Smart Structures and Systems, *Vol. 29, No. 1 (2022) 63-75*
https://doi.org/10.12989/sss.2022.29.1.063

Convolutional neural network-based data anomaly detection considering class imbalance with limited data

Yao Du [a], Ling-fang Li*, Rong-rong Hou [b], Xiao-you Wang [c], Wei Tian [d] and Yong Xia [e]

Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China

(*Received April 9, 2021, Revised May 25, 2021, Accepted June 21, 2021*)

Abstract. The raw data collected by structural health monitoring (SHM) systems may suffer multiple patterns of anomalies, which pose a significant barrier for an automatic and accurate structural condition assessment. Therefore, the detection and classification of these anomalies is an essential pre-processing step for SHM systems. However, the heterogeneous data patterns, scarce anomalous samples and severe class imbalance make data anomaly detection difficult. In this regard, this study proposes a convolutional neural network-based data anomaly detection method. The time and frequency domains data are transferred as images and used as the input of the neural network for training. ResNet18 is adopted as the feature extractor to avoid training with massive labelled data. In addition, the focal loss function is adopted to soften the class imbalance-induced classification bias. The effectiveness of the proposed method is validated using acceleration data collected in a long-span cable-stayed bridge. The proposed approach detects and classifies data anomalies with high accuracy.

Keywords: class imbalance; data anomaly detection; focal loss; limited labelled data; transfer learning

1. Introduction

Structural health monitoring (SHM) systems generate a massive amount of sensing data for various applications such as model verification, damage detection and condition assessment (Chang *et al*. 2003, Li *et al*. 2014, Sun *et al*. 2017, An *et al*. 2019, Hou and Xia 2020, Xia *et al*. 2020). However, in practice, sensor installation, sensor malfunction, data acquisition and transmission, and harsh operational and environmental conditions may cause data anomaly, which may lead to inaccurate model verification and damage identification (Kullaa 2011, Xia *et al*. 2012). Therefore, the abnormal data should be detected and then cleaned or removed before SHM applications. As an SHM system consists of multiple types of sensors, reaching hundreds or even more in quantity, manual data anomaly detection is time-consuming and may be inaccurate. Online SHM requires automatic and efficient data anomaly detection methods.

Conventional data anomaly detection is commonly conducted by comparing the collected raw data with baseline normal data. The raw data is considered abnormal if the difference exceeds a predefined threshold. For example, Yi *et al*. (2016) used a cumulative sum chart to detect small but persistent shifts of GPS data. The chart would show an obvious peak if the shift existed. Hernandez-Garcia and Masri (2014) proposed a principal component analysis (PCA)-based detection method, which was more sensitive to bias and drift anomalies. Rao *et al*. (2015) defined a residual matrix based on PCA and then detected data anomalies by comparing the change in the orthogonality of the residual matrix. Sharifi *et al*. (2010) detected anomalies by calculating the Mahalanobis squared distance of the data's projection onto the noise subspace and validated its effectiveness for bias and drift anomalies. Smarsly and Law (2014) compared the residual norm between the actual and reconstructed data using an artificial neural network (ANN). They found that the bias anomaly could be easily detected, whereas the drift anomaly was detected with a time delay. Li *et al*. (2019) used generalised likelihood ratio and correction coefficient as indicators to detect and classify data anomalies. Fu *et al*. (2019) proposed a distributed similarity test and ANN-based framework to detect and classify data anomalies in a wireless smart sensor network.

The above anomaly detection methods rely heavily on the proper selection of anomaly indicators, which might be problem-dependent and tend to be effective for only a few specific anomaly patterns. In recent years, deep learning (DL)-based methods have been rapidly developed owing to the improvement of computing performance and the availability of large-scale training data. These methods have been applied to a wide range of fields, such as facial recognition, audio recognition, natural language processing and automatic driving (LeCun *et al*. 2015). In the SHM field, the applications of DL techniques are mainly focused on damage detection of numerical structures or scaled models in the laboratory (Lin *et al*. 2017b, Yu *et al*. 2018,

*Corresponding author, Ph.D.,
E-mail: lingfang.li@connect.polyu.hk
[a] Ph.D. Student, E-mail: duyao.du@connect.polyu.hk
[b] Ph.D., E-mail: rongrong.hou@connect.polyu.hk
[c] Ph.D. Candidate, E-mail: xiaoyou.wang@connect.polyu.hk
[d] Ph.D. Candidate, E-mail: cewei.tian@connect.polyu.hk
[e] Professor, E-mail: y.xia@polyu.edu.hk

ISSN: 1738-1584 (Print), 1738-1991 (Online)

