

Computer vision and deep learning-based data anomaly detection method for structural health monitoring

Structural Health Monitoring

1–21

© The Author(s) 2018

Reprints and permissions:

sagepub.co.uk/journalsPermissions.nav

DOI: 10.1177/1475921718757405

journals.sagepub.com/home/shm



Yuequan Bao^{1,2,3}, Zhiyi Tang^{1,2,3}, Hui Li^{1,2,3}  and Yufeng Zhang^{4,5}

Abstract

The widespread application of sophisticated structural health monitoring systems in civil infrastructures produces a large volume of data. As a result, the analysis and mining of structural health monitoring data have become hot research topics in the field of civil engineering. However, the harsh environment of civil structures causes the data measured by structural health monitoring systems to be contaminated by multiple anomalies, which seriously affect the data analysis results. This is one of the main barriers to automatic real-time warning, because it is difficult to distinguish the anomalies caused by structural damage from those related to incorrect data. Existing methods for data cleansing mainly focus on noise filtering, whereas the detection of incorrect data requires expertise and is very time-consuming. Inspired by the real-world manual inspection process, this article proposes a computer vision and deep learning-based data anomaly detection method. In particular, the framework of the proposed method includes two steps: data conversion by data visualization, and the construction and training of deep neural networks for anomaly classification. This process imitates human biological vision and logical thinking. In the data visualization step, the time series signals are transformed into image vectors that are plotted piecewise in grayscale images. In the second step, a training dataset consisting of randomly selected and manually labeled image vectors is input into a deep neural network or a cluster of deep neural networks, which are trained via techniques termed stacked autoencoders and greedy layer-wise training. The trained deep neural networks can be used to detect potential anomalies in large amounts of unchecked structural health monitoring data. To illustrate the training procedure and validate the performance of the proposed method, acceleration data from the structural health monitoring system of a real long-span bridge in China are employed. The results show that the multi-pattern anomalies of the data can be automatically detected with high accuracy.

Keywords

Structural health monitoring, data anomaly detection, computer vision, deep learning, stacked autoencoder deep neural network

Introduction

With the aim of sensing and understanding the behavior of complex engineering structures, structural health monitoring (SHM) systems are widely implemented in the fields of aerospace, civil, and mechanical engineering.^{1–3}

In the field of civil engineering, several types of infrastructure have been fitted with monitoring systems that include various types of sensors.^{4–9} A large amount of SHM data are generated every day. Recently, SHM applications have expanded from single units of infrastructure to infrastructure networks and may soon be used to cover whole cities. Indeed, such SHM technology will collect more and more data. Therefore, more efficient methods of processing these SHM data, based

on the algorithms used to handle big data, should be developed.

¹Key Lab of Structures Dynamic Behavior and Control of the Ministry of Education, Harbin Institute of Technology, Harbin, China

²Key Lab of Intelligent Disaster Mitigation of the Ministry of Industry and Information Technology, Harbin Institute of Technology, Harbin, China

³School of Civil Engineering, Harbin Institute of Technology, Harbin, China

⁴The State Key Laboratory on Safety and Health of In-service Long-span Bridges, Nanjing, China

⁵JSTI Group, Nanjing, China

Corresponding author:

Yuequan Bao, School of Civil Engineering, Harbin Institute of Technology, Harbin 150090, China.
Email: baoyuequan@hit.edu.cn

One common issue with data-intensive systems concerns data anomalies. These are widespread in SHM systems because of malfunctions caused by imperfect sensors and the poor quality of data transmission. These anomalies not only cause false alarms but also affect the structural performance assessment. Furthermore, data preprocessing (or data cleansing)² is time- and labor-intensive, making it very expensive. Therefore, effective data cleansing algorithms that make SHM data more reliable for online monitoring and further analysis are urgently required.

To eliminate anomalies in the measured data, the hardware redundancy could be increased by placing backup sensors at all sensor locations; the backup sensor switches on when the active sensor output exhibits an anomaly.¹⁰ For an SHM system with a large number of sensors, this would be very expensive. Another approach involves the deployment of self-validating (SEVA) sensors.^{11,12} Compared to traditional sensors, SEVA sensors have stronger computing units that self-diagnose the measurement value online and output multi-type data such as raw measurement value, raw uncertainty, validated measurement value, and validated uncertainty. Some prototype SEVA sensors have been developed.^{12,13} The limitation of SEVA sensors is that the methods of uncertainty calculation and anomaly correction are expertise-dependent in specific scenarios, which differ according to the measurands and anomaly patterns. The recently developed wireless smart sensor Xnode¹⁴ is similar to SEVA sensors, being capable of self-diagnosis, self-calibration, self-identification, and self-adaptation.

In terms of data transmission, wired systems are usually more reliable than wireless transmission. Data anomalies such as data loss are more common in wireless data transmission. In wireless systems, high link quality in the sensor network is crucial to ensuring high data fidelity. Link quality varies with transmission power, distance between sensor nodes, and operational conditions (electromagnetic field environment, temperature, and humidity effects). Enhancements and metrics for link quality have been widely studied.^{15–17}

Currently, data defects are an inevitable problem in SHM systems. In the process of data cleansing, data are accepted, rejected, or first corrected and then accepted into the subsequent process. Many data cleansing methods have been proposed. One technique diagnoses outliers, which constitute a category of frequent anomalies (note that outliers may not be erroneous, but actually very informative data. For example, the intensive earthquake response of a structure is anomalous compared with the data monitored in normal operational conditions, but is very useful data) and uses statistical methods to diagnose and filter outliers with a certain confidence interval.^{18,19} To eliminate

measurement noise, spectral analysis techniques such as singular-value decomposition (SVD),²⁰ principal component analysis (PCA),^{21,22} and wavelet analysis^{23,24} are widely used. Measurement noise can be reduced via a signal decomposition–reconstruction process. To reduce burst noise, which is sparse in the time domain but has a large amplitude, a principal component pursuit denoising method was proposed, which can rectify gross noise data in the ambient vibration response of civil structures competently.²⁵ In addition, machine learning techniques employing autoassociative neural networks²⁶ have been used for noise reduction and anomaly detection. The key feature of an autoassociative network is a bottleneck layer that enforces learning for information compression. Noise can then be reduced in the output data, and the residuals between the input and output data can be used to detect anomalies. However, the methods mentioned above are mainly for noise filtering, or handling simple anomalies such as data loss and sporadic outliers, which makes them incapable of handling complex *in situ* data with multiple data anomaly patterns. Therefore, the data cleansing process often demands expert intervention to determine whether a piece of data should be accepted, rejected, or corrected by a selective approach. This manual intervention is laborious, time-consuming, and error-prone. Little effort has been made to develop a general-purpose and automatic data anomaly detection method. Recently, computer vision techniques and deep learning techniques have become popular research topics in SHM. Tamilselvan and Wang²⁷ used deep belief networks to diagnose failures in aircraft engines and electric power transformers. Wei et al.²⁸ identified the cable tension patterns of a cable-stayed bridge under normal operating conditions using clustering techniques. Li et al.²⁹ addressed gearbox fault diagnosis using a multimodal deep support vector classification approach. Ye et al.³⁰ conducted vision-based dynamic displacement measurements of a long-span bridge using a pattern matching algorithm. Koch et al.³¹ reviewed the computer vision-based defect detection and condition assessment of civil infrastructure. More recently, Cha et al.³² used convolutional neural networks to detect structural crack damage from images and Yang et al.³³ identified high-spatial-resolution vibration modes from video measurements. Yeum et al.³⁴ developed an image localization technique with an aerial vision sensor platform to automatically screen out the small fraction of images that are of interest.

Inspired by practical manual intervention (essentially biological vision-based data acquisition and brain-based decision-making, which have high intuition and comprehensive abstraction ability), we propose a computer vision and deep learning-based data anomaly detection method. The proposed method can

automatically identify multiple patterns of data anomalies in SHM systems with high accuracy.

Framework of the computer vision and deep learning-based data anomaly detection method

The framework of the proposed method includes two main steps, as shown in Figure 1: (1) data conversion by data visualization and (2) deep neural network (DNN) training for data anomaly classification. The procedure imitates human biological vision and logical thinking. In the data visualization step, the time series signals are transformed into image vectors by splitting them into sections and plotted in grayscale images. In the second step, a training set consisting of randomly selected and manually labeled image vectors is fed into DNNs, which is then trained via techniques termed stacked autoencoders and greedy layer-wise training. The trained DNNs detect potential anomalies in large amounts of SHM data.

Data visualization

To automatically detect multiple anomalies in SHM data as a human expert does, the first step is data

visualization. Raw data are split into hour-long segments, then plotted in figures and saved as image files, as shown in Figure 1. Splitting data can be thought of as windowing data with no overlap between two adjacent windows. Each image is 8-bit grayscale with 100×100 pixel resolution, which is sufficient to represent the graphical features of the data while ensuring reasonably low storage requirements. At such resolution, the size of each image file is less than 2 KB, and so the total file size for one channel per year is about 17 megabyte (MB). The image vector assembled by sequentially connecting pixel columns is the input of the neural network. The data visualization procedure can be regarded as a feature selection process.

Supervised training and feature extraction

Label data. Because the proposed method evaluates data quality via “observing” figures as a human expert would, the graphical features of each image are the key points, and they are selected as the criteria for classification. However, data anomalies in SHM systems vary according to the structure, sensor type, and placement location. To acquire prior knowledge of a certain structure for labeling, such as the form of each pattern of anomaly and the total number of patterns, it is

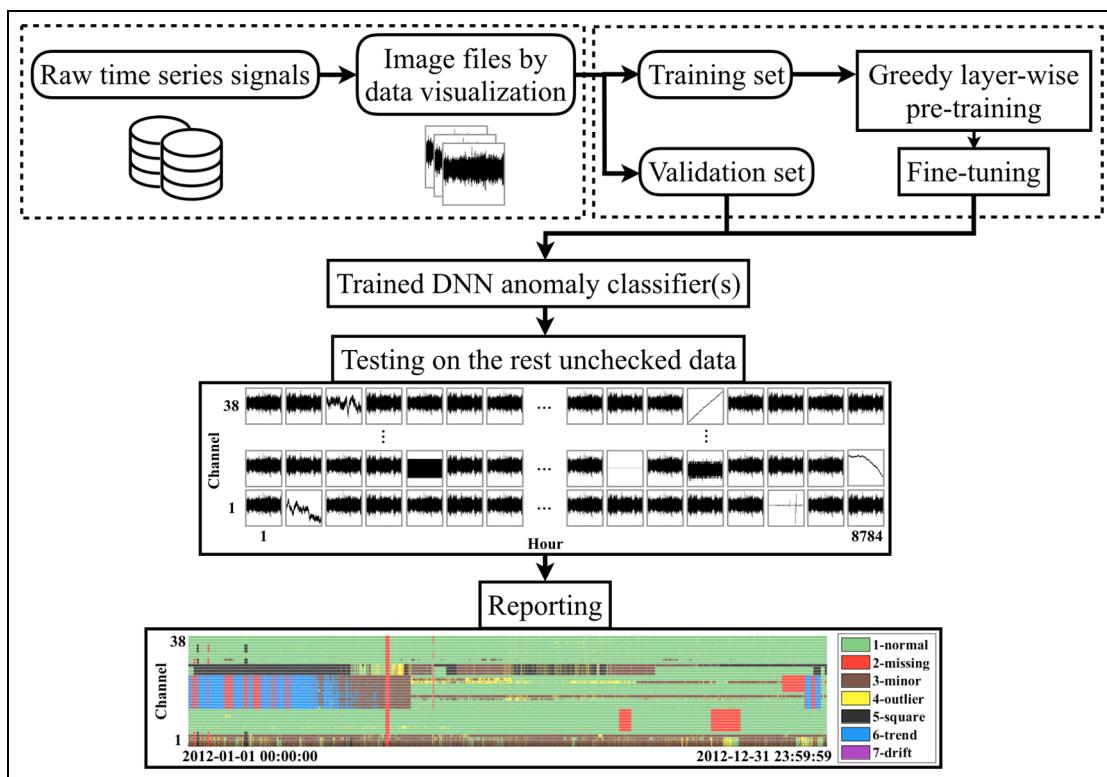


Figure 1. Framework of the proposed data anomaly detection method.

necessary for a human expert to observe images of the collected data.

