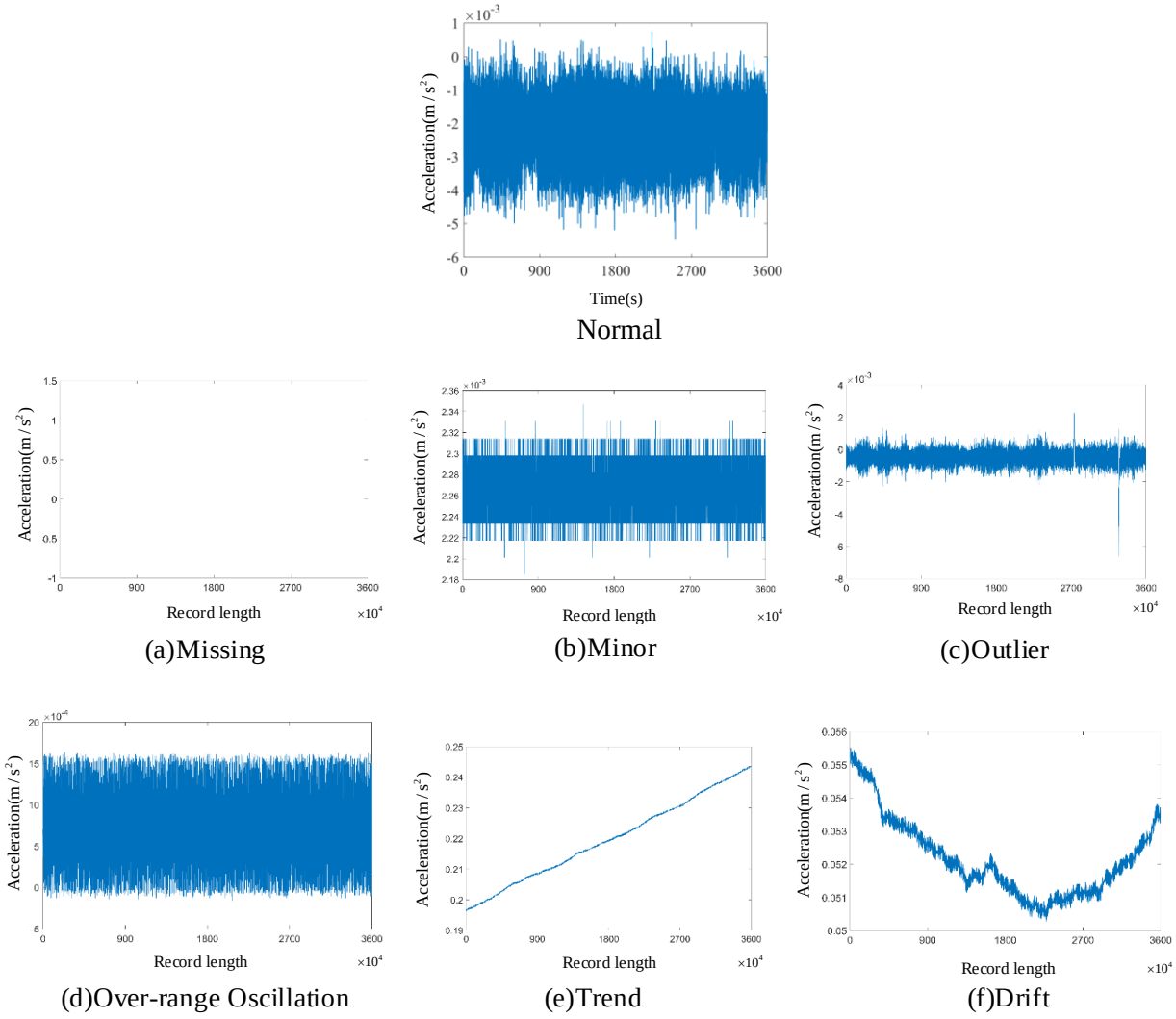


Figure 9. Example of different data patterns.

3.1.3. Data Anomaly Patterns

(1) Missing

"Missing" is a common type of data anomaly in structural health monitoring systems. It is characterized by a continuous period of time where the monitored data collected by a sensor are either blank or a constant value. The main cause of this data anomaly is external interference that the sensor encounters during its operation, such as sudden power supply interruption or system reboot. The mathematical expression for missing data is given as follows:

$$x_{out}(t) = C \tag{16}$$

where $x_{out}(t)$ represents the output of the sensor at time t, and C is a constant value.

(2) Minor

Minor, also referred to as "precision degradation", is a prevalent data anomaly in structural health monitoring systems. It is primarily caused by partial sensor malfunction or damage, resulting in significantly smaller amplitudes compared to normal data. When minor values persist over a continuous time interval, the visual representation of the original data exhibits a distinct sawtooth pattern. The mathematical expression for this anomaly is given by Equation (17)

$$x_{out}(t) = \alpha \cdot x(t) + \delta(t) \tag{17}$$

where α represents the fault coefficient (less than 1), and $\delta(t)$ denotes the noise component.

(3) Outlier

Outliers, in general, refer to abrupt or extreme values observed in the monitoring data at a specific moment. The occurrence of outliers can be attributed to factors such as sudden voltage surges in sensors or random electromagnetic interference. Although outliers are typically sporadically distributed among normal data, their presence can lead to significant errors in data preprocessing of the structural health monitoring system, such as inaccurate data normalization, incorrect data feature aggregation, and so on.

(4) Over-range oscillation

Over-range oscillation is characterized by abnormal oscillations in the amplitude of the vibration response within the range of the accelerometer. The amplitude of over-range oscillation is larger compared to normal data, and exhibits a square wave-like shape. This data anomaly type is usually caused by factors such as unstable sensor voltage. The specific mathematical expression for this data anomaly type is as follows.

$$x_{out}(t) = b \cdot x(t) + \delta(t) \tag{18}$$

where "b" represents the gain coefficient of the over-range oscillation, which is greater than 1.

(5) Trend

When normal data are subjected to long-term disturbances causing deviation, it results in a data anomaly known as trend. The visual representation of the original data exhibits characteristics similar to a linear function, showing a monotonic increasing or decreasing trend. Equation (19) is the corresponding mathematic model

$$x_{out}(t) = x(t) + dt + \delta(t) \tag{19}$$

where d represents the tangent.