Zhang *et al*. 2019), image-based structural surface defect detection and classification (Yeum and Dyke 2015, Cha *et al*. 2017, Zhang *et al*. 2017, Bao *et al*. 2019b, Mohtasham Khani *et al*. 2019, Spencer *et al*. 2019), structural response prediction (Oh *et al*. 2019, Wu and Jahanshahi 2019), and data compression and recovery (Fan *et al*. 2019, 2020, Jeong *et al*. 2019, Lei *et al*. 2020, Ni *et al*. 2020). Recently, DL techniques have also been used for data anomaly detection (Bao and Li 2020). For example, Bao *et al*. (2019a) constructed a deep neural network to classify data anomalies, which converted the time history data into images as the network input and pretrained the network with a stacked autoencoder. Tang *et al*. (2019) proposed a convolutional neural network (CNN)-based method to classify the raw data into seven patterns, which used the visualised time and frequency information as the input. Ni *et al*. (2020) proposed a one-dimensional CNN for binary anomaly detection, which directly took raw data as the input and simply labelled the data as a normal or abnormal pattern. Liu *et al*. (2020) proposed a long short-term memory-based method, in which features were extracted using Tsfresh and Benjamini-Yekutieli procedure and then were used as the input. Mao *et al*. (2020) proposed a combined generative adversarial network and autoencoder-based method to detect anomalies by comparing the latent loss of generated latent vectors between the raw and reconstructed data.

Abnormal data contains several distinctive characteristics, such as heterogeneous anomaly patterns, limited dataset size and extreme class imbalance. Heterogeneity means that data anomalies usually have multiple classes and even new classes that may appear in the future. Compared with the massive normal data, the anomaly is a relatively small group combined with severe inter-class imbalance. All these factors make it difficult to effectively detect and classify data anomalies, either with the traditional methods or aforementioned DL-based methods.

In this regard, this study proposes a new intelligent data anomaly detection method. By leveraging transfer learning (TL), ResNet18, a widely adopted deep CNN that has been pretrained using ImageNet database (Krizhevsky *et al*. 2012, He *et al*. 2016), is assigned as the feature extractor of our model to facilitate the model training without the requirement of large training data or training from scratch. Instead of the commonly adopted cross-entropy (CE) loss for classification tasks, the focal loss is used to focus on model training of hardly classified examples to eliminate the negative effect of class imbalance (Lin *et al*. 2017a). The effectiveness of the proposed method is demonstrated by detecting the abnormal acceleration data of a long-span cable-stayed bridge in China.

2. Framework of the proposed CNN-based method

Fig. 1 shows the overall framework of the proposed CNN-based method. The input module transforms the raw sensor data into grayscale images, which are manually labelled and served as the training dataset of the model. The proposed TL-based CNN is then established. During the training stage, the focal loss function is used to correct the classification bias induced by class imbalance, and the k-fold cross-validation is adopted to make the best use of the available data (Bengio and Grandvalet 2004). After the training, the model could be applied to automatically detect anomalies in the actual SHM data.

2.1 Data visualisation: Input to CNN

CNN is a type of deep feed-forward neural network, which is powerful to process visualised data. Given the experience of manual inspection that data anomalies could be easily classified when visualised with apparent graphic features, this study uses CNN to conduct data anomaly detection. To meet the requirement of automatic detection and extract more graphic information, the raw SHM data are first clipped into hourly segments without overlapping and then presented in both time and frequency domains. The graphic information from both domains is sufficient to distinguish different data patterns. Then, the images from both domains are stitched together as the integrated images and manually labelled. These integrated images are served as the input to the proposed CNN model. In order to reduce the training difficulty, the grayscale images with only one channel are adopted instead of the common RGB images with three channels. As the number of initial input channel is set to 3 in ResNet18, an additional convolution layer is added before ResNet18 module to achieve compatibility.