Next, labeled training samples are randomly selected and manually labeled with an index of the anomaly pattern. These are denoted as $(\mathbf{s}^{(p)}, l^{(p)})$, where $\mathbf{s}^{(p)}$ is the p th data piece, and $l^{(p)}$ is the corresponding anomaly pattern index. Generally, a training ration of 1%–5% is sufficient to cover all the representative data anomaly patterns. In this article, a single-label classification method is employed, that is, when an image shows multiple anomalous features, the label is determined by the major intrinsic feature for all data in that image.

Construction and training of DNNs. The process of image acquisition is different between human vision and computer vision. In a three-dimensional (3D) space, humans watch images in full view at once, whereas computers “see” an image by scanning its pixel columns serially. Hence, to show images to a computer, the 100×100 image pixel array is transformed into an 10000×1 image vector by sequentially stacking the pixel columns.

To imitate the human decision-making process for a deep understanding of the characteristics of anomalies, the fast and effective stacked autoencoder DNN is

employed.³⁵ DNN is a type of artificial neural network (ANN) with more than one hidden layer, which allows them to learn high-level abstractions of the input. In the late 1990s, DNNs were largely discarded by the machine learning community because of their significantly long training time. In 2006, a breakthrough was achieved by introducing an unsupervised layer-wise pre-training procedure, which made it possible to train a deep architecture in a short time with unlabeled data.³⁶ The stacking of pre-trained layers overcomes the disappearing gradient of the weights tuning.^{35,37}

A two-hidden-layer stacked autoencoder neural network is constructed to demonstrate the two-stage training procedure (Figure 2). Let $\mathbf{s}^{(p)}$ denote the p th piece of source data out of a total of P data pieces, f_s denote the sampling frequency, and T_d denote the duration of data in each plot. Therefore, the sample point number of each $\mathbf{s}^{(p)}$ is $f_s \times T_d$. If the dimension of an image pixel array is $n \times n$, the dimension of the corresponding image vector $\mathbf{x}^{(p)}$ is $n^2 \times 1$, as shown in Figure 3.

Generally, in a neural network, each unit has an activation value, defined as

$$a_i^{(l+1)} = f(z_i^{(l+1)}) \quad (1)$$

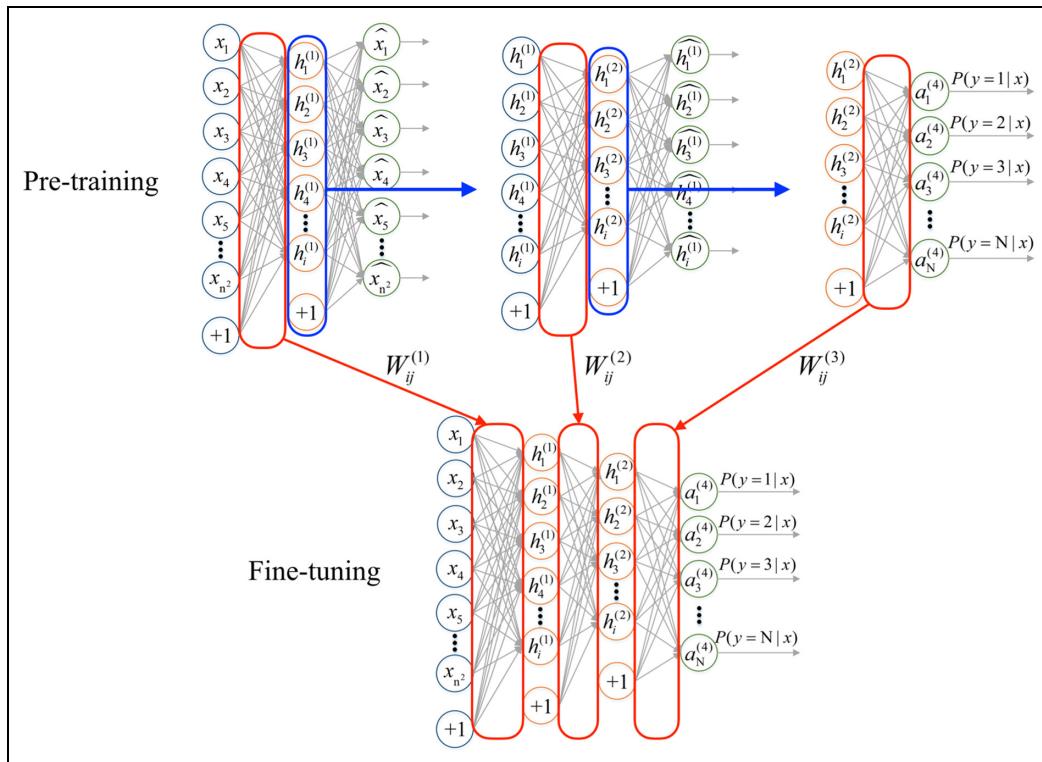


Figure 2. Construction of a two-hidden-layer neural network: in each autoencoder, the weights between the input layer and hidden layer are extracted, and the node values of the hidden layer of each training sample are fed into the subsequent autoencoder or softmax classifier.

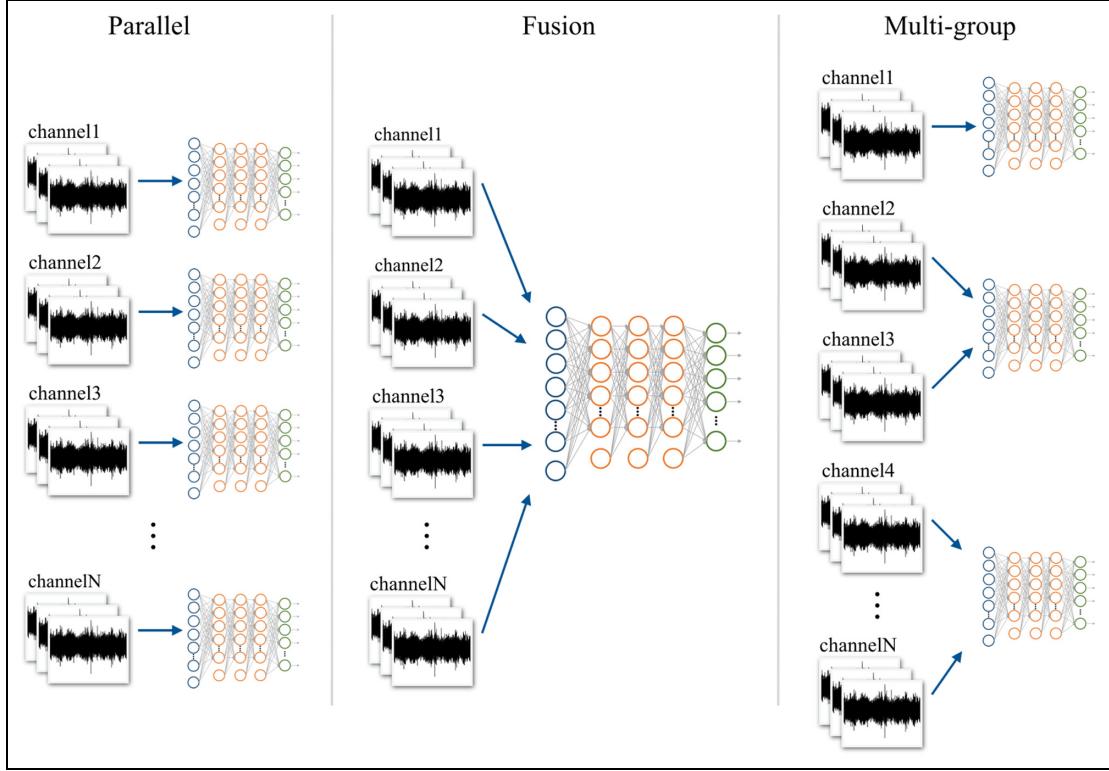


Figure 3. Layout of DNNs.

$$z_i^{(l+1)} = \sum_{j=1}^J W_{ij}^{(l)} a_j^{(l)} + b_i^{(l)} \quad (2)$$

where $a_j^{(l)}$ is the general notation for the activation of unit j in layer l ; in the input layer, $a_j^{(1)} = x_j$, whereas in the hidden layers, $a_j^{(l)} = h_j^{(l)}$. $z_i^{(l+1)}$ is the input value of unit i in layer $l+1$; n is the size of layer l ; $W_{ij}^{(l)}$ is the weight associated with the link between unit j in layer l and unit i in layer $l+1$; $b_i^{(l)}$ is the bias associated with unit i in layer $l+1$, which acts as the constant term in the polynomial of $z_i^{(l+1)}$; and $f(\cdot)$ is the activation function.

For concise notation, vectorization is used in an element-wise manner

$$\begin{aligned} \mathbf{a}^{(1)} &= \mathbf{x} \\ \mathbf{a}^{(l+1)} &= f(\mathbf{z}^{(l)}) \\ &= f(\mathbf{W}^{(l)} \mathbf{a}^{(l)} + \mathbf{b}^{(l)}) \end{aligned} \quad (3)$$

where

$$\begin{aligned} \mathbf{a}^{(l)} &= [a_1^{(l)}, a_2^{(l)}, \dots, a_J^{(l)}]^T \\ \mathbf{b}^{(l)} &= [b_1^{(l)}, b_2^{(l)}, \dots, b_J^{(l)}]^T \\ \mathbf{W}^{(l)} &= \begin{bmatrix} W_{11}^{(l)} & \cdots & W_{1J}^{(l)} \\ \vdots & W_{ij}^{(l)} & \vdots \\ W_{J1}^{(l)} & \cdots & W_{JJ}^{(l)} \end{bmatrix} \end{aligned} \quad (4)$$

An autoencoder is a three-layer ANN with input and output layers of the same size, in which the output values are set to be equal to the inputs. Thus, the training for an autoencoder is unsupervised. In this article, the sigmoid function is used as the activation function for the hidden layer and output layer

$$f(z) = \frac{1}{1+e^{-z}} \quad (5)$$

The first stage of constructing a two-hidden-layer stacked autoencoder neural network is the greedy layer-wise pre-training. Image vectors $\mathbf{x}^{(p)}$ are input to the first autoencoder, the mean squared error (MSE) is set as the objective function, and the scaled conjugate gradient (SCG) algorithm³⁸ is employed to tune the weights. After the training has been completed, the weights between the input layer and the hidden layer are extracted and deployed into the four-layer neural network as $\mathbf{W}^{(1)}$. The node values of the hidden layer of each training sample, $\mathbf{h}^{(1)}$, are fed into the second autoencoder. Once the training has been completed, the weights between the input layer and the hidden layer of the second autoencoder are extracted as $\mathbf{W}^{(2)}$. The output values of each autoencoder are useless. Finally, in pre-training, $\mathbf{h}^{(2)}$ and the corresponding label of each training sample are input into the softmax classifier, which is capable of multi-class classification. The activation function of the output layer in a softmax classifier is defined as

$$\mathbf{y}(\mathbf{x}; \boldsymbol{\theta}) = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} P(\text{prediction} = 1 | \mathbf{x}; \boldsymbol{\theta}) \\ \vdots \\ P(\text{prediction} = k | \mathbf{x}; \boldsymbol{\theta}) \\ \vdots \\ P(\text{prediction} = K | \mathbf{x}; \boldsymbol{\theta}) \end{bmatrix} = \frac{1}{\sum_{m=1}^K e^{\boldsymbol{\theta}_m \mathbf{x}}} \begin{bmatrix} e^{\boldsymbol{\theta}_1 \mathbf{x}} \\ \vdots \\ e^{\boldsymbol{\theta}_k \mathbf{x}} \\ \vdots \\ e^{\boldsymbol{\theta}_K \mathbf{x}} \end{bmatrix} \quad (6)$$

where $\mathbf{y}(\mathbf{x}; \boldsymbol{\theta})$ is a K -dimensional output vector; \mathbf{x} is the input column vector; $P(y=k|\mathbf{x}; \boldsymbol{\theta})$ is the predicted probability for a certain class k , and therefore, $\sum_{k=1}^K y_k = 1$; $\boldsymbol{\theta}_k$ is the weight row vector associated with the link between all units in the input layer and the k th unit in the output layer; and $\boldsymbol{\theta}$ is the weight matrix