(6) Drift

Drift is a data anomaly type that occurs when the monitored data deviate continuously over time, due to various environmental factors or sensor malfunctions. It can manifest as local monotonic increases or decreases, forming a curved line with alternating patterns. It can also be understood as a combination of multiple "trend" anomalies, resulting in a complex deviation pattern. This anomaly pattern could be expressed as follows:

$$x_{out}(t) = x(t) + c + dt + \delta(t) \tag{20}$$

where coefficient c and d are random variables that may change at different t.

3.2. Data Preprocessing

In general, the raw data provided by the original database cannot be directly used as input for the dual-channel convolutional neural network. Therefore, a data preprocessing step is required before training the classification model. In this study, three data preprocessing techniques were employed, namely database augmentation, 2D data conversion, and dataset partition.

(1) Database augmentation

Due to the limited number of samples for the "minor", "over-range oscillation", and "drift" categories in the database, which is less than 1000, it was necessary to expand the sample size for these three types of anomalies to facilitate the data set partitioning in subsequent experiments. In this study, a method was employed to augment the sample size by adding Gaussian white noise to the original data. Two sets of new data were generated for each of the three anomaly types by adding 10% and 15% Gaussian white noise to the original data, respectively. Finally, the expanded database's sample counts and proportions for each anomaly class are shown in Table 5.

Table 5. The used data set.

Label	1	2	3	4	5	6	7
Pattern	Normal	Missing	Minor	Outlier	Over-range oscillation	Trend	Drift
Quantity	13,575	2942	1775	1581	1581	5778	2037
Proportion	46.4%	10.1%	6.1%	5.5%	5.5%	19.4%	7.0%

(2) Two-dimensional data conversion

In order to train the classification model using the dual-channel convolutional neural network for data anomaly detection in this study, one of the channels requires the input data to be in the form of 2D images (as for the remaining channel, the required 1D vector can be directly obtained from the sequential monitored data). Therefore, prior to training the model, the data need to be visualized and transformed into training images. In this study, continuous wavelet transform (CWT) was selected as the method for time–frequency analysis of the original data. This method allows the conversion of time-series data into time–frequency images while preserving the temporal and spectral characteristics of the original data (details of wavelet transform theory can be found in Section 2.1.2).

The specific implementation process involves the following three steps: determining the length and sampling frequency of the input data, computing the wavelet transform coefficients for each data segment using continuous wavelet transform, and plotting the wavelet transform coefficients as time–frequency images of suitable sizes, and saving them. Following these steps, each data sample was visualized, resulting in a total of 29,269 wavelet time–frequency images. Partial results are shown in Figure 10.

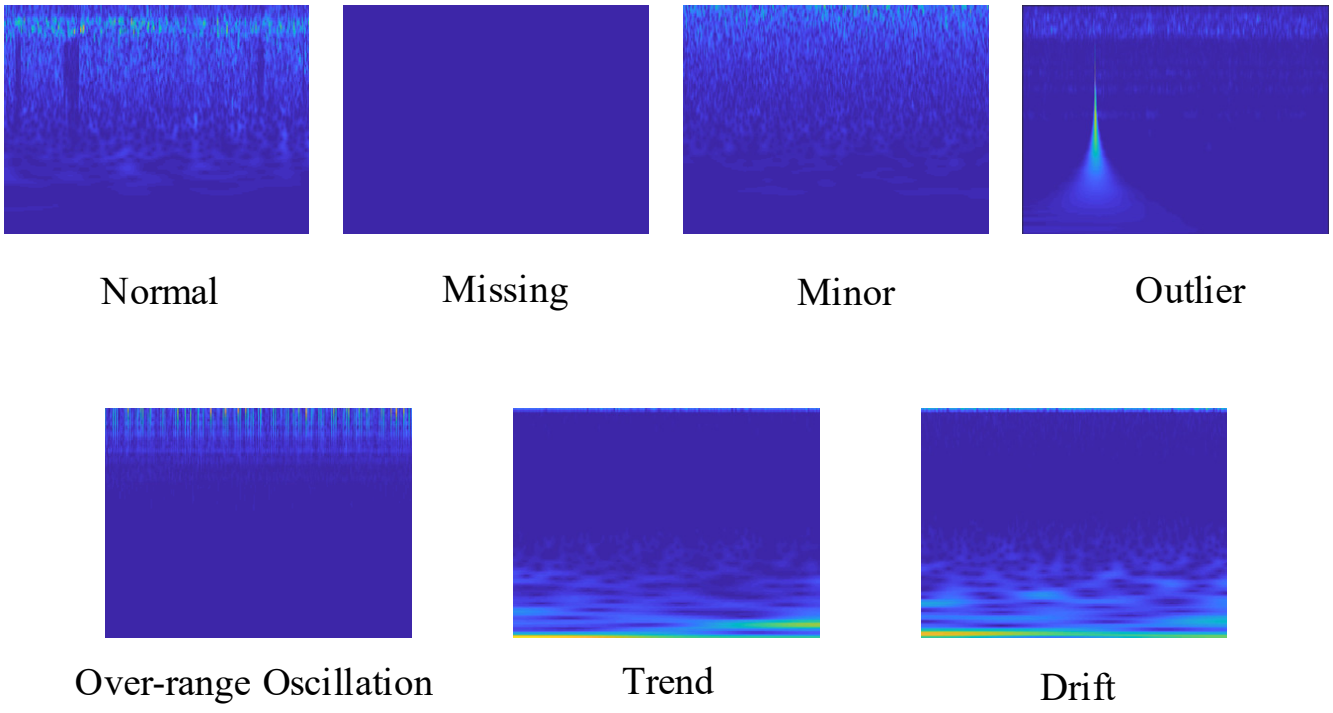


Figure 10. Wavelet time-frequency images of different anomaly pattern.

(3) Dataset partition

The database consists of 29,269 data samples, with varying proportions of different data types within the database. During the development of deep learning, it has been pointed out in [38] that an imbalance in the sample quantities of different data categories used for training a deep learning classification model can affect the model's prediction performance. Therefore, in order to gain a deeper understanding of the performance of

the model trained by the dual-channel convolutional neural network, this study designed the experimental dataset proportions based on [19]. The dataset is mainly divided into balanced and imbalanced conditions, with three data ratios for each condition: 12%, 24%, and 36%. In total, there are six cases, with Cases 1, 3, and 5 being balanced conditions, and Cases 2, 4, and 6 being imbalanced conditions. Half of the randomly sampled data according to the aforementioned ratios from the database are used as the training set, another half as the validation set, while the remaining samples (with the ratio of 88%, 76%, and 64%) in the database are used as the test set. Details of the training, validation and testing sets are shown in Tables 6 and 7.