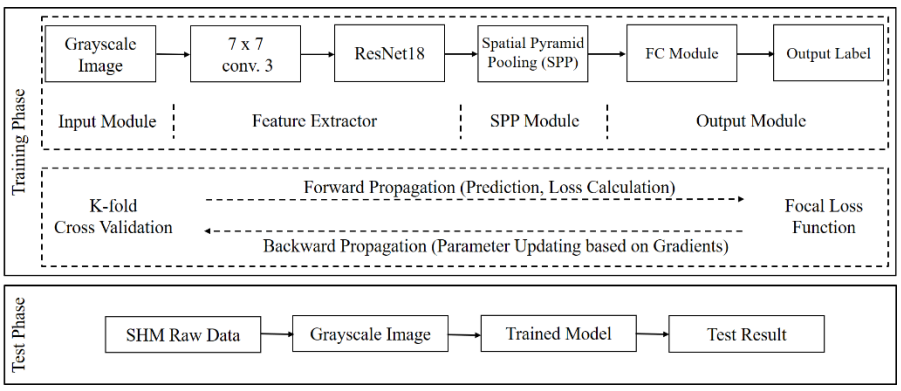


Fig. 1 Framework of the proposed CNN-based method

2.2 Network structure

2.2.1 Basic layers of general CNNs

A general CNN is usually composed of a set of basic functional layers, such as the convolutional layer, non-linear activation layer, batch normalisation (BN) layer, pooling layer, and fully connected (FC) layer. These layers are introduced briefly below.

The convolutional layer is the most representative layer of a CNN. It may have multiple filters as required. Each filter is a feature exactor, which slides through the input and produces a feature map representing certain characteristics of the input. The depth of a filter is the same as the channel number of the input feature maps, and each 2D slide of a filter is called a kernel. The size of a kernel is defined by users, and a rule of thumb is 3 × 3 (or 5 × 5). Feature maps created by different filters serve as the input to the next layer (Rawat and Wang 2017).

The non-linear activation function is to improve the expressivity of the network by adding nonlinearity. One of the most popular activation functions is ReLU (Ramachandran *et al*. 2017), which equals the input value if the input is positive and changes to zero otherwise. This function detects whether a certain feature is presented at a given location and decides whether to activate the corresponding value of the feature maps or not.

BN is commonly used to prevent large distortion of the input data and maintain the training stability by normalising the output value of a certain network layer based on the mean and standard deviation of the minibatch data (Ioffe and Szegedy 2015). It can also serve as a regularisation operator due to the minor variance of statistical distributions between the minibatch data and the entire dataset. This variance can be regarded as a random noise added to the network, which enhances the overall generalisation performance.

The pooling layer can be regarded as a special type of convolutional layer. It can reduce the dimensionality of given feature maps. There are two commonly used pooling layers: max pooling layer and average pooling layer. For each kernel-sized region over a feature map, the former selects the maximum value while the latter calculates the mean value. Max pooling layer is often adopted, as it tends to select the most representative feature of each kernel-sized region.

The structure of an FC layer is similar to that of a multi-layer perceptron, which takes the flatten feature map vector as the input and is often cascaded by a SoftMax classifier for classification tasks.

Above is a brief introduction to the commonly used CNN layers. The items added especially in this study will be introduced in the next section.

2.2.2 ResNet18 as feature extractor

(1) Transfer learning

Machine learning-based data mining models work well only on datasets that have the same feature space and follow the identical data distribution with the training data. The performance degrades when the data distribution changes. Thereby, most models inevitably have to be re-trained from scratch with the requirement of recollecting and labelling large training data, which is expensive and time-consuming. TL, as a subfield of machine learning, is motivated to transfer the knowledge learned previously to a new related problem to facilitate model training. The definition of TL is given as follows: Given a source domain D_S with the corresponding learning task T_S and a target domain D_T with the corresponding learning task T_T, the purpose of TL is to utilise the knowledge hidden in D_S and T_S to facilitate the learning of the target prediction function $F(\cdot)$ in T_T, where $D_S \neq D_T$ or $T_S \neq T_T$ (Pan and Yang 2010).

Model fine-tuning, as a subcategory of transfer learning, leverages a previously trained model for another task to customise the model for the target data. A pre-trained model, which is typically trained on a large-scale dataset, is used as the start model for the target task. Through unfreezing the entire or a part of the model and re-training it on the target data with a small learning rate, the previously trained model can adapt to the new classification task incrementally. In recent years, model fine-tuning techniques have been extensively utilised for visual recognition tasks. Several powerful deep CNNs, such as Inception, GoogleNet, VGG, ResNet and DenseNet (Iandola *et al*. 2014, He *et al*. 2016, Szegedy *et al*. 2017), have been released online with pre-trained parameters to facilitate researchers to conduct specific image-based tasks by leveraging their excellent feature extraction capabilities without the need of massive manually labelled data, which are not easy to acquire in some cases. These CNN models are typically trained on the vast labelled datasets with millions of samples, such as ImageNet, COCO and Open Images (Krizhevsky *et al*. 2012, Kuznetsova *et al*. 2020). The intensive training ensures that the models could learn the representative mapping from the input to the output. Therefore, these deep CNNs usually have stronger generality abilities for other-related image-based tasks and exhibit considerable potentials for TL-based applications.