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_K \end{bmatrix} \quad (7)$$

Next, the cross-entropy is utilized as the objective function to measure the error between the actual labels and the predicted classes. This is defined as

$$E(\boldsymbol{\theta}) = -\frac{1}{P} \left[\sum_{p=1}^P \sum_{k=1}^K 1\{l^{(p)} = k\} \log(y_k^{(p)}) \right] \quad (8)$$

where $E(\boldsymbol{\theta})$ is the objective function; P is the number of samples; $1\{\cdot\}$ is an indicator function in which $1\{\text{a true statement}\} = 1$ and $1\{\text{a false statement}\} = 0$; $y_k^{(p)}$ is the probability for class k of sample p , that is, $y_k^{(p)} = e^{\boldsymbol{\theta}_k \mathbf{x}^{(p)}} / \sum_{m=1}^K e^{\boldsymbol{\theta}_m \mathbf{x}^{(p)}}$; and $l^{(p)}$ is the label of sample p .

Substituting the notation in this training procedure, the activation function and objective function of softmax classifier are, respectively, given as

$$\mathbf{y}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{a}^{(4)}(\mathbf{h}^{(2)}; \mathbf{W}^{(3)}) = \begin{bmatrix} a_1^{(4)} \\ \vdots \\ a_k^{(4)} \\ \vdots \\ a_K^{(4)} \end{bmatrix} = \begin{bmatrix} P(\text{prediction} = 1 | \mathbf{h}^{(2)}; \mathbf{W}^{(3)}) \\ \vdots \\ P(\text{prediction} = k | \mathbf{h}^{(2)}; \mathbf{W}^{(3)}) \\ \vdots \\ P(\text{prediction} = K | \mathbf{h}^{(2)}; \mathbf{W}^{(3)}) \end{bmatrix} = \frac{1}{\sum_{m=1}^K e^{\mathbf{W}_m^{(3)} \mathbf{h}^{(2)}}} \begin{bmatrix} e^{\mathbf{W}_1^{(3)} \mathbf{h}^{(2)}} \\ \vdots \\ e^{\mathbf{W}_k^{(3)} \mathbf{h}^{(2)}} \\ \vdots \\ e^{\mathbf{W}_K^{(3)} \mathbf{h}^{(2)}} \end{bmatrix} \quad (9)$$

$$E(\mathbf{W}) = -\frac{1}{P} \left[\sum_{p=1}^P \sum_{k=1}^K 1\{l^{(p)} = k\} \log \frac{e^{\mathbf{W}_k^{(3)} \mathbf{h}^{(2)}(p)}}{\sum_{m=1}^K e^{\mathbf{W}_m^{(3)} \mathbf{h}^{(2)}(p)}} \right] \quad (10)$$

where $M_{6,1}$ denotes the weight vector on row $M_{7,8}$ in $\mathbf{W}^{(3)}$ and $[\mathbf{h}^{(2)}]^{(p)}$ denotes $\mathbf{h}^{(2)}$ of training sample p .

$\mathbf{W}^{(3)}$ can be obtained by minimizing the objective function using SCG tuning. Thus, a four-layer neural network is constructed through the orderly stacking of an input layer, two hidden layers, and a softmax layer using the same activation function as the autoencoders. Note that the initial weights are set as $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and $\mathbf{W}^{(3)}$ instead of being randomly generated.

The second stage is fine-tuning. The training set consisting of $(\mathbf{x}^{(p)}, l^{(p)})$ is input to train the big network using SCG tuning. The use of the initial weights obtained in the pre-training process ensures rapid convergence.^{35,37}

The deployment of well-trained DNNs is flexible depending on the SHM system structure. An SHM system of large-scale civil infrastructure usually consists of several subnets, each containing many nodes in multiple sensor types. The faults that occur in subsystems cause nodes in the same subnet to share homologous data anomalous patterns, such as the total number of anomalous data patterns for each sensor and the graphical characteristics of each pattern. Moreover, a particular pattern of data anomaly will appear differently in different sensors and different subsystems. Therefore, to achieve better classification performance, it is advisable to design a suitable layout for the networks before training. Three network layout strategies for DNNs are proposed in this article. These are parallel, fusion, and multi-group layouts, as shown in Figure 3. In the parallel layout, each channel has a private network to detect its data anomaly. This layout has better local performance but is incapable of sharing information between sensors. In contrast, the fusion layout mixes all sensors' data to train a global network. The multi-group layout combines the above two, that is, all sensors in the same group share a network that is trained by their fusion dataset. If the sensor subsystem structure information of the SHM system is known, then the sensors can be divided into different groups according to the actual sensor subsystem structure and the multi-group layout can be employed. If the sensor subsystem structure information is unknown, the DNN layout should be chosen according to the similarity of the same anomaly category between each sensor. When there are huge differences in the same category of anomaly between different sensors, the parallel layout is suitable; if the anomalies of each category are similar between different sensors, the fusion layout is more practical.

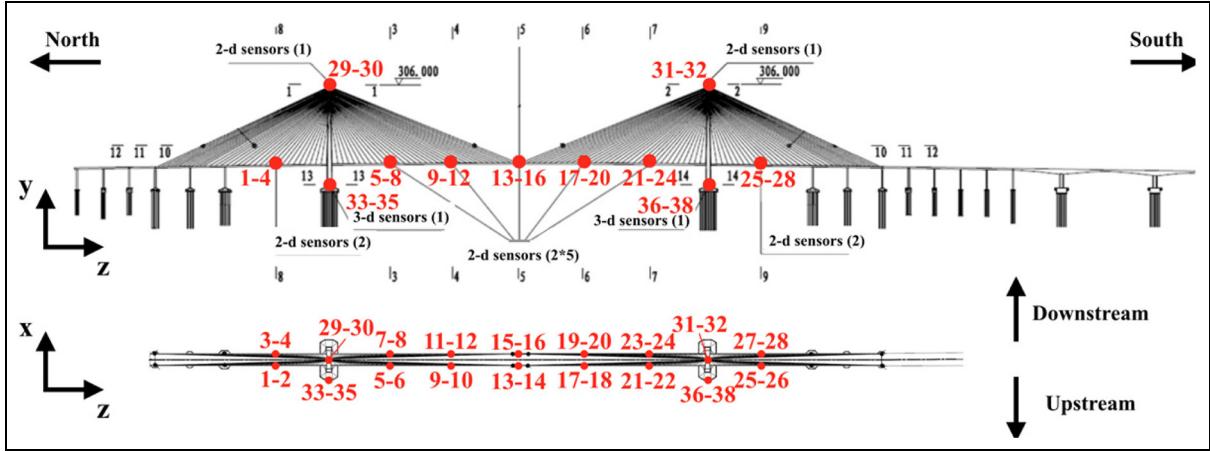


Figure 4. The investigated bridge and placement of accelerometers on the deck and towers.

Table I. Accelerometer information.

Position	Channel	Direction
Deck	1	y
	2	x
	3	y
	4	x
	5	y
	6	x
	7	y
	8	x
	9	y
	10	x
	11	y
	12	x
	13	y
Deck	14	x
	15	y
	16	x
	17	y
	18	x
	19	y
	20	x
	21	y
	22	x
	23	y
	24	x
	25	y
	26	x
Deck	27	y
	28	x
Tower top	29	x
	30	z
	31	x
	32	z
Tower base	33	x
	34	z
	35	y
	36	x
	37	z
	38	y

Case study

Introduction of the SHM system of a long-span bridge

As a case study, we consider a long-span cable-stayed bridge in China (see Figure 4). This bridge has a main span of 1088 m, two side spans of 300 m each, and two 306-m-high towers. Since its completion in 2008, the bridge has had an SHM system that includes accelerometer, anemometer, strain gauge, global positioning system (GPS), thermometer, and so on. Information from the accelerometers is shown in Figure 4 and Table 1. The total number of channels used by the accelerometers on the deck and the towers is 38, including 16 two-channel accelerometers on the deck and the top of the towers, and 2 three-channel accelerometers at the bottom of the towers.

In this case study, only the acceleration data measured from the SHM system are considered, and the anomalies of the acceleration data are divided into six patterns: missing, minor, outlier, square, trend, and drift. Table 2 gives a brief description of the characteristics of these six patterns of data anomalies. Figure 5 shows the typical anomalies for each pattern. Such data anomalies are common not only in the acceleration data but also in GPS data, strain data, wind speed data, and so on.

Data visualization and labeling

All acceleration data measured in 2012 were plotted in hourly 8-bit grayscale images at 100 × 100 pixel resolution, making a total of 333,792 samples (here, one image is one sample). For a sampling ratio of 3%, 10,014 samples were randomly chosen and labeled, 50% of which generated the training set with the other