Table 6. The number of data in the training and validation sets.

Anomaly Pattern	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Normal	251	814	502	1629	753	2443
Missing	251	176	502	353	753	529
Minor	251	106	502	213	753	319
Outlier	251	94	502	189	753	284
Over-range oscillation	251	94	502	189	753	284
Trend	251	346	502	693	753	1040
Drift	251	122	502	244	753	366

Table 7. The number of data in the training and testing set.

Pattern	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Normal	13,073	11,946	12,571	10,317	12,069	8688
Missing	2440	2589	1938	2236	1436	1883
Minor	1273	1562	771	1349	269	1136
Outlier	1079	1392	577	1202	75	1012
Over-range oscillation	1079	1392	577	1202	75	1012
Trend	5276	5085	4774	4392	4272	3698
Drift	1535	1793	1033	1549	531	1304

4. Model Performance

4.1. Training and Validation Results

The dual-channel convolutional neural network was trained using the six training sets obtained from the previous section. As a result, six models were obtained, along with their corresponding prediction results. The detailed classification results of the models on the training and validation sets are shown in Tables 8–10, while the results on the test set are presented in Table 11.

Table 8. F1-score and accuracy of Case 1 and 2.

Anomaly Pattern	Indicator	Case 1		Case 2	
		Training Set	Validation Set	Training Set	Validation Set
Normal	F1-Score	0.99	0.834	0.998	0.949
Missing		0.99	0.99	0.986	0.986
Minor		0.955	0.876	0.946	0.717
Outlier		0.953	0.857	0.933	0.696
Over-range oscillation		1	0.992	1	0.984
Trend		0.973	0.882	0.984	0.907
Drift		0.972	0.883	0.962	0.782
Summation	Accuracy	97.61%	90.27%	98.52%	90.73%

Table 9. F1-score and accuracy of Case 3 and 4.

Anomaly Pattern	Indicator	Case 3		Case 4	
		Training Set	Validation Set	Training Set	Validation Set
Normal	F1-Score	0.987	0.890	0.990	0.964
Missing		0.995	0.990	0.982	0.978
Minor		0.973	0.910	0.946	0.771
Outlier		0.959	0.892	0.890	0.827
Over-range oscillation		1.000	1.000	1.000	0.995
Trend		0.973	0.937	0.981	0.959
Drift		0.976	0.944	0.950	0.900
Summation	Accuracy	98.04%	93.68%	97.61%	94.25%

Table 10. F1-score and accuracy of Case 5 and 6.

Anomaly Pattern	Indicator	Case 5		Case 6	
		Training Set	Validation Set	Training Set	Validation Set
Normal	F1-Score	0.969	0.898	0.994	0.975
Missing		0.989	0.983	0.977	0.965
Minor		0.943	0.901	0.934	0.862
Outlier		0.922	0.888	0.909	0.755
Over-range oscillation		1.000	1.000	0.993	0.995
Trend		0.964	0.927	0.974	0.968
Drift		0.966	0.936	0.941	0.914
Summation	Accuracy	96.45%	93.36%	97.64%	95.18%

Table 11. The F1-score of different anomaly patterns in the testing set.

Pattern	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Normal	0.885	0.947	0.939	0.963	0.939	0.976
Missing	0.983	0.981	0.987	0.980	0.990	0.971
Minor	0.642	0.666	0.715	0.753	0.477	0.862
Outlier	0.552	0.711	0.496	0.760	0.135	0.774
Over-range oscillation	0.975	0.991	0.989	0.993	0.956	0.994
Trend	0.940	0.916	0.941	0.948	0.932	0.961
Drift	0.818	0.778	0.977	0.870	0.668	0.900

From Table 8, it can be observed that Case 2 performs slightly better than Case 1, both in terms of the training and testing sets. Additionally, it is worth noting that both Case 1 and Case 2 exhibit signs of overfitting, as the accuracy on the training set is higher than the accuracy on the validation set by 7.34% and 7.79%, respectively.

Furthermore, although the model may have slightly lower performance in classifying certain categories, it can be seen that the model trained on Case 2 performs better for the "normal" and "trend" data types. On the other hand, the model trained on Case 1 performs better for other data types such as "missing", "minor", and "outlier". Considering the larger number of samples in the training set, it can be concluded that in the validation set, regardless of the data set balance, a training set with a larger number of samples leads to better performance in identifying that particular data anomaly.

Since both Case 1 and Case 2 used relatively small training datasets, we also considered the performance of the model when trained on a larger dataset, as shown in Tables 9 and 10.

Table 9 presents the training results for the medium-sized datasets, Case 3 and Case 4, including the results for the training and validation sets. Compared to the training on small datasets (Case 1 and Case 2), the overfitting phenomenon is alleviated in Case 3 and Case 4. In terms of accuracy, the training sets of Case 3 and Case 4 have higher accuracy than the validation sets by 4.36% and 3.36%, respectively.

As for the F1-score, it can be observed that for the "normal" and "trend", the model trained on the training set of Case 4 performs better. On the other hand, for other data types, the model trained on the training set of Case 3 shows better performance. This observation further validates the conclusion stated earlier: in the validation set, regardless of the data set balance, a training set with a larger number of samples leads to a better ability of the trained model to recognize that particular data anomaly.

Table 10 presents the training results for the large dataset. Compared to the small datasets of Case 1 and 2, and the medium datasets of Case 3 and 4, the overfitting phenomenon continues to be alleviated for Cases 5 and 6. In terms of accuracy, the training sets of Case 5 and 6 outperform the validation sets by 3.09% and 2.46%, respectively.

From the perspective of the F1-score, in the large dataset, the model trained on the training set of Case 6 performs better for the "normal" and "trend" data types. On the other hand, for other data types, the model trained on the training set of Case 5 shows better performance.

4.2. Testing Results

4.2.1. Results of F1-Score and Confusion Matrixes

After training the model on different cases, these models were used to identify data anomalies using testing sets. The test sets were not used in the training or validation process. This allowed for the evaluation of how the various network models performed when presented with a large number of unknown data. Table 11 and Figure 11 present the F1-scores of each model on the test set, indicating their performance across different types of data anomalies.