For the classification task in this study, the model fine-tuning technique can be utilised to facilitate the model training process. The pre-trained ResNet18 is selected as the feature extractor of the proposed model to help train the classifier and maximise the information value with the available labelled data. The details of ResNet18 will be elaborated in the following part.

(2) ResNet18 architecture

The network depth is of great importance to the model performance. A deeper network is expected to enrich multiscale features for better representation learning. However, an unexpected phenomenon is that as the network depth increases, the accuracy asymptotically becomes saturated and then rapidly degrades. This degradation is not caused by overfitting or vanishing/exploding gradients, but the difficulty to optimise the deeper network by current solvers (He *et al*. 2016). Indeed, the shallower network should be in the solution space of the corresponding deeper network, and these two become essentially the same if the extra layers of the deeper network are all replaced by identity mapping, if there are no more optimal solutions. Inspired by this, He *et al*. (2016) proposed a residual

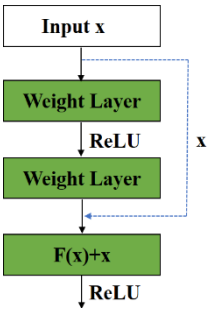


Fig. 2 Example of residual learning (He *et al*. 2016)

learning framework to address the network degradation issue. Given a block consisting of multiple nonlinear layers with the input of **x** and the underlying mapping of $H(x)$, as shown in Fig. 2, the stacked layers are assigned to learn a residual mapping: $F(x) = H(x) - x$ instead of directly fitting $H(x)$. The output becomes $F(x) + x$ before passing through the nonlinear activation function as the input to the next layer.

The experiments conducted by He *et al*. (2016) indicate that it was much easier for the network to learn a residual mapping than the unknown and unreferenced original mapping. The adding operation is accomplished by the shortcut connection plotted in a blue dashed line in Fig. 3. In the residual learning framework, the shortcut connection is a simple identity mapping, which introduces no extra training parameters or computational complexity. Based on residual learning, He *et al*. (2016) further designed ResNet, which was much deeper than other deep CNNs while simultaneously achieving excellent performance. ResNet has several variants with different layers (18, 34, 50, 101, 152) to fit tasks with different complexities. In this study, ResNet18 is selected as the feature extractor to balance the network depth and the information extraction capability. Fig. 3 shows the structure of ResNet18, in which 18 indicates that the initial network has 18 weighted layers (17 convolutional layers and 1 FC layer). The dashed lines in Fig. 3 are shortcut connections. The blue ones indicate that the input and output have the same dimensions and can be directly added up element-wise. The orange lines show another type of shortcut connection with a 1×1 convolution, which is adopted when the input and output have different dimensions. It can increase or decrease the dimensions of the feature maps to match the output.

2.2.3 Spatial pyramid pooling (SPP)

FC layers are usually directly cascaded with the convolutional blocks in conventional CNN classifiers, which account for the majority of network parameters and are prone to causing overfitting (He *et al*. 2016). In the proposed model, an SPP block is inserted after the final convolutional block. SPP adopts a special pooling strategy at different scales on each feature map and ends up with a cascaded feature vector, which simultaneously reduces the number of training parameters and helps extract more robust features for the task. Additionally, the adoption of SPP removes the constraint of fixed-size input. Therefore, images with arbitrary sizes can be served as the input to the proposed network (He *et al*. 2015).

2.3 CNN training strategy

2.3.1 Focal loss function

Class imbalance is a commonly encountered issue in the DL-based classification tasks. Samples of the dominant classes have more chances to be used to train the network and thus will be classified with lower loss. However, the vast amount of these samples still comprises the majority of the entire loss and dominates the network training. Consequently, samples of the rare classes are prone to be classified with poor performance. Rather than the traditional CE loss function, this study uses the focal loss function, which adds a modulating factor to the traditional loss function and weighs down the loss caused by easily classified samples to soften class imbalance-induced classification bias (Lin *et al*. 2017a).