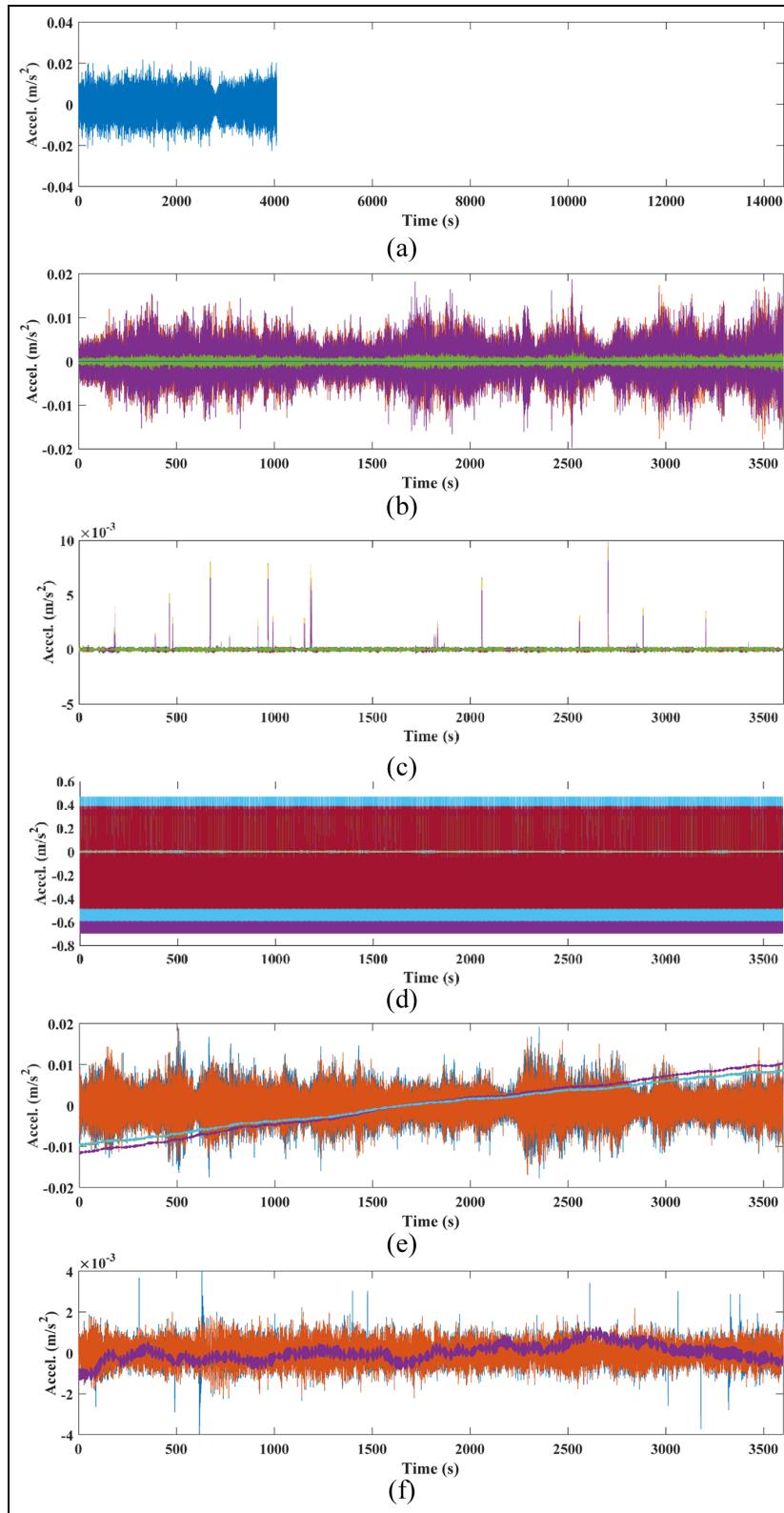
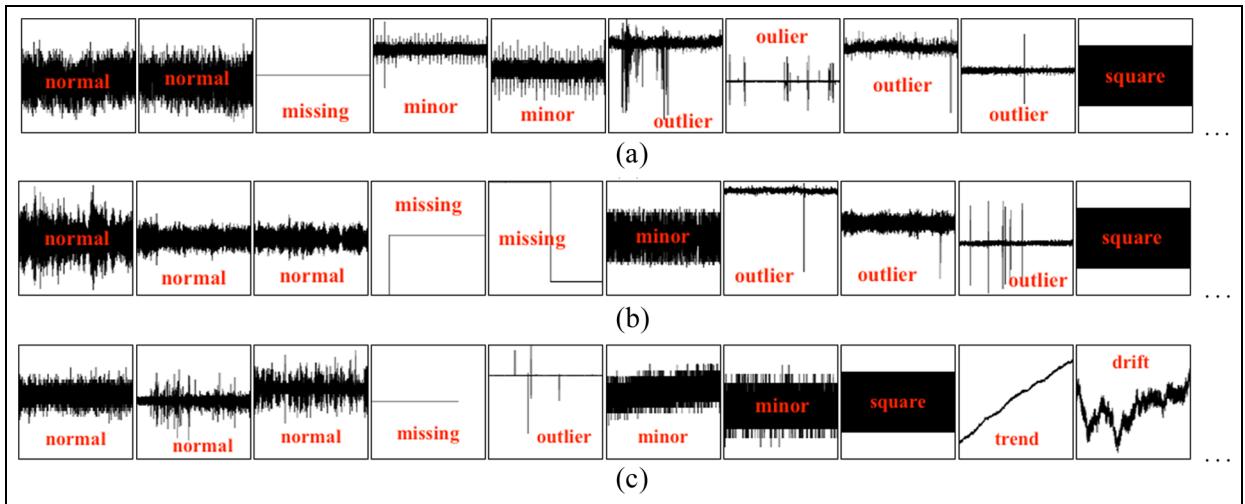
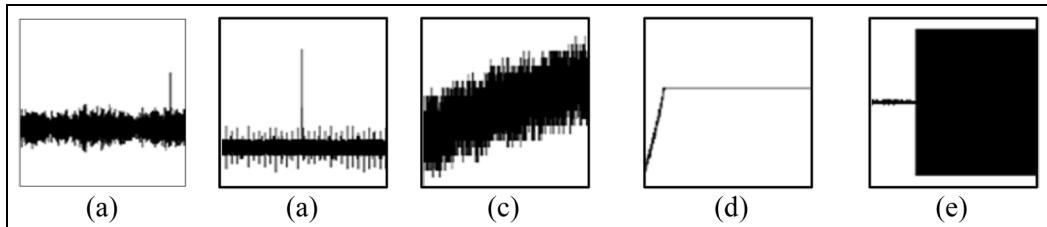


Figure 5. Representative anomalies of acceleration data from SHM system of the bridge: (a) missing values, (b) minor values (the slim blue curve near $y = 0$), (c) outliers, (d) square (oscillatory out-range), (e) trend with minor values, and (f) drift with minor values; the normal values are plotted in (c), (e), and (f) for comparison with anomalies.

Table 2. Description of the six patterns of anomaly.

No.	Anomaly patterns	Description
1	Missing	Most/all of data are missing, most/all of the area of the image is blank
2	Minor	Vibration response oscillates with a tiny amplitude, so the minimum resolution of measurement is visibly unsmooth
3	Outlier	One or more outliers appear in the image of a data piece
4	Square	Vibration response oscillates abnormally over the range of accelerometer
5	Trend	Vibration response is non-stationary with monotonous trend
6	Drift	Vibration response is non-stationary with random drift

**Figure 6.** Examples of data visualization for channels 1, 2, and 3 (samples are reordered by anomaly patterns): (a) channel 1, (b) channel 2, and (c) channel 3.**Figure 7.** Multi-anomalous feature in a sample: (a) is labeled as normal, ignoring the single outlier; (b) is labeled as minor, ignoring the single outlier; (c) is labeled as minor in spite of the mild trend; (d) is labeled as missing, despite the initial trend; and (e) is labeled as square, despite initially being normal.

half used as the validation set. As examples, Figure 6 shows 10 images each for channels 1, 2, and 3 from the training set. Note that the coordinate system is invisible, because the duration and amplitude information of the vibration response are not required for an outline-based classification. Figure 7 demonstrates five examples of the labeling when a multi-anomalous feature exists in a sample, as discussed in section “Supervised training and feature extraction.”

Design and training of DNN

In this example, detailed information about the sensors in the SHM system of the bridge is unknown. Thus, the multi-group DNN layout may be infeasible. In addition, some data anomaly patterns are relatively rare in the training dataset and are prone to be missed, which will reduce the data anomaly detection accuracy of the trained DNN. The fusion DNN layout is employed to

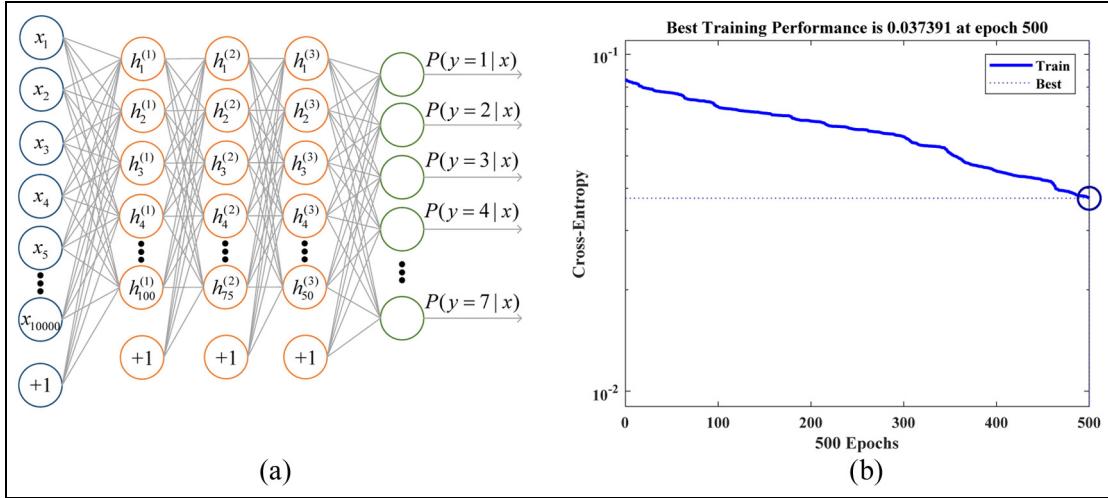


Figure 8. DNN architecture and training results: (a) architecture of the DNN and (b) convergence in training process.

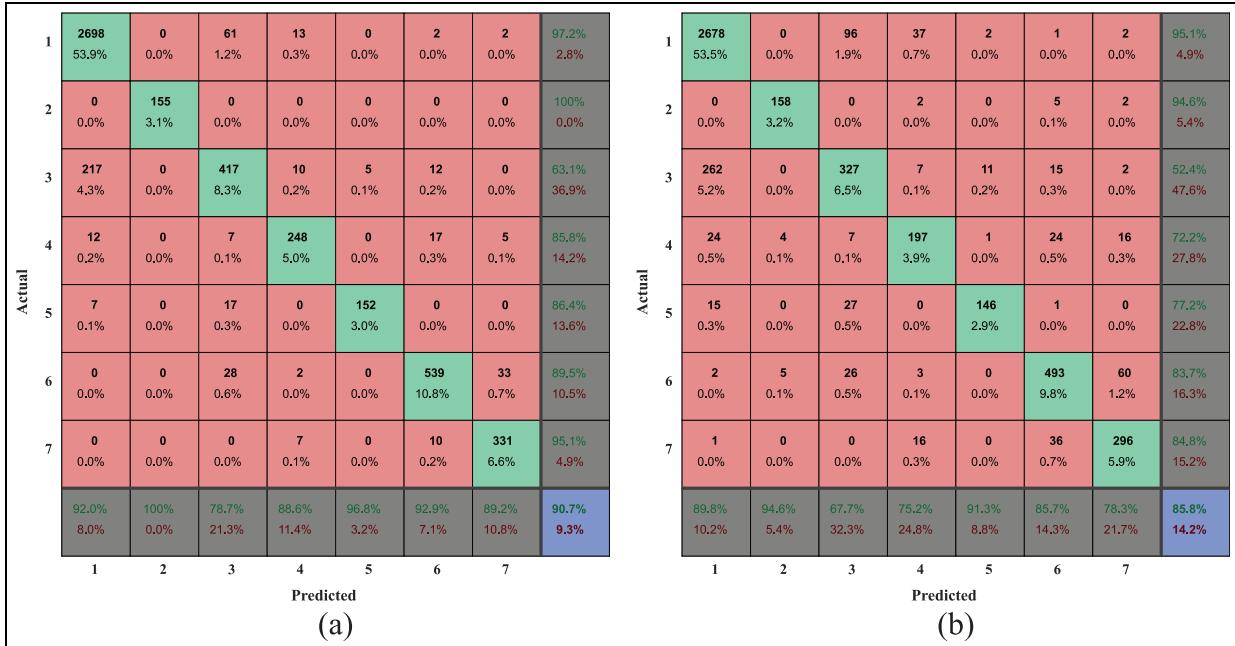


Figure 9. Confusion matrix: (a) training set and (b) validation set.

Anomaly patterns: 1—normal, 2—missing, 3—minor, 4—outlier, 5—square, 6—trend, and 7—drift.

mitigate this problem by sharing the data anomaly patterns that have been labeled.