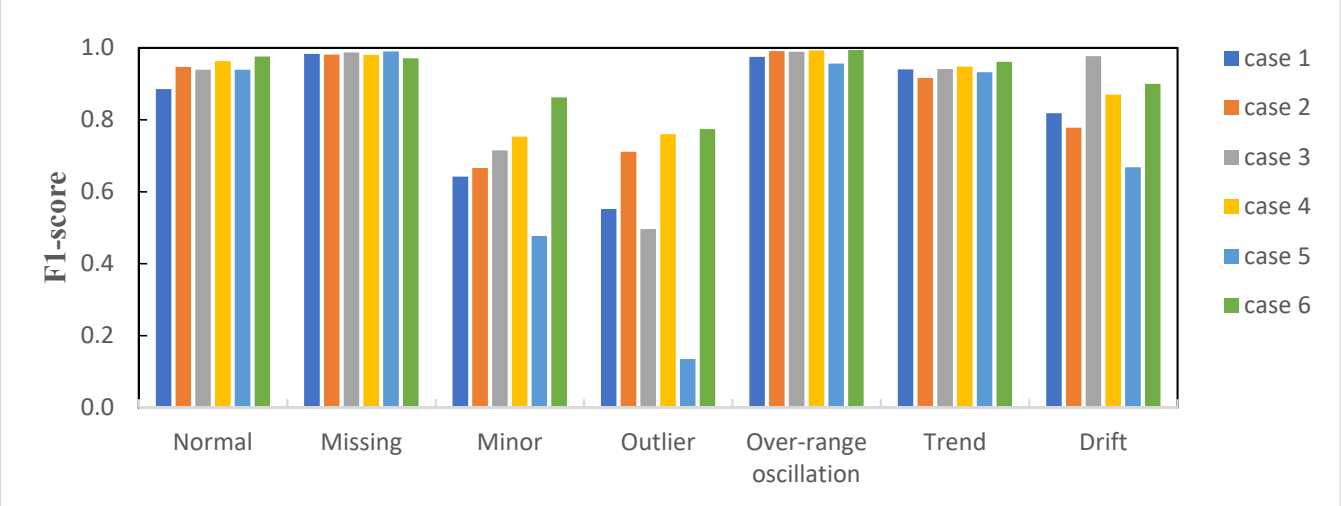


Figure 11. F1-score of different anomaly patterns.

Based on Table 11 and Figure 11, it can be observed that models trained on imbalanced training sets outperformed those trained on balanced training sets for data types such as "normal", "minor", "outlier", "over-range oscillation", and "trend". Among the models, the one trained on Case 6 showed the best classification performance, achieving F1-scores of 0.976, 0.862, 0.774, 0.994, and 0.961 for the respective data types in the large dataset.

Furthermore, for the "normal", minor" and "outlier", larger datasets resulted in better classification performance within the same data type (i.e., balanced or imbalanced). In the case of identifying "missing", even the F1-score of Case 6, the lowest, also achieved a very high score (0.971). Regarding the identification of "missing" and "over-range oscillation", all models trained across the six cases demonstrated excellent classification performance, with F1-scores exceeding 0.95. For the "trend" data type, the model trained on Case 3 achieved the highest F1-score, of 0.977.

It is worth noting that in Case 5, for the identification of "minor" and "outlier", the model trained on a balanced dataset showed a significant decrease in F1-scores, reaching only 0.477 and 0.135, respectively. To further analyze the reasons behind it, the confusion matrix of Case 5 is abstracted and displayed, as shown in Figure 11.

Upon observing the highlighted red boxes in Figure 12, it becomes apparent that the "minor" exhibits a recall rate of 86.6% and a precision rate of 31.7%. Specifically, the confusion matrix reveals that lots of "normal" data are incorrectly classified as "minor". Moreover, the test set contains a relatively small number of outlier samples. These two reasons together contribute to the low precision rate and subsequently result in a lower F1-score.

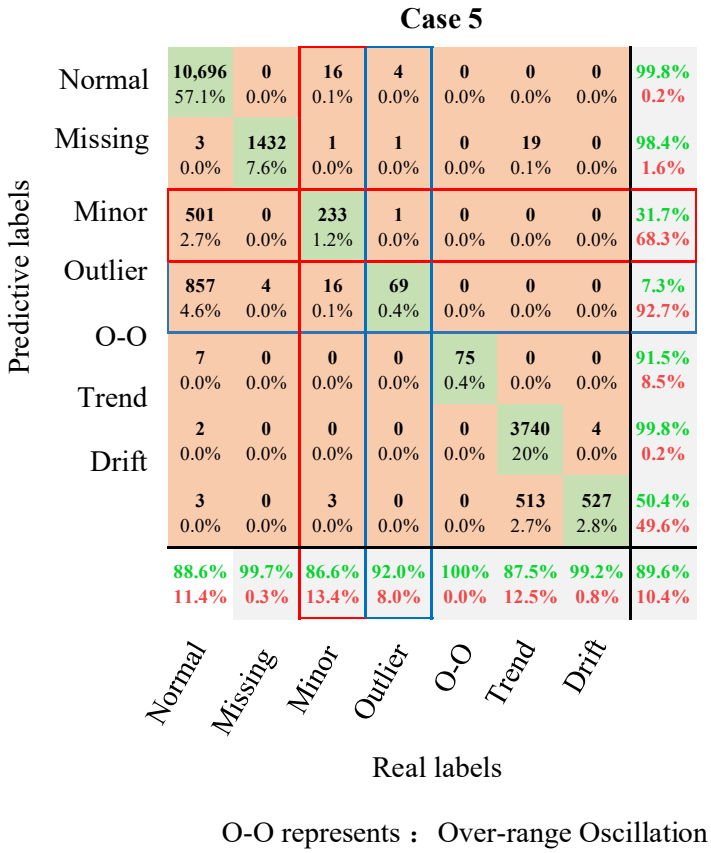


Figure 12. Confusion matrix of Case 5 Testing set.

Similarly, in the blue box, the "outlier" demonstrates a notable recall rate of 92.0%. However, its precision rate is considerably lower, only 7.3%. This is because of a considerable number of "normal" data being misclassified as "outlier". Additionally, the scarcity of outlier samples in the test set contributes to the low precision rate, consequently leading to a lower F1-score.

To gain a better understanding of the overall classification performance of the models, Figures 13–15 display the confusion matrices of the test sets under Cases 1 to 6.