Take a binary classification problem as an example. The traditional CE loss function is defined as

$$CE(p,y)=\begin{cases} -log(p), \text{if } y=1 \\ -log(1-p), \text{if } y=-1 \end{cases} \quad (1)$$

where $CE(p,y)$ is the loss, p denotes the probability of $y = 1$ and $1 - p$ is the probability of $y = -1$. This equation can be rewritten in a simple form as $CE(p,y) = CE(p_t) =$

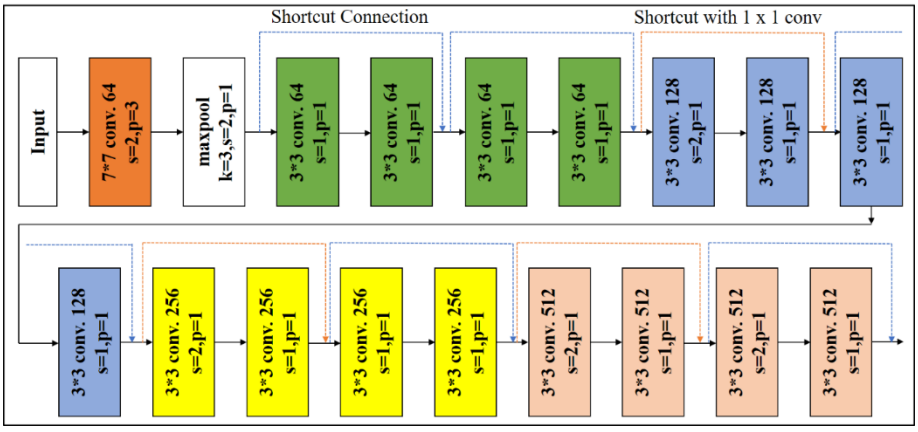


Fig. 3 ResNet18 architecture

$-log(p_t)$ with p_t as the probability of the relevant ground-truth class. A weighting factor α, defined as the inverse of the class percentage in the entire dataset, is usually added to address the class imbalance. The balanced CE loss function becomes

$$CE(p,y) = \alpha CE(p_t) = -\alpha\, log(p_t) \tag{2}$$

This revision considers the sample quantities of different classes and adjusts the loss function at the class level. However, it still fails to consider the difference between easily classified samples and hardly classified samples, which is the hidden nature of the class imbalance. Therefore, another modulating factor $(1-p_t)^r$ is introduced to the balanced loss function as

$$CE(p,y) = -\alpha(1-p_t)^r\, log(p_t). \tag{3}$$

This function is named as focal loss function, where r is a positive hyperparameter (Lin *et al*. 2017a) that can weigh down the loss caused by easily classified samples. A large p_t indicates a good classification result. In this case, the modulating factor $(1-p_t)^r$ is almost 0, and thus the related loss is weighed down. In contrast, if p_t is close to 0, which means the classification performance is poor, then $(1-p_t)^r$ is near 1, and the loss remains unchanged. In this way, the focal loss forces the network to emphasise on the training loss caused by hardly classified samples to improve the overall classification performance.

2.3.2 K-fold cross-validation

The k-fold cross-validation is commonly adopted to select the optimal model and hyperparameters in the case of limited dataset (Rodriguez *et al*. 2009). The training dataset is first separated into k folds. Each fold is assigned as the validation set by turns, leading to k different combinations of training and validation sets for model training. In this study, cascaded training is employed instead of parallel training, and shuffle operation is added to maximise the use of available labelled data and enhance the robustness of the training process. Here, k is set as 4 to balance the fold size and iteration steps. After k epochs, the entire training set is shuffled to disrupt the sample order and thus enhance the variability of the training dataset and prevent overfitting. This shuffle-training-validation takes 15 iterations to achieve the convergence of network parameters, and the total epoch number is 60 in this study. Then, the trained CNN is applied on the test dataset. Fig. 4 shows the entire training procedure.

Table 1 Confusion matrix of a binary classification

Labelled \ Predicted	Predicted positive	Predicted negative
Positive label	True Positive (TP)	False Negative (FN)
Negative label	False Positive (FP)	True Negative (TN)

2.4 Model performance indices

Table 1 displays the confusion matrix of a binary classification.

Based on the confusion matrix, the following indices can be defined.

(1) True positive rate (TPR), or sensitivity

$$TPR = \frac{TP}{TP+FN} \tag{4}$$

(2) False positive rate (FPR), or specificity

$$FPR = \frac{FP}{TN+FP} \tag{5}$$

(3) Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

(4) Precision

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

(5) Recall

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

TPR is the ratio of the correctly predicted positives over the ground-truth positives while FRP is the ratio of the falsely predicted positives over the ground-truth negatives. Accuracy represents the overall correctly predicted samples among all classes, which is not objective when class imbalance exists. For each single class, precision is the ratio of the correctly predicted positives over all predicted positives while recall is the ratio of the falsely predicted positives over the ground truth.