The architecture of the designed DNN is shown in Figure 8(a). The input layer has 10,000 nodes, because the visualized image is a $10,000 \times 1$ vector in which each element is input into one node of the input layer. The output layer has seven nodes corresponding to six patterns of data anomalies and normal data. There are three hidden layers with 100, 75, and 50 nodes,

respectively. The decreasing number of nodes in each hidden layer is designed to extract the higher abstraction features for generalization, which can overcome the risk of overfitting due to the unbalanced datasets collected by random sampling. Early stopping is employed in the training stage, and the performance is demonstrated in Figure 8(b), which indicates that the convergence measured by cross-entropy becomes the optimal at epoch 500. Figure 9(a) and (b) is the

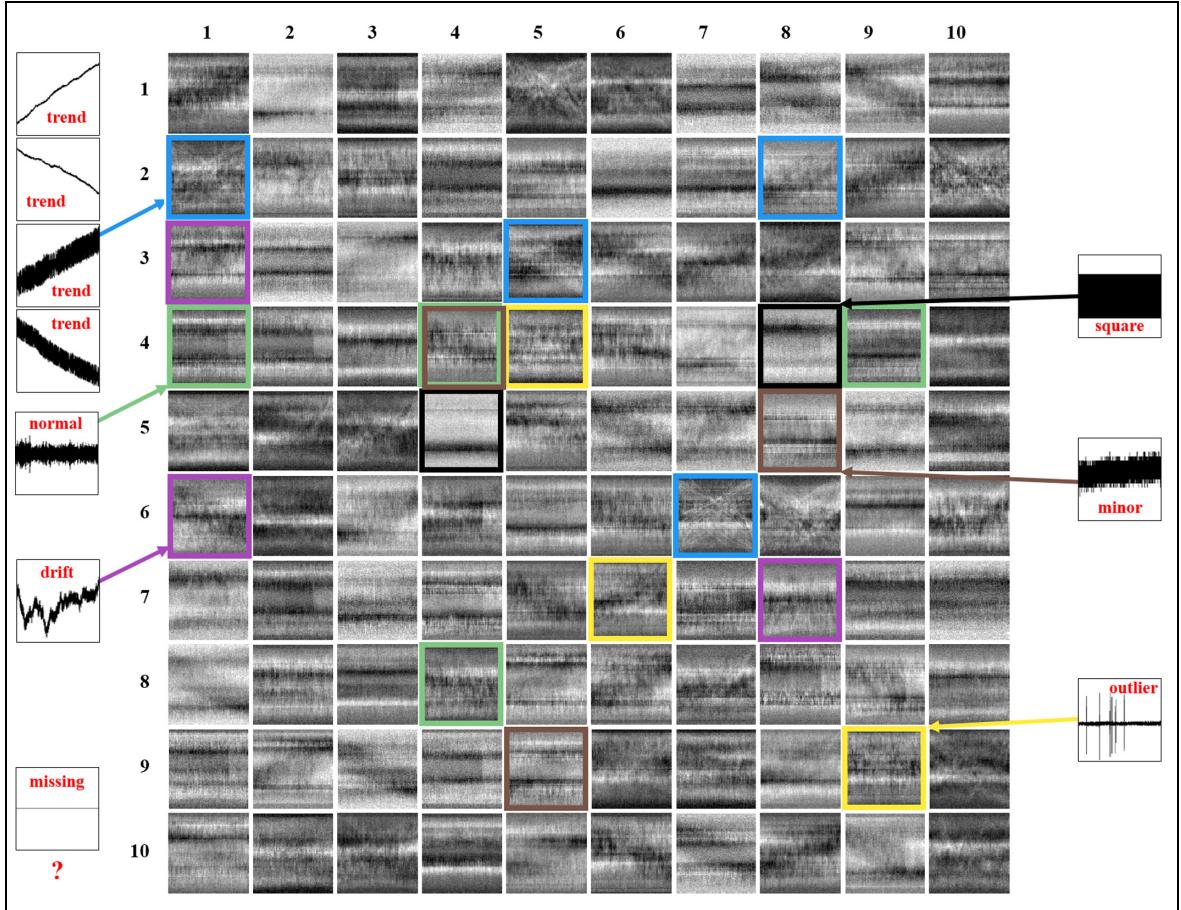


Figure 10. Feature visualization of the first hidden layer.

confusion matrices for inspecting the classification results; the training set and validation set give accuracy levels of 90.7% and 85.8%, respectively.

The recall (rightmost column) is the ratio of the true positive (TP) number and the total number of times a particular pattern appears in the actual dataset. This metric evaluates the reliability of the classifier from the input facts. The recall of the “minor” anomaly pattern on the training and validation sets is only 63.1% and 52.4%, respectively. This poor performance is because portions of the “minor” samples are misclassified as “normal,” although the misclassification ratio is low for the “normal” data.

The precision (bottom row of the confusion matrix) is a ratio of the TPs to the total number of a particular class in the predicted results. This metric evaluates the reliability of the classifier based on the output predictions. The precision of the “minor” anomaly pattern is higher than its recall, reaching 78.7% and 67.7%, respectively. This improvement is because fewer samples from other pattern are misclassified as the “minor” pattern.

For convenience of description, we use a figure matrix M to represent the feature maps. Figure 10 illustrates the features learnt in the first hidden layer of the designed DNN. Each feature image in Figure 10 will maximally activate one of the 100 nodes in the first hidden layer, so the corresponding output in equation (5) is equal to 1. Some features are clearly recognizable, such as in figures $M_{2,1}$, $M_{2,8}$, $M_{3,5}$, and $M_{6,7}$ ($M_{i,j}$ denotes the figure in row i and column j of the matrix), where the “trend” is represented by crossed lines in an “X” shape. The features of $M_{4,1}$, $M_{4,4}$, $M_{4,9}$, and $M_{8,4}$ learns the “normal” pattern, which is dark in the horizontal middle and chaotic at the top and bottom. The “square” is learnt in $M_{4,8}$ and $M_{5,4}$, where the edges are well defined. More than one feature describes the “minor” pattern, such as $M_{4,4}$, $M_{5,8}$, and $M_{9,5}$. The “drift” is implicitly superimposed with other layers of characteristics, as in $M_{6,1}$ and $M_{7,8}$. The features in $M_{4,5}$, $M_{7,6}$, and $M_{9,9}$ generate bright vertical lines to represent outliers. Finally, the “missing” pattern is not found in the features of the first hidden layer of the designed DNN, although this pattern has the most

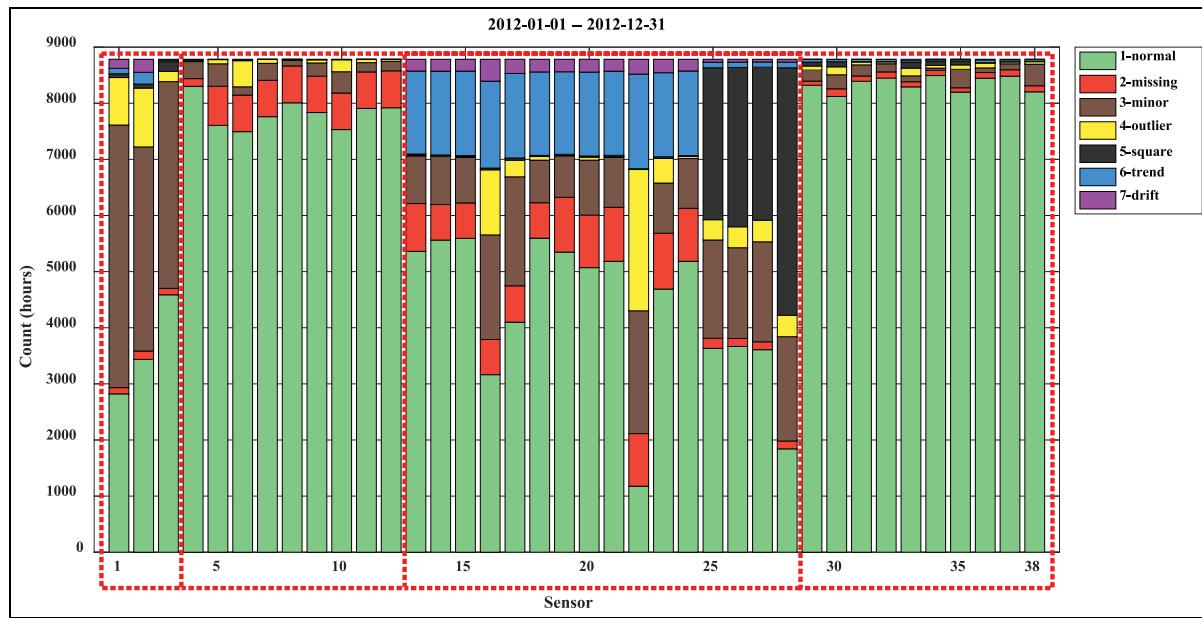


Figure 11. Count results of each anomaly in each channel.

Table 3. Ratio of each pattern in detection results.

Anomaly patterns	Quantity (pieces of data)	Ratio of anomalies (%)	Ratio of total (%)
Normal	233,385	—	69.92
Missing	17,149	17.08	5.14
Minor	36,691	36.54	10.99
Outlier	9943	9.90	2.98
Square	13,734	13.68	4.11
Trend	19,293	19.22	5.78
Drift	3597	3.58	1.08
Total anomalies	100,407	100.00	30.08
Total	333,792	—	100.00

anomalies. Notice that the “missing” pattern is almost blank with pure white pixels, which means there is no feature for learning this pattern in the first hidden layer. However, the “no feature” pattern can be represented by the superposition of multiple features in the learning step for successive layers.

Automatic anomaly detection

To test the data anomaly detection ability of the fully trained DNN, the acceleration data for the entire year were used. The detection took about 6 h on a PC (CPU: Intel i7-3770, RAM: 20 GB, 7200 r/min HDD), which is a low-cost, time-saving solution when compared with human inspection.

Figure 11 shows the counts of each pattern of data anomaly in each channel. Each bar has a total count of 8784 h. The 38 channels in Figure 11 are roughly divided into four groups: channels 1–3, 4–12, 13–28,

and 29–38. Compared with each other, the data quality of group 2 and group 4 is acceptable, with normal data for over 7000 h, and most anomalies are classified as “missing” or “minor”; groups 1 and 3 are worse, with all six data anomaly patterns dominating the data. The count results are presented in Table 3, which demonstrates that 30.08% of data are anomalous. The “minor” pattern is the major anomalous form of data, accounting for 10.99% of the total data.

Figure 12 illustrates the data anomaly distribution in 2012. It is evident that several extensive clusters occur in space and in time; these are marked with numbers in chronological order. Channel 1 to 3 constitute the cluster 1, which mainly consists of the “minor” pattern through the year. Notice that cluster 2 contains channels 13–24. As shown in Figure 4, these channels are all at the southern side of the main span of the bridge, indicating that they are probably in a subnet, and serious errors may occur in this sensor subsystem at that time.