Based on Figures 13–15, the following conclusions can be drawn. The models trained on datasets from Case 1 to Case 6 achieved accuracies of 86.50%, 90.60%, 90.70%, 93.30%, 89.60%, and 95.10%, respectively, when tested on the corresponding datasets. This indicates that as the training dataset expands and the network models are adequately trained, they can more accurately detect data anomalies. Furthermore, comparing models trained on imbalanced datasets to those trained on balanced datasets, it can be observed that models trained on imbalanced datasets outperformed others, achieving the highest accuracy of 95.10%. The high accuracy of 95.10% also indicates the excellent classification performance of the network models.

Additionally, from the six confusion matrices, it can be analyzed that in cases of misclassification, the "normal" is sometime predicted as "minor " and "outlier", the "minor" is prone to be predicted as "normal" and "outlier", the "outlier" data type may be predicted as "normal" and "minor ", the "trend" is sometime mispredicted as "drift ", and the "drift" may be predicted as "trend".

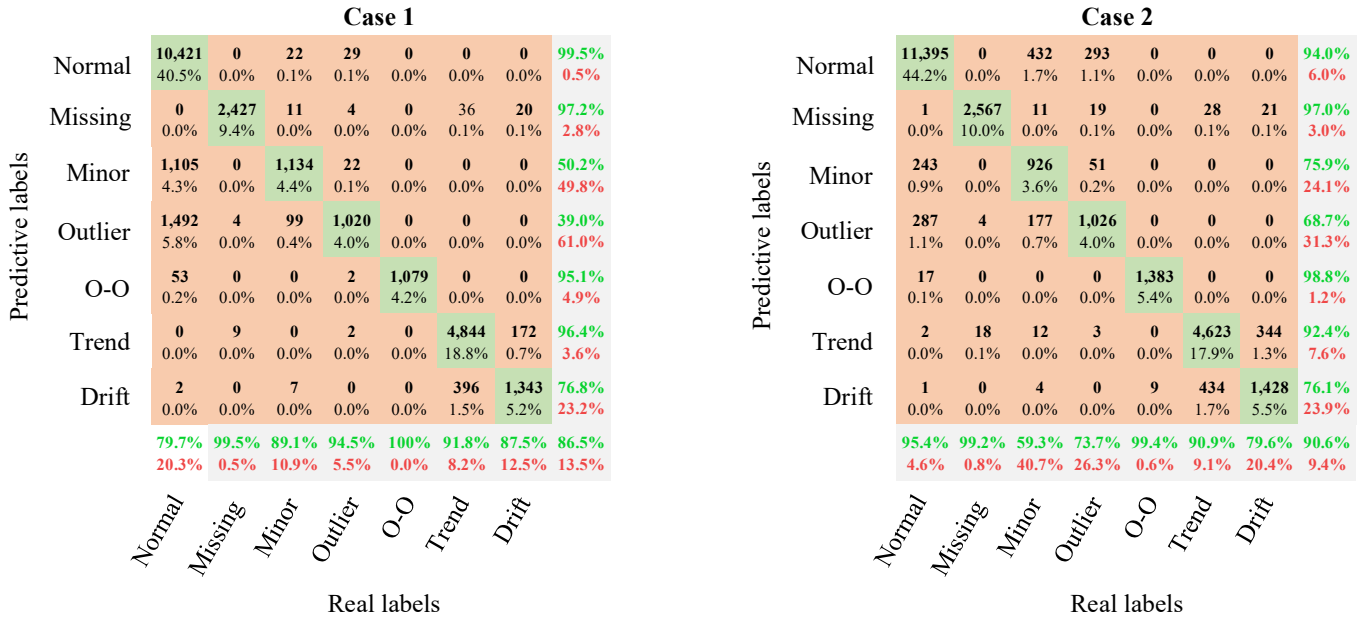


Figure 13. Confusion matrix of Cases 1–2 under testing set.

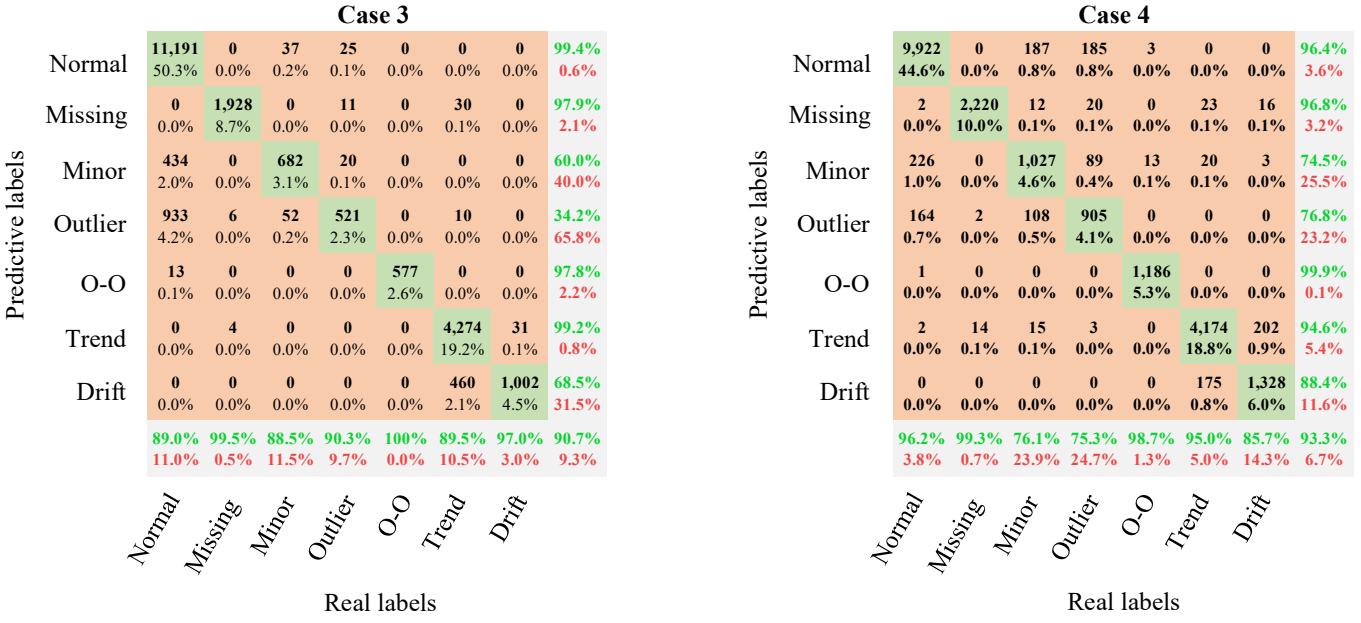


Figure 14. Confusion matrix of Cases 3–4 under testing set.