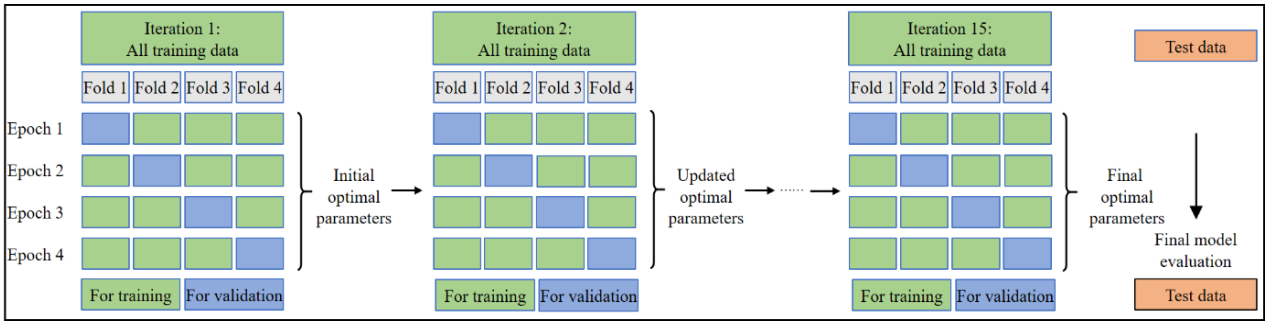


Fig. 4 K-fold cross-validation

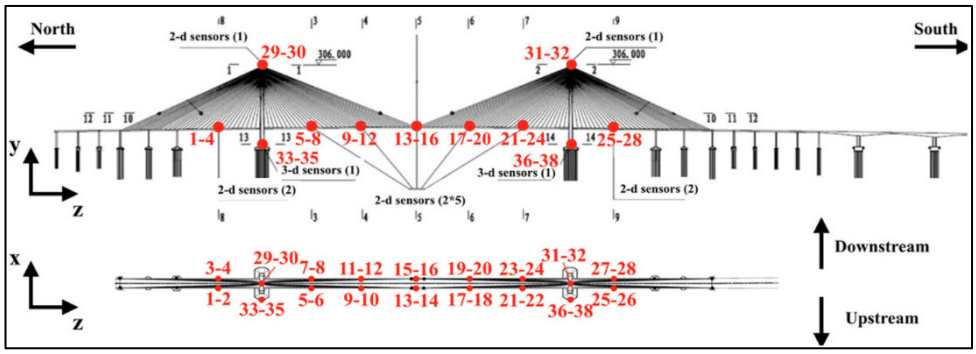
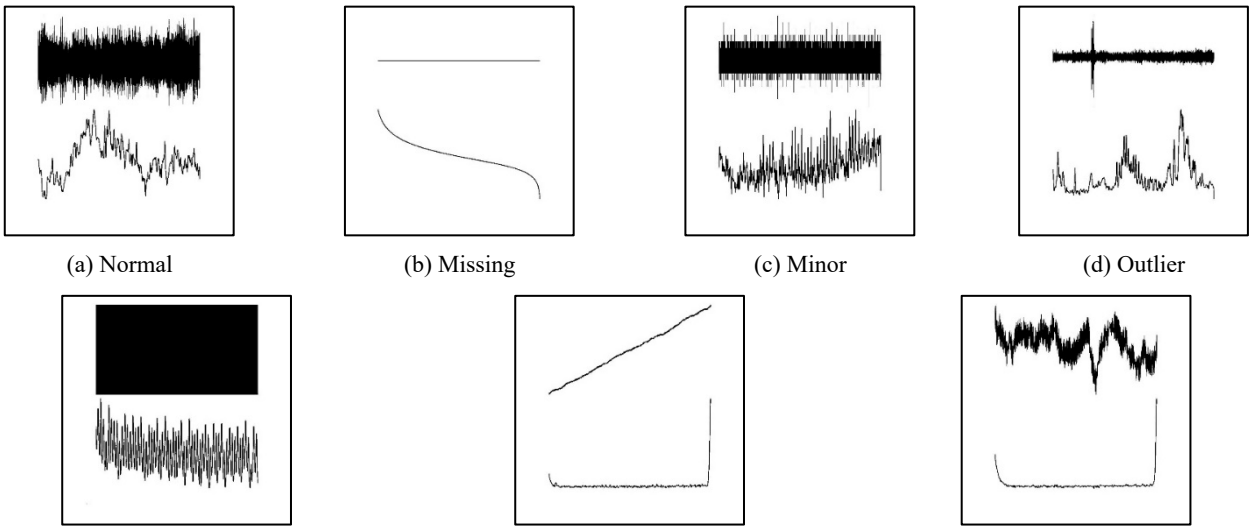


Fig. 5 Accelerometer layout of a long-span bridge (Bao *et al*. 2019a)



(a) Normal (b) Missing (c) Minor (d) Outlier

(e) Square (f) Trend (g) Drift

Fig. 6 Acceleration data patterns

3. A case study

3.1 Brief description of the SHM data

The proposed method is validated using the SHM data of a long-span cable-stayed bridge in China (Bao *et al*. 2019a). The bridge is equipped with multiple types of sensors, including accelerometers, strain gauges, thermometers, anemometers, etc. The proposed method is applied to detect the anomaly of the acceleration data.