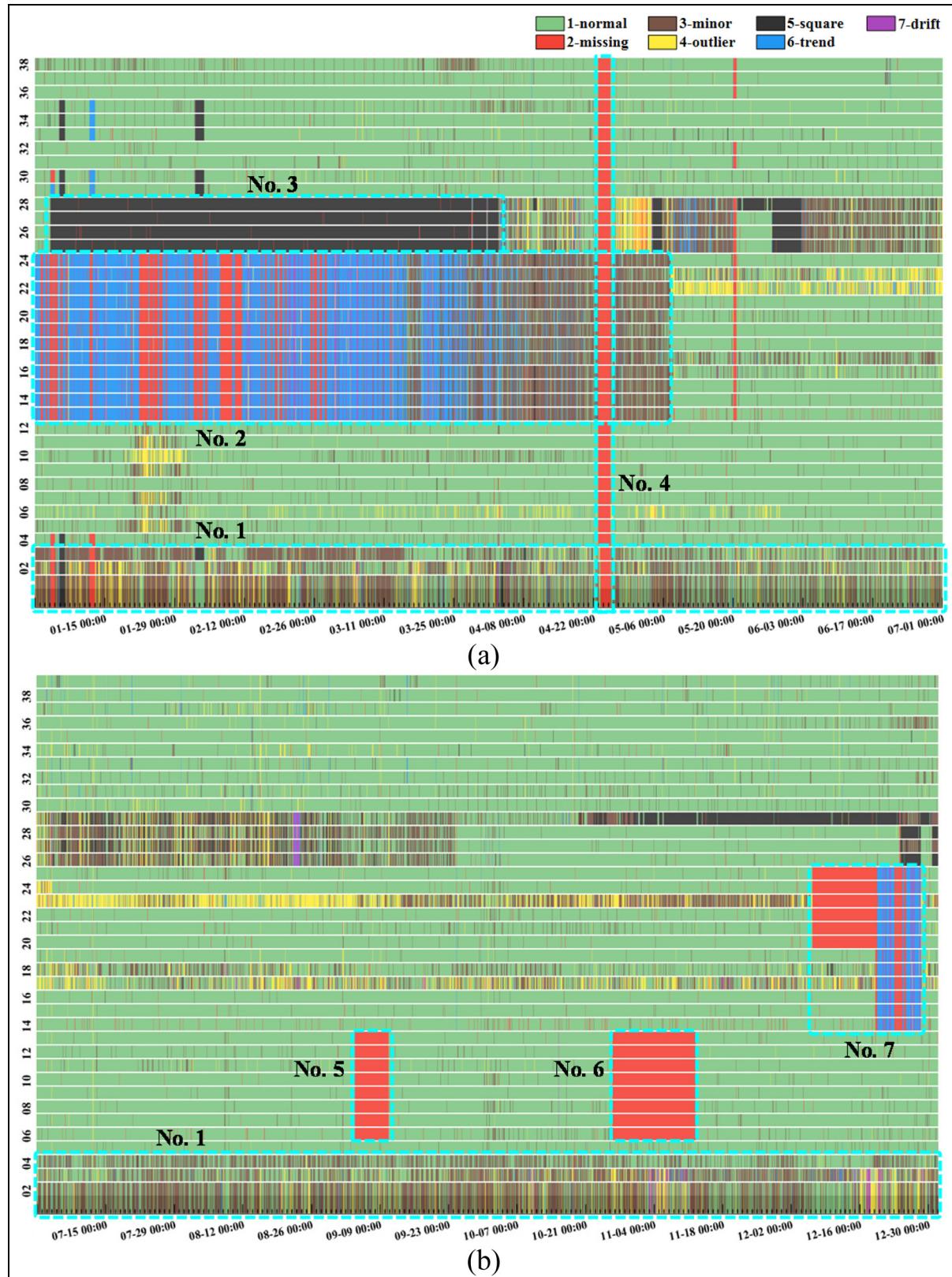


Figure 12. Data anomaly detection in 2012: (a) data anomaly detection from January to June in 2012 and (b) data anomaly detection from July to December in 2012.

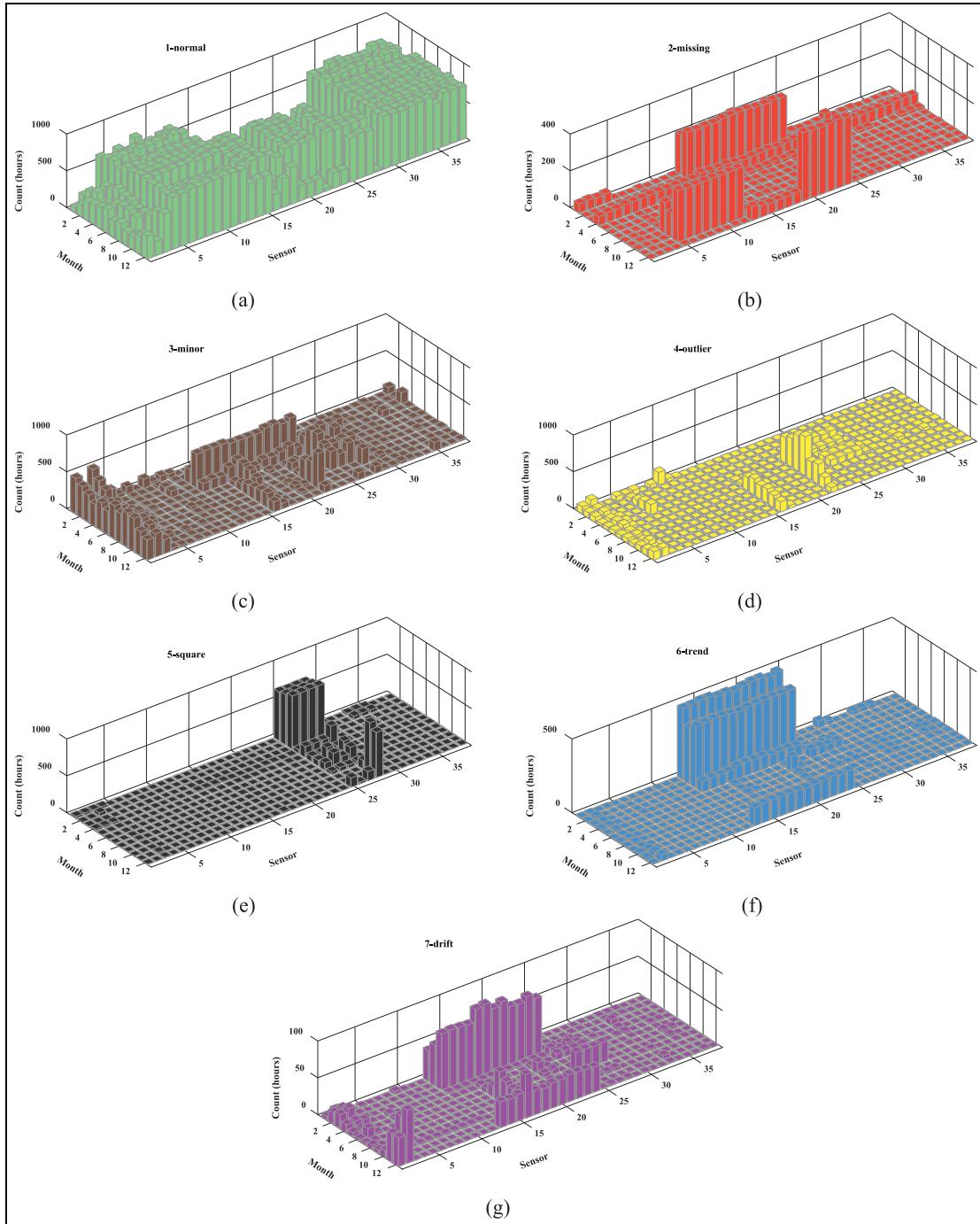
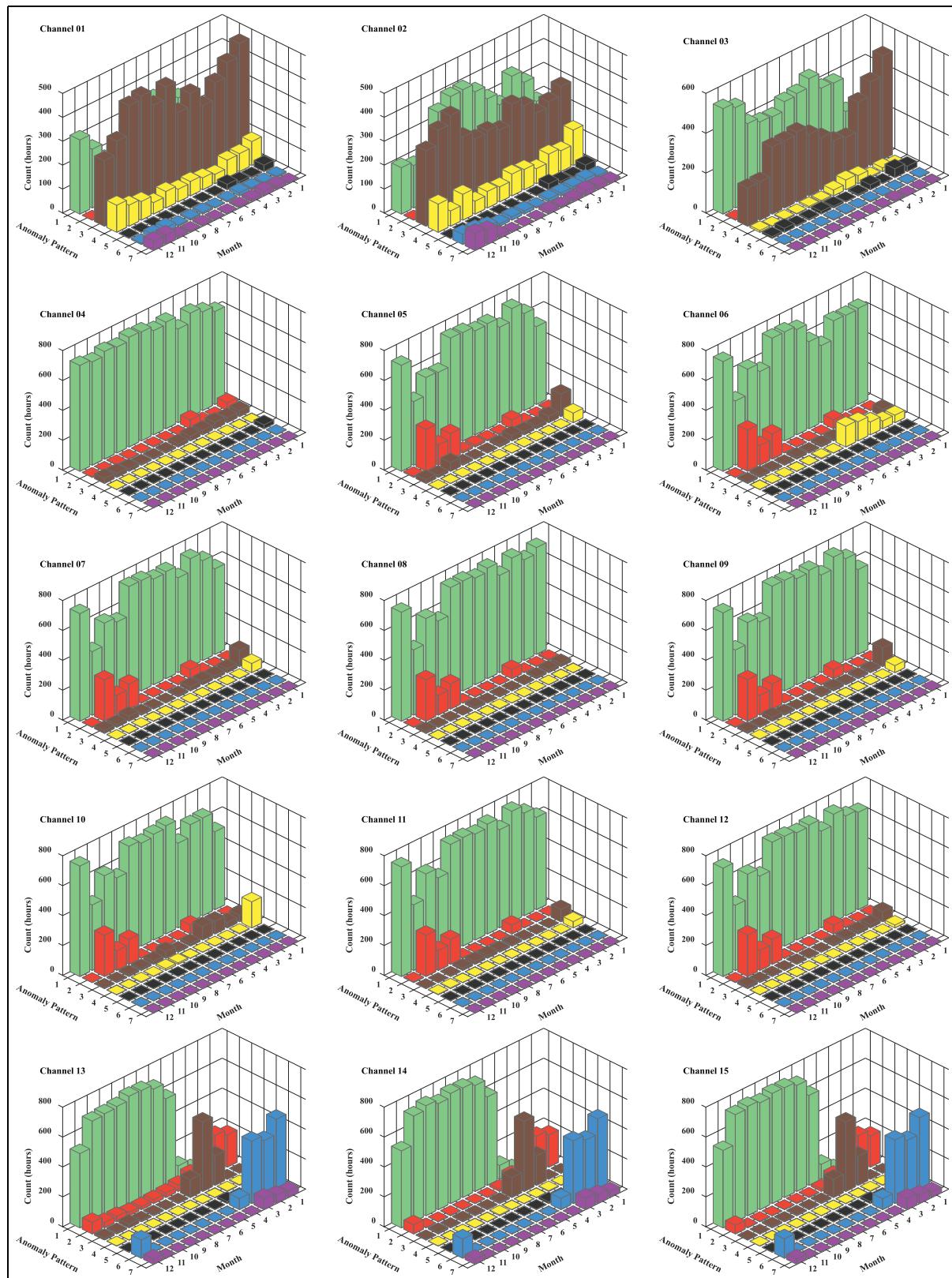


Figure 13. Count results of the distribution of different data anomaly patterns: (a) “normal,” (b) “missing,” (c) “minor,” (d) “outlier,” (e) “square,” (f) “trend,” and (g) “drift.”

Likewise, cluster 3 contains channels 25–28, which are located at the southern side-span of the bridge and simultaneously produced a chunk of the “square” pattern from January to early April. Cluster 4 shows that the whole SHM system failed in late April, because none of the sensors recorded any data. Clusters 5 and 6 consist of the “missing” pattern in channels 5–12; these

channels are located close to the northern side of the main span. Finally, channels 13–24 uniformly malfunction, producing cluster 7, which is similar to cluster 2. In addition to these clusters, there are sporadic anomalies scattered through the year, as shown in Figure 12.