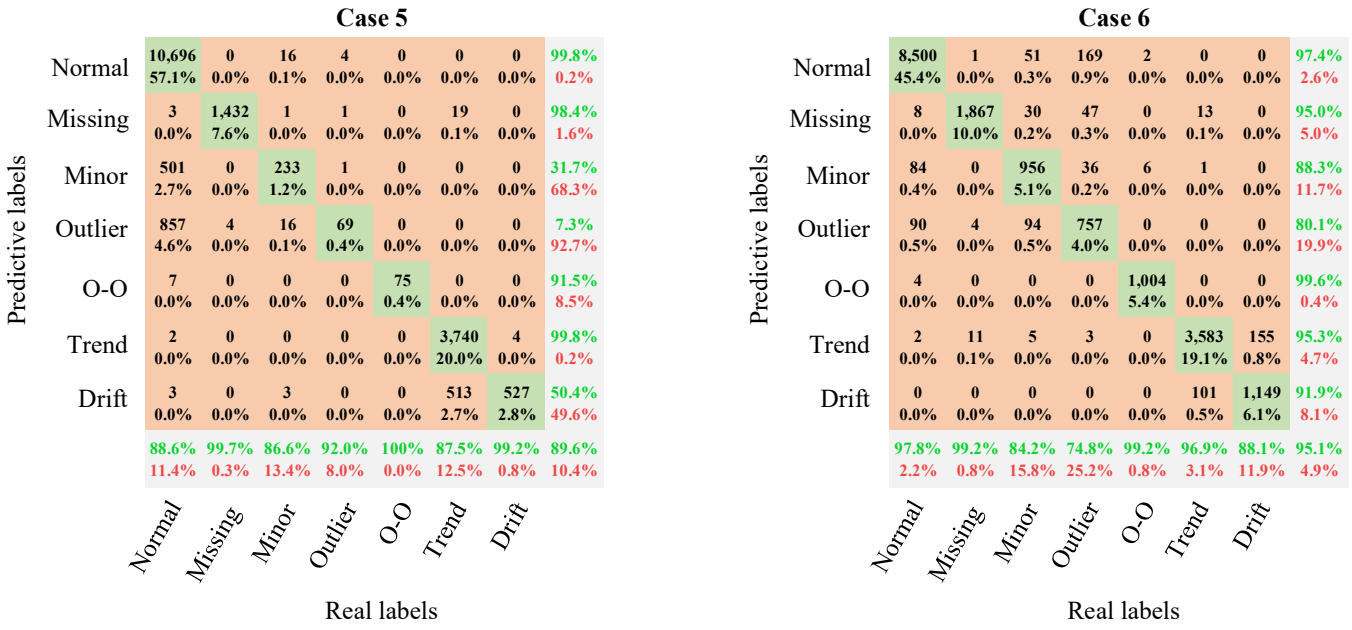


Figure 15. Confusion matrix of Cases 5–6 under testing set.

4.2.2. Results of Precision and Recall

In order to conduct a more detailed analysis of the classification performance of the models for different types of anomaly detection, we present Tables 12–18, showcasing the precision and recall rates for the six types of anomalous data.

Table 12. Precision and recall of "Normal".

Case	1	2	3	4	5	6
Precision	99.50%	94.00%	99.40%	96.40%	99.80%	97.40%
Recall	79.70%	95.40%	89.00%	96.20%	88.60%	97.80%

Table 13. Precision and recall of "Missing".

Case	1	2	3	4	5	6
Precision	97.20%	97.00%	97.90%	96.80%	98.40%	95.00%
Recall	99.50%	99.20%	99.50%	99.30%	99.70%	99.20%

Table 14. Precision and recall of "Minor".

Case	1	2	3	4	5	6
Precision	50.20%	75.90%	60.00%	74.50%	31.70%	88.30%
Recall	89.10%	59.30%	88.50%	76.10%	96.60%	84.20%

Table 15. Precision and recall of "Outlier".

Case	1	2	3	4	5	6
Precision	39.00%	68.70%	34.20%	76.80%	7.30%	80.10%
Recall	94.50%	73.70%	90.30%	75.30%	92.00%	74.80%

Table 16. Precision and recall of "Over-range oscillation".

Case	1	2	3	4	5	6
Precision	95.10%	98.80%	97.80%	99.90%	91.50%	99.60%
Recall	100.00%	99.40%	100.00%	98.70%	100.00%	99.20%

Table 17. Precision and recall of "Trend".

Case	1	2	3	4	5	6
Precision	96.40%	92.40%	99.20%	94.60%	99.80%	95.30%
Recall	91.80%	90.90%	89.50%	95.00%	87.50%	96.90%

Table 18. Precision and recall of "Drift".

Case	1	2	3	4	5	6
Precision	76.80%	76.10%	98.50%	88.40%	50.40%	91.90%
Recall	87.50%	79.60%	97.00%	85.70%	99.20%	88.10%

Table 12 displays the precision and recall of "normal". Specifically, all cases achieve a precision rate of over 94%. Moreover, models trained on balanced datasets tend to exhibit higher precision rates. This indicates that other types of data are rarely misclassified as "normal". As for recall, it can be observed that models trained on imbalanced datasets outperform those trained on balanced datasets in all six cases, with the highest recall rate reaching 97.80%. This can be attributed to the larger number of "normal" samples in the imbalanced datasets, allowing the models to better learn the characteristics of "normal" and improve the data anomaly detection ability, thus leading to higher recall rates. However, due to the abundance of "normal" samples and the learned features, the models trained on imbalanced datasets are more prone to classifying other data types as "normal", resulting in slightly lower precision rates compared to the models trained on balanced datasets.

Table 13 displays the precision and recall of "missing". It can be observed that the model trained using the proposed framework exhibits excellent classification performance for this anomaly pattern. Specifically, the precision and recall of all conditions are over 95% and 99%, respectively. This indicates that the model demonstrates high precision and recall in correctly identifying and classifying instances of "missing" across all conditions.

Tables 14 and 15 show the precision and recall of "minor" and "outlier". According to the aforementioned findings, it can be observed that there is a tendency for the model to misclassify "minor" and "outlier." Specifically, the model is inclined to predict "minor" as either "normal" or "outlier", and "outlier" as either "normal" or "minor", which results in the low values of precision and recall.