Fig. 5 shows the accelerometer layout of the bridge, consisting of 38 channels in total with a sampling frequency of 20 Hz. Acceleration data of all 38 channels collected in two months (2012-01-01 ~ 2012-02-29) are utilised in this study, which have been divided into hourly segments without overlapping and transformed into integrated dual-domain grayscale images as mentioned earlier. There are totally 54,720 segments (38 × 24 × 60) in the dataset.

Fig. 6 shows the data that are manually labelled into seven data patterns, including one normal pattern and six abnormal patterns. For each image, the upper part is the hourly acceleration time history and the lower part is the frequency spectrum. “Normal” indicates that the time response oscillates normally; “Missing” denotes that most of the time series data are lost; “Minor” shows that the signal magnitude is very small or the signal contains noise only; “Outlier” is the case when some data points are far away from the normal data; “Square” means the vibration response is probably out of the sensing range of the sensor; “Trend” shows the changing trend of the time series data; and “Drift” implies that the time series data drifts randomly. These images present the most representative and distinguishable patterns. However, in many cases, the time domain images of “Normal”, “Minor” and “Outlier” are similar and might be mis-labelled; while the frequency domain images of “Trend” and “Drift” are very similar and cannot be distinguished. The inability of single domain images necessitates the use of dual-domain images as the input to the CNN.

3.2 Dataset generation

Table 2 summarises the statistical information of the two-month acceleration data. The data show similar pattern distributions that are significantly imbalanced. “Normal” pattern is the dominant class in both months, accounting for nearly half of the total data, whereas “Outlier” and “Drift” patterns are less than 4%. This class imbalance severely

affects the classification performance of different patterns. Given that manual labelling is a time- and labour-consuming task, only the data in January 2012 are used for classical training, validation and testing to simulate the real limited labelled SHM data condition. The data in February 2012 are used as a large blind dataset to validate the effectiveness of the proposed data anomaly detection method.

The acceleration data in January 2012 are first separated into two parts using average sampling. The first part contains 19,152 samples (21-day data) and is used for training. The second part contains 9,120 samples (10-day data) and is used for the small-scale test. These samples are chosen randomly instead of continuously over time because the anomaly patterns on the 31 days show a certain time-clustering behaviour, and the selection of continuous data for training and testing may cause poor generalisation and learning performance of the network.

3.3 Training process

The training epoch, initial learning rate and batch size are set to 60, 0.0015 and 12. The learning rate is updated to 1/10 of the last value after every 6 epochs to maintain the network stability. The training process is performed on a computer with a CPU of Intel Core i7-8700 @3.20 Ghz, and a GPU of GeForce RTX 2080Ti.

During the training process, the overall validation accuracy of all classes per epoch is employed as the performance index to select the optimal parameters. Fig. 7 shows the validation accuracy and loss with respect to the training epoch. As the epoch proceeds, the accuracy and loss vary gradually and become stable after 8 epochs. The validation accuracy reaches its maximum of 99.54% at the 12^{th} epoch. The corresponding model parameters are saved as the optimal model parameters and tested with the unseen dataset in the next part. The total training takes approximately 8.5 hours.

3.4 Small-scale test

The well-trained model is first tested on the unused 9120 samples in January 2012 during the training process. Fig. 8 shows the confusion matrix of the classification results. The horizontal axis represents the predicted labels, and the vertical axis represents the ground truths of the data patterns. The red blocks in the diagonal area represent the correctly classified samples, while the green blocks refer to the falsely identified samples. As shown in the figure, the precision and recall rates of most data patterns are above 95%, and the overall accuracy is 98.3%, indicating that the proposed model can identify most data patterns with excellent performance. Nevertheless, the identification of "Outlier" is less satisfactory, with the precision of 87.7%