The count results of the data anomaly distribution with data anomaly patterns and sensor channels are

**Figure 14.** Continued

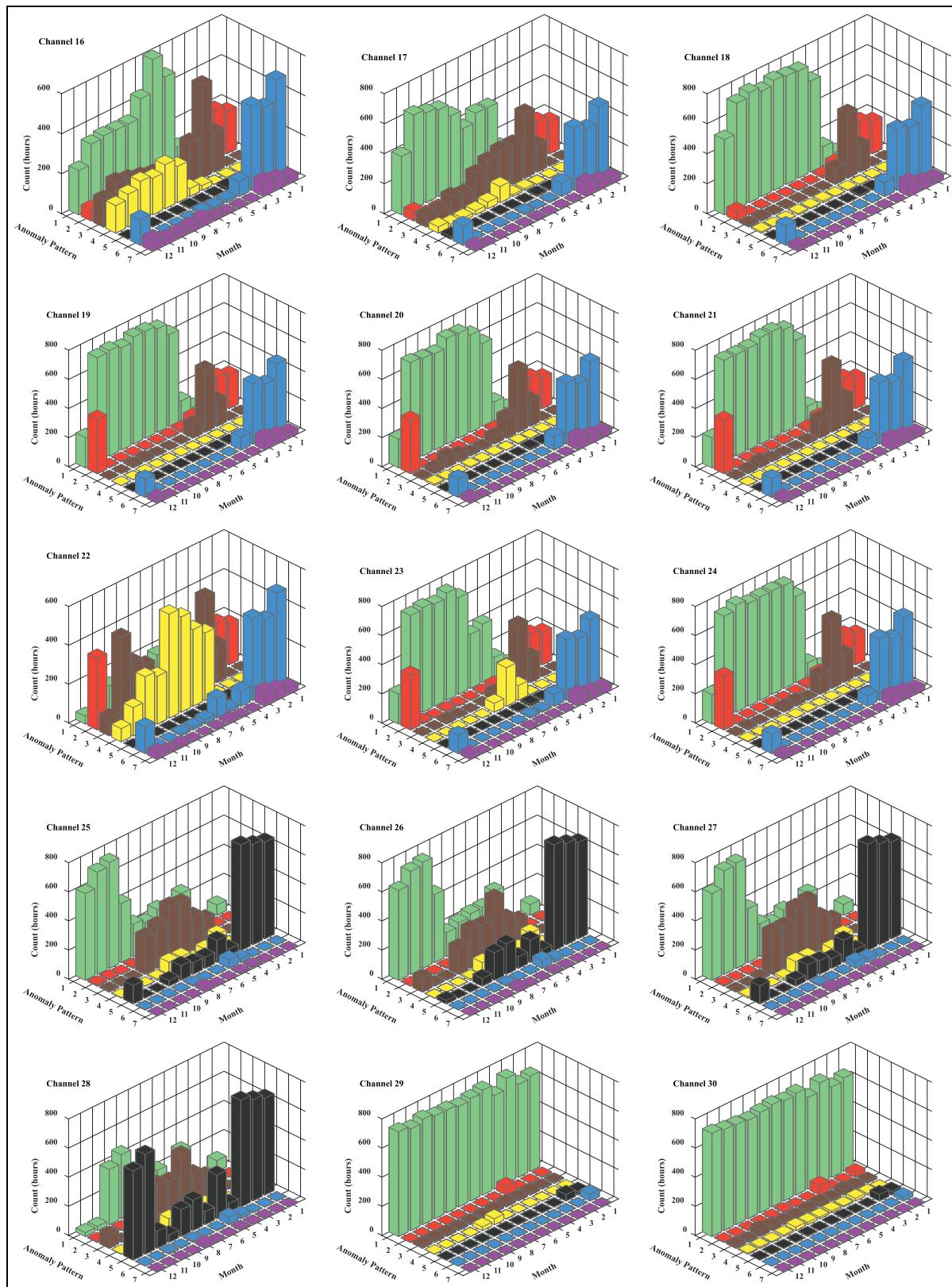


Figure 14. Continued

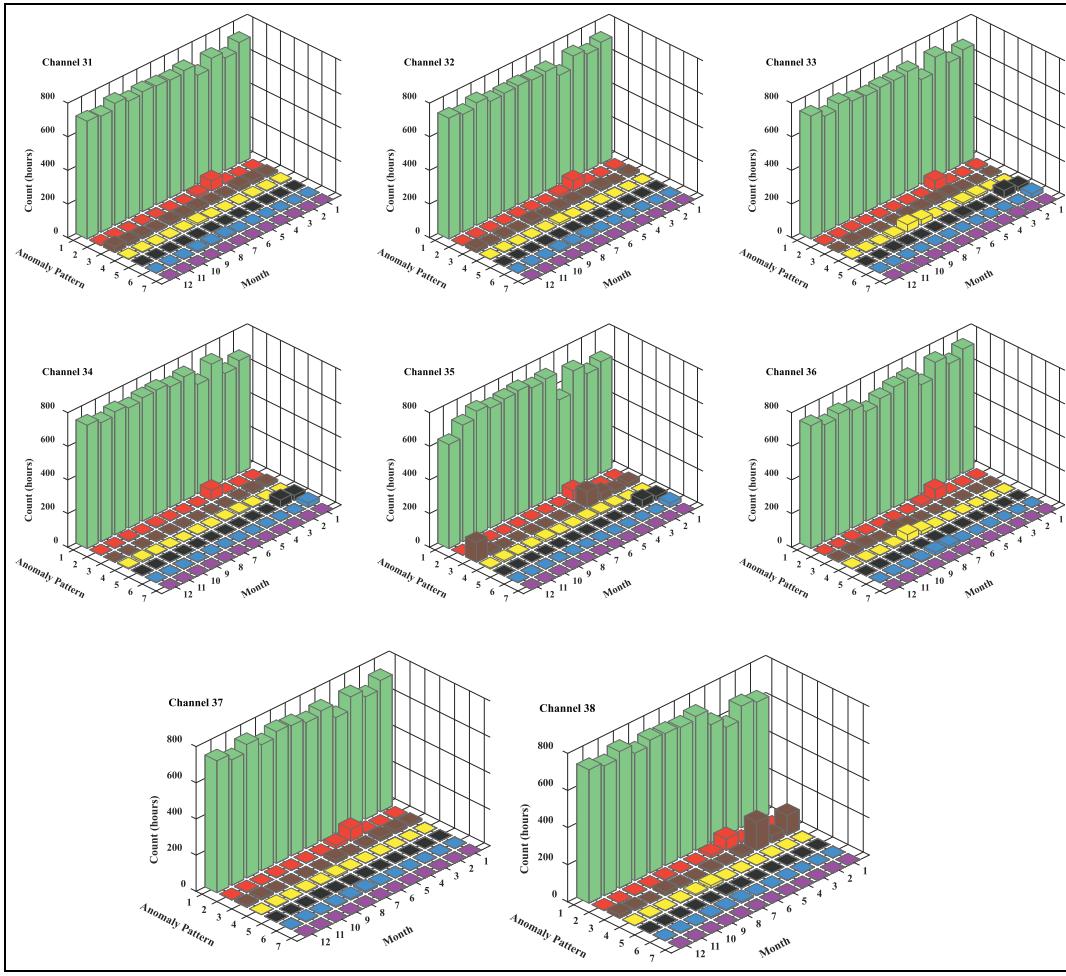


Figure 14. Count results of data anomaly distribution of channels 1–38.

Table 4. Ratio of each pattern in actual data anomaly distribution.

Anomaly patterns	Quantity (pieces of data)	Ratio of anomalies (%)	Ratio of total (%)
Normal	220,013	—	65.91
Missing	17,388	15.28	5.21
Minor	49,786	43.76	14.92
Outlier	9488	8.34	2.84
Square	16,626	14.61	4.98
Trend	18,387	16.16	5.51
Drift	2104	1.85	0.63
Total anomalies	113,779	100.00	34.09
Total	333,792	—	100.00

shown in Figures 13 and 14, respectively. In Figure 13, the distribution of the “missing,”“square,”“trend,” and “drift” data anomaly patterns are relatively clustered, whereas the “minor” and “outlier” patterns are scattered in space-time. Figure 14 shows that channels in the same group (see Figure 11) not only have a similar general data quality but also are similar in terms of anomaly patterns distribution and temporal trend. The

above information provides guidance for further data cleansing and analysis, as well as accurate maintenance of SHM systems. In addition, the proposed method can significantly reduce the number of false alarms caused by anomalies in SHM systems.

To validate the reliability of the proposed method, all image samples in 2012 are manually labeled to make comparison with the detection results of proposed

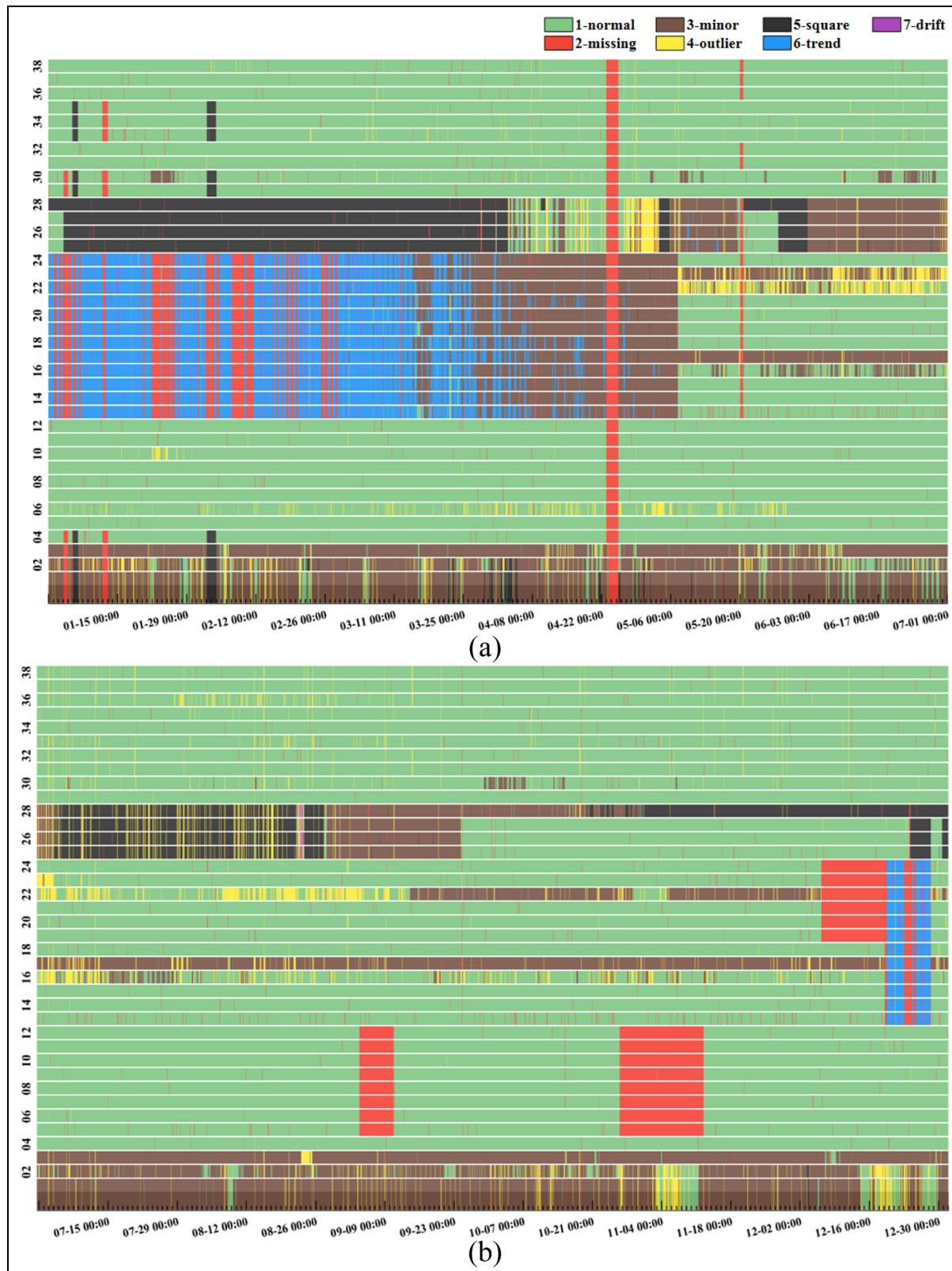


Figure 15. Actual data anomaly distribution in 2012: (a) actual data anomaly distribution from January to June in 2012 and (b) actual data anomaly distribution from July to December in 2012.