The findings derived from the data provided in Table 16 indicate that the model trained by the network demonstrates outstanding classification performance for the "over-range oscillation ". The precision values across all conditions exceed 91%, underscoring the model's ability to accurately identify and classify instances of this particular data type. Furthermore, the recall metrics for all conditions surpass 98%, with several conditions achieving a perfect recall rate of 100%. These results substantiate the model's exceptional capability in correctly capturing instances of "over-range oscillation", and highlight its high sensitivity in detecting these occurrences.

Based on the findings presented in Table 17, it is evident that the classification performance of the models trained on different conditions for "trend" is highly commendable, particularly when evaluating precision. Across all conditions, the precision values exceed 92%, with the performance being notably superior in the balanced condition. When considering recall as an evaluation metric, the results remain highly satisfactory, with the most exceptional model in Case 6 achieving an impressive recall rate of 96.9%. With the exception of conditions 1 and 2, the models trained on imbalanced conditions consistently outperform their counterparts.

Table 18 presents the classification results of "drift". When evaluating precision as a performance metric, it is evident that, except for conditions 3, 4, and 6, the prediction performance of the remaining models is suboptimal. The limited performance of the models trained on conditions 1 and 2 can be attributed to the insufficient number of training samples, leading to inadequate training. In Case 5, a considerable portion of instances belonging to the "trend" were misclassified as "drift", suggesting inadequate training specifically for the "drift" class in that condition. However, in terms of recall, the models trained on balanced conditions demonstrated superior performance, with condition 5 achieving the highest recall rate. This finding suggests that a balanced training set with an adequate representation of "drift" instances enables the models to better identify and classify them accurately.

The comprehensive analysis of the double-channel convolutional neural network model trained in both balanced and imbalanced conditions for data anomaly detection, as presented in this study, provides support for further data anomaly repair and maintenance of structural health monitoring systems. Overall, the main factor influencing the classification performance of the network model is the number of samples containing the specific data anomaly type in the training set. The larger the number of such samples in the training set, the better the training received by the network model, resulting in stronger recognition ability. The optimal model trained using the double-channel convolutional neural network achieved a 95.10% accuracy in data anomaly detection, with a training time of approximately one hour on the selected computational platform. Therefore, compared to traditional manual detection methods, this approach proves to be efficient, accurate, and cost-effective.

5. Comparative Study

To substantiate the efficacy of the proposed method, a performance comparison with an alternative model incorporating CNN was conducted [21]. The used DNN architecture employed in this study comprised one input layer, three hidden layers, and one output layer, all interconnected in a fully connected manner. The F1-score for the models on the test set is presented in Table 19. As the F1-score amalgamates precision rate and recall rate, it serves as a comprehensive metric to assess the classification ability of the model.

Table 19. F1-score of two model in Test set.

Model Data Pattern	DNN	Multi-Channel CNN
Normal	90.0%	94.4%
Missing	93.3%	98.2%
Minor	80.8%	81.0%
Drift	92.0%	94.9%
Trend	90.8%	96.2%
Outlier	83.0%	75.3%
Square	99.6%	99.7%

As can be seen from the table, multi-channel CNN is better than DNN in predicting and classifying all anomaly patterns except for "outlier", which proves the superiority of the proposed method. Consequently, the proposed approach holds significant potential as a reliable reference for data anomaly detection in SHM applications.

6. Conclusions

This paper introduces a deep learning approach for data anomaly detection by proposing a framework based on a double-channel convolutional neural network. The feasibility of this framework is investigated through detailed analysis. Firstly, data preprocessing is performed on the data anomalies, including database augmentation, visualization using CWT, and database partitioning. The database partitioning involves dividing the data into training, validation, and testing sets, as well as creating balanced and imbalanced training

sets for different set size. Six double-channel convolutional neural network models are then trained using the partitioned balanced or imbalanced training sets, and the best models are used. The results show that the best model achieves an accuracy of 95.10%, indicating that the network can accurately detect data anomalies. In addition, the use of a double-channel convolutional neural network for data anomaly classification shows that having a balanced training set does not significantly improve the classification performance of the network model. The main factor influencing the classification performance is the number of samples containing the specific data anomaly type in the training set. The larger the number of such samples in the training set, the better the training received by the network model, resulting in a stronger anomaly detection ability.

Author Contributions: Conceptualization, X.N. and X.L.; methodology, S.L.; validation, Y.R., X.Y. and X.L; formal analysis, Y.R.; investigation, X.L.; resources, X.N.; data curation, Y.R.; writing—original draft preparation, X.Y.; writing—review and editing, X.L.; visualization, Y.R.; supervision, X.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All models, or codes of this study are available from the corresponding author upon reasonable request.

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

References

1. Yang, Q.; Wang, J.; Kim, S.; Chen, H.; Spencer, B.F., Jr. Design and implementation of a SHM system for a heritage timber building. *Smart Struct. Syst.* **2022**, *29*, 561.
2. Li, X.; Yu, W.; Villegas, S. Structural health monitoring of building structures with online data mining methods. *IEEE Syst. J.* **2015**, *10*, 1291–1300. [CrossRef]
3. Aytulun, E.; Soyöz, S. Implementation and application of a SHM system for tall buildings in Turkey. *Bull. Earthq. Eng.* **2022**, *20*, 4321–4344. [CrossRef]
4. Abdelghani, M.; Friswell, M.I. Sensor validation for structural systems with additive sensor faults. *Struct. Health Monit.-Int. J.* **2004**, *3*, 265–275. [CrossRef]
5. Abdelghani, M.; Friswell, M.I. Sensor validation for structural systems with multiplicative sensor faults. *Mech. Syst. Signal Process.* **2007**, *21*, 270–279. [CrossRef]
6. Yuen, K.V.; Mu, H.Q. A novel probabilistic method for robust parametric identification and outlier detection. *Probabilistic Eng. Mech.* **2012**, *30*, 48–59. [CrossRef]
7. Thiyagarajan, K.; Kodagoda, S.; Van Nguyen, L. Predictive Analytics for Detecting Sensor Failure Using Autoregressive Integrated Moving Average Model. In Proceedings of the 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), Siem Reap, Cambodia, 18–20 June 2017; pp. 1926–1931.
8. Wan, H.P.; Ni, Y.Q. Bayesian Modeling Approach for Forecast of Structural Stress Response Using Structural Health Monitoring Data. *J. Struct. Eng.* **2018**, *144*, 04018130. [CrossRef]
9. Zhang, Y.M.; Wang, H.; Wan, H.P.; Mao, J.X.; Xu, Y.C. Anomaly detection of structural health monitoring data using the maximum likelihood estimation-based Bayesian dynamic linear model. *Struct. Health Monit.-Int. J.* **2021**, *20*, 2936–2952. [CrossRef]
10. Kerschen, G.; De Boe, P.; Golinval, J.C.; Worden, K. Sensor validation using principal component analysis. *Smart Mater. Struct.* **2005**, *14*, 36–42. [CrossRef]
11. Goebel, K.; Yan, W.Z. Correcting Sensor Drift and Intermittency Faults With Data Fusion and Automated Learning. *IEEE Syst. J.* **2008**, *2*, 189–197. [CrossRef]
12. Kramer, M.A. Autoassociative neural networks. *Comput. Chem. Eng.* **1992**, *16*, 313–328. [CrossRef]
13. Tamilselvan, P.; Wang, P.F. Failure diagnosis using deep belief learning based health state classification. *Reliab. Eng. Syst. Saf.* **2013**, *115*, 124–135. [CrossRef]
14. Arul, M.; Kareem, A. Data anomaly detection for structural health monitoring of bridges using shapelet transform. *Smart Struct. Syst.* **2022**, *29*, 93–103.
15. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
16. Gibert, X.; Patel, V.; Chellappa, R. Deep Multitask Learning for Railway Track Inspection. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 153–164. [CrossRef]
17. Kim, B.; Cho, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. *Sensors* **2018**, *18*, 3452. [CrossRef]