	1	2	3	4	5	6	7	
Actual	209952 62.9%	3 0.0%	7550 2.3%	1869 0.6%	164 0.0%	125 0.0%	350 0.1%	95.4% 4.6%
1	14 0.0%	16758 5.0%	3 0.0%	70 0.0%	0 0.0%	439 0.1%	104 0.0%	96.4% 3.6%
2	20977 6.3%	0 0.0%	25792 7.7%	807 0.2%	782 0.2%	1347 0.4%	81 0.0%	51.8% 48.2%
3	867 0.3%	159 0.0%	270 0.1%	7023 2.1%	1 0.0%	737 0.2%	431 0.1%	74.0% 26.0%
4	1535 0.5%	0 0.0%	2250 0.7%	8 0.0%	12787 3.8%	12 0.0%	34 0.0%	76.9% 23.1%
5	39 0.0%	229 0.1%	825 0.2%	120 0.0%	0 0.0%	16300 4.9%	874 0.3%	88.6% 11.4%
6	1 0.0%	0 0.0%	1 0.0%	46 0.0%	0 0.0%	333 0.1%	1723 0.5%	81.9% 18.1%
7	90.0% 10.0%	97.7% 2.3%	70.3% 29.7%	70.6% 29.4%	93.1% 6.9%	84.5% 15.5%	47.9% 52.1%	87.0% 13.0%
	1	2	3	4	5	6	7	Predicted

Figure 16. Confusion matrix between actual data anomalies and the anomaly detection results.

Anomaly patterns: 1—normal, 2—missing, 3—minor, 4—outlier, 5—square, 6—trend, and 7—drift.

method. The actual data anomaly distribution in 2012 is shown in Figure 15, which shows that the results are almost consistent with the detection results in Figure 12 obtained by the proposed method. The count results of actual data anomalies are presented in Table 4, which demonstrates that 34.09% of data are anomalous. Very close to 30.08% of total data anomalies of the detection results as shown in Table 3, the “minor” pattern is the major anomalous form of data, accounting for 14.92% in the total data.

Figure 16 is the confusion matrix for further verifying the reliability of the proposed approach. For the “normal” and “missing” patterns, both the recall and the precision keep a high value over 90%; for the “minor” pattern, 42.13% of the “minor” samples are misclassified as “normal,” and 3.43% of the “normal” samples are misclassified as “minor,” so the recall and the precision are both lower than other patterns; the “outlier” pattern has moderate recall and precision, which is 74.0% and 70.6%, respectively; for the “square” pattern, 76.9% of the actual “square” samples are correctly detected, and 93.1% are TPs in the detection results, which is acceptable; the “trend” pattern has good recall and precision, reaching 88.6% and 84.5%, respectively; due to the small amount of the “drift” pattern, the precision is dragged to 47.9% by other patterns’ misclassified samples. Finally, the total accuracy of 1-year test data is 87.0%, which illustrates the good ability of the proposed method for SHM data anomaly detection.

Conclusion

In this article, we present a computer vision and deep learning-based data anomaly detection method to auto-detect anomalies in SHM systems. By imitating human expert, the SHM time series data are first converted into image, which can be visualized for computer, and then the image vectors of grayscale figures are used as training set of DNN. The DNN is designed and then trained by the greedy layer-wise training technique. The measured acceleration data of a long-span cable-stayed bridge by SHM system are employed to validate the feasibility and accuracy of the designed and trained DNN. The used data in the example contain six patterns of data anomaly, the global accuracy of data anomaly detection results by the designed and trained DNN can achieve 87.0%. The data anomaly distribution and the sensor-wise anomaly count results are obtained, which are useful to the further accurate data cleansing and the maintenance of SHM system. Compared with the manual inspection method, the proposed computer vision and deep learning-based method is much higher efficient.

The proposed method provides a new perspective in SHM data preprocessing, which is essential for automatic real-time monitoring and alarming of SHM systems and data-based off-line long-term performance analysis of structures. While this article focuses on the acceleration data only, it can be also applied to the other types of sensors data. In the future work, more attention should be paid on the unsupervised learning of the representations of anomalies’ images to reduce manual intervention. In addition, the multi-label classification method can be employed for concurrent anomalies in data measured from SHM systems.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by grants from the National Key R&D Program of China (Grant No. 2017YFC1500603), the National Natural Science Foundation of China (Grant Nos 51638007, 51678203, and U1711265), and the research fund of the State Key Laboratory on Safety and Health of In-service Long-span Bridges (Grant No. 2015KF01).

ORCID iD

Hui Li  <https://orcid.org/0000-0001-9198-3951>

References

1. Inman DJ, Farrar CR, Vicente LJ, et al. *Damage prognosis: for aerospace, civil and mechanical systems*. Chichester: John Wiley & Sons, 2005.
2. Sohn H, Farrar CR, Hemez FM, et al. *A review of structural health monitoring literature: 1996–2001*. Los Alamos, NM: Los Alamos National Laboratory, 2004.
3. Li SL, Li H, Liu Y, et al. SMC structural health monitoring benchmark problem using monitored data from an actual cable-stayed bridge. *Struct Control Hlth* 2014; 21: 156–172.
4. Li H and Ou JP. The state of the art in structural health monitoring of cable-stayed bridges. *J Civ Struct Health Monit* 2016; 6: 43–67.
5. Li H, Ou JP, Zhang XG, et al. Research and practice of health monitoring for long-span bridges in the mainland of China. *Smart Struct Syst* 2015; 15: 555–576.
6. Ni YQ, Wong KY and Xia Y. Health checks through landmark bridges to sky-high structures. *Adv Struct Eng* 2011; 14: 103–119.
7. Dalvi A, Norouzi M, Hunt V, et al. Structural health monitoring system of Ironton-Russell Bridge during sub-structure construction. In: *SPIE 9805, proceedings of the smart structures and materials + nondestructive evaluation and health monitoring*, Las Vegas, NV, 20–24 March 2016. Bellingham, WA: SPIE.
8. Huang L, Duan SW, Zhu KM, et al. Study on deformation monitoring of subway station deep foundation construction. *Appl Mech Mater* 2016; 847: 425–430.
9. Park JW, Kim K and Cho YK. Framework of automated construction-safety monitoring using cloud-enabled BIM and BLE mobile tracking sensors. *J Constr Eng M* 2016; 143: 05016019.
10. Worden K and Dulieu-Barton JM. An overview of intelligent fault detection in systems and structures. *Struct Health Monit* 2004; 3: 85–98.
11. Henry M and Clarke D. The self-validating sensor: rationale, definitions and examples. *Control Eng Pract* 1993; 1: 585–610.
12. Feng Z, Wang Q and Shida K. A review of self-validating sensor technology. *Sensor Rev* 2007; 27: 48–56.
13. Shen Z and Wang Q. Failure detection, isolation, and recovery of multifunctional self-validating sensor. *IEEE T Instrum Meas* 2012; 61: 3351–3362.
14. Spencer B, Park J-W, Mechitov K, et al. Next generation wireless smart sensors toward sustainable civil infrastructure. *Procedia Engineer* 2017; 171: 5–13.
15. Lin S, Zhang JB, Zhou G, et al. ATPC: adaptive transmission power control for wireless sensor networks. In: *Proceedings of the 4th international conference on embedded networked sensor systems*, Boulder, CO, 31 October–3 November 2006, pp. 223–236. New York: ACM.
16. Rondinone M, Ansari J, Riihijärvi J, et al. Designing a reliable and stable link quality metric for wireless sensor networks. In: *Proceedings of the workshop on real-world wireless sensor networks*, Glasgow, 1 April 2008, pp. 6–10. New York: ACM.
17. Nagayama T, Moinzadeh P, Mechitov K, et al. Reliable multi-hop communication for structural health monitoring. *Smart Struct Syst* 2010; 6: 481–504.
18. Barnett V and Lewis T. *Outliers in statistical data*. Hoboken, NJ: John Wiley & Sons, 1978.
19. Johnson RA and Wichern DW. *Applied multivariate statistical analysis*. London: Pearson PLC, 2002.
20. Yang WX and Peter WT. Development of an advanced noise reduction method for vibration analysis based on singular value decomposition. *NDT&E Int* 2003; 36: 419–432.
21. Žvokelj M, Zupan S and Prebil I. Non-linear multivariate and multiscale monitoring and signal denoising strategy using Kernel Principal Component Analysis combined with Ensemble Empirical Mode Decomposition method. *Mech Syst Signal Pr* 2011; 25: 2631–2653.
22. Calabrese L, Campanella G and Proverbio E. Noise removal by cluster analysis after long time AE corrosion monitoring of steel reinforcement in concrete. *Constr Build Mater* 2012; 34: 362–371.
23. Katicha SW, Flintsch G, Bryce J, et al. Wavelet denoising of TSD deflection slope measurements for improved pavement structural evaluation. *Comput-Aided Civ Inf* 2014; 29: 399–415.
24. Jiang X, Mahadevan S and Adeli H. Bayesian wavelet packet denoising for structural system identification. *Struct Control Hlth* 2007; 14: 333–356.
25. Yang YC and Nagarajaiah S. Blind denoising of structural vibration responses with outliers via principal component pursuit. *Struct Control Hlth* 2014; 21: 962–978.
26. Kramer MA. Autoassociative neural networks. *Comput Chem Eng* 1992; 16: 313–328.
27. Tamilselvan P and Wang P. Failure diagnosis using deep belief learning based health state classification. *Reliab Eng Syst Safe* 2013; 115: 124–135.
28. Wei SY, Li SL and Li H. Clustering of vehicular cable tension of cable-stayed bridge under normal operation conditions. In: *Proceedings of the 10th international workshop on structural health monitoring (IWSHM 2015)*, Stanford, CA, 1–3 September 2015.
29. Li C, Sanchez R-V, Zurita G, et al. Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. *Neurocomputing* 2015; 168: 119–127.
30. Ye XW, Ni YQ, Wai TT, et al. A vision-based system for dynamic displacement measurement of long-span bridges: algorithm and verification. *Smart Struct Syst* 2013; 12: 363–379.
31. Koch C, Georgieva K, Kasireddy V, et al. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv Eng Inform* 2015; 29: 196–210.
32. Cha Y-J, Choi W and Buyukozturk O. Deep learning-based crack damage detection using convolutional neural network. *Comput-Aided Civ Inf* 2017; 32: 361–378.
33. Yang YC, Dorn C, Mancini T, et al. Blind identification of full-field vibration modes from video measurements with phase-based video motion magnification. *Mech Syst Signal Pr* 2017; 85: 567–590.

34. Yeum CM, Choi J and Dyke SJ. Autonomous image localization for visual inspection of civil infrastructure. *Smart Mater Struct* 2017; 26: 035051.
35. Bengio Y, Lamblin P, Popovici D, et al. Greedy layer-wise training of deep networks. *Adv Neur In* 2007; 19: 153–160.
36. LeCun Y, Bengio Y and Hinton G. Deep learning. *Nature* 2015; 521: 436–444.
37. Hinton GE, Osindero S and Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Comput* 2006; 18: 1527–1554.
38. Møller M. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 1993; 6: 525–533.