18. Li, Y.; Zhao, W.; Zhang, X.; Zhou, Q. A Two-Stage Crack Detection Method for Concrete Bridges Using Convolutional Neural Networks. *Ieice Trans. Inf. Syst.* **2018**, *E101D*, 3249–3252. [CrossRef]
19. Maya, S.; Ueno, K.; Nishikawa, T. dLSTM: A new approach for anomaly detection using deep learning with delayed prediction. *Int. J. Data Sci. Anal.* **2019**, *8*, 137–164. [CrossRef]
20. Chen, J.; Pi, D.; Wu, Z.; Zhao, X.; Pan, Y.; Zhang, Q. Imbalanced satellite telemetry data anomaly detection model based on Bayesian LSTM. *Acta Astronaut.* **2021**, *180*, 232–242. [CrossRef]
21. Bao, Y.; Tang, Z.; Li, H.; Zhang, Y. Computer vision and deep learning-based data anomaly detection method for structural health monitoring. *Struct. Health Monit.-Int. J.* **2019**, *18*, 401–421. [CrossRef]
22. Tang, Z.; Chen, Z.; Bao, Y.; Li, H. Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Struct. Control. Health Monit.* **2019**, *26*, e2296.1–e2296.22. [CrossRef]
23. Mao, J.X.; Wang, H.; Spencer, B.F. Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders. *Struct. Health Monit.-Int. J.* **2021**, *20*, 1609–1626. [CrossRef]
24. Zhang, Y.X.; Lei, Y. Data Anomaly Detection of Bridge Structures Using Convolutional Neural Network Based on Structural Vibration Signals. *Symmetry* **2021**, *13*, 1186. [CrossRef]
25. Maleki, S.; Maleki, S.; Jennings, N.R. Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. *Appl. Soft Comput.* **2021**, *108*, 107443. [CrossRef]
26. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
27. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. Dive into Deep Learning. *arXiv* **2021**, arXiv:2106.11342.
28. Mustaqeem, K.S. 1D-CNN: Speech Emotion Recognition System Using a Stacked Network with Dilated CNN Features. *Comput. Mater. Contin.* **2021**, *67*, 4039–4059. [CrossRef]
29. Jian, X.D.; Zhong, H.Q.; Xia, Y.; Sun, L. Faulty data detection and classification for bridge structural health monitoring via statistical and deep-learning approach. *Struct. Control. Health Monit.* **2021**, *28*, e2824. [CrossRef]
30. Cofre-Martel, S.; Kobrich, P.; Lopez Droguett, E.; Meruane, V. Deep convolutional neural network-based structural damage localization and quantification using transmissibility data. *Shock. Vib.* **2019**, *2019*, 9859281. [CrossRef]
31. Liu, Y.; Chen, X.; Wang, Z.; Wang, Z.J.; Ward, R.K.; Wang, X. Deep learning for pixel-level image fusion: Recent advances and future prospects. *Inf. Fusion* **2018**, *42*, 158–173. [CrossRef]
32. Khan, A.; Sung, J.E.; Kang, J.-W. Multi-channel fusion convolutional neural network to classify syntactic anomaly from language-related ERP components. *Inf. Fusion* **2019**, *52*, 53–61. [CrossRef]
33. Neupauer, R.M.; Powell, K.L. A fully-anisotropic Morlet wavelet to identify dominant orientations in a porous medium. *Comput. Geosci.* **2005**, *31*, 465–471. [CrossRef]
34. Dominik, Ł. Mechanical Vibrations Analysis in Direct Drive Using CWT with Complex Morlet Wavelet. *Power Electron. Drives* **2023**, *8*, 65–73.
35. Ren, L.H.; Ye, Z.F.; Zhao, Y.P. Long short-term memory neural network with scoring loss function for aero-engine remaining useful life estimation. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2023**, *237*, 547–560. [CrossRef]
36. Mahmoud, S.; Gaber, M.; Farouk, G.; Keshk, A. Heart Disease Prediction Using Modified Version of LeNet-5 Model. *Int. J. Intell. Syst. Appl.* **2022**, *14*, 1–12. [CrossRef]
37. Bao, Y.; Li, J.; Nagayama, T.; Xu, Y.; Spencer, B.F., Jr.; Li, H. The 1st International Project Competition for Structural Health Monitoring (IPC-SHM, 2020): A summary and benchmark problem. *Struct. Health Monit.-Int. J.* **2021**, *20*, 2229–2239. [CrossRef]
38. Geng, R.; Buelo, C.J.; Sundaresan, M.; Starekova, J.; Panagiotopoulos, N.; Oechtering, T.H.; Lawrence, E.M.; Ignaciuk, M.; Reeder, S.B.; Hernando, D. Automated MR Image Prescription of the Liver Using Deep Learning: Development, Evaluation, and Prospective Implementation. *J. Magn. Reson. Imaging JMRI* **2023**, *58*, 429–441. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.