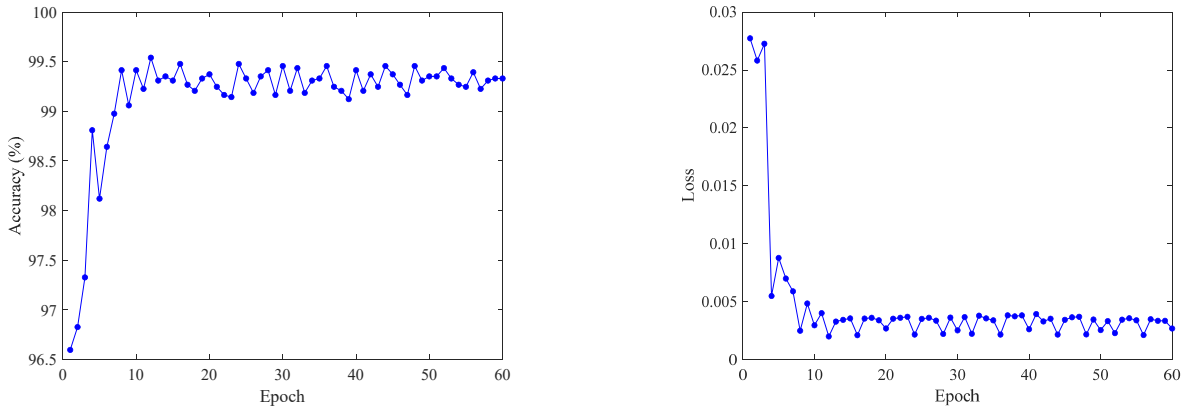
Table 2 Data statistical information

(a) January 2012

Data pattern	Normal	Missing	Minor	Outlier	Square	Trend	Drift
Quantity	13,575	2,942	1,775	527	2,996	5,778	679
Percentage (%)	48.02	10.41	6.28	1.86	10.60	20.44	2.40
Total	28,272/100%						

(b) February 2012

Data pattern	Normal	Missing	Minor	Outlier	Square	Trend	Drift
Quantity	12,897	2,967	1,650	331	3,214	4,558	831
Percentage (%)	48.76	11.22	6.24	1.25	12.15	17.23	3.15
Total	26,448/100%						



(a) Validation accuracy per epoch (b) Validation loss per epoch

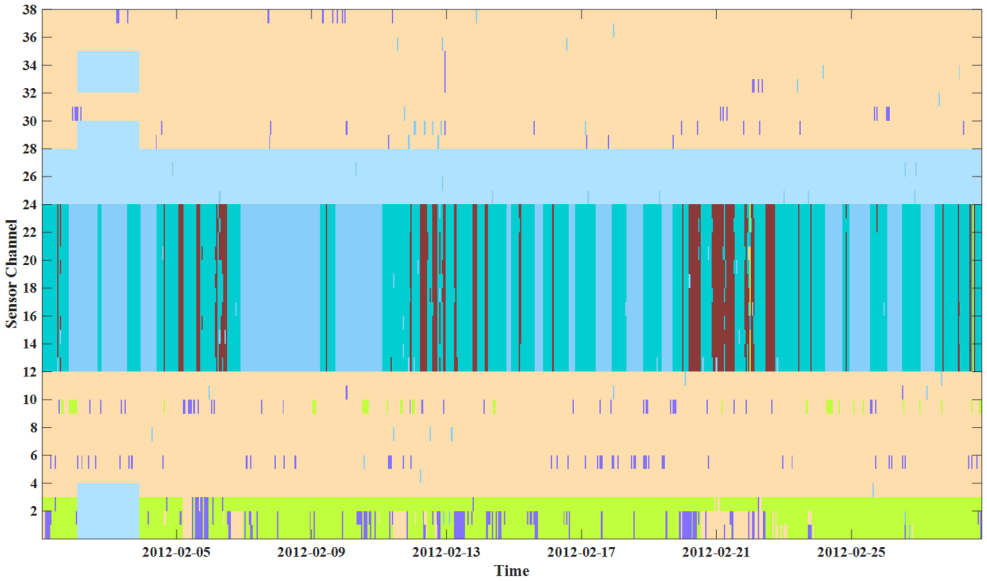
Fig. 7 Validation accuracy and loss in the training process

Labelled \ Predicted	Normal	Missing	Minor	Outlier	Square	Trend	Drift	Recall
Normal	4307	0	23	23	3	0	0	98.9%
Missing	2	1306	0	2	0	0	0	99.7%
Minor	10	0	628	5	0	0	0	97.7%
Outlier	19	2	8	214	0	0	0	88.1%
Square	22	0	0	0	958	0	0	97.8%
Trend	0	0	0	0	0	1407	36	97.5%
Drift	0	0	0	0	0	4	141	97.2%
Precision	98.8%	99.9%	95.3%	87.7%	99.7%	99.7%	80.0%	98.3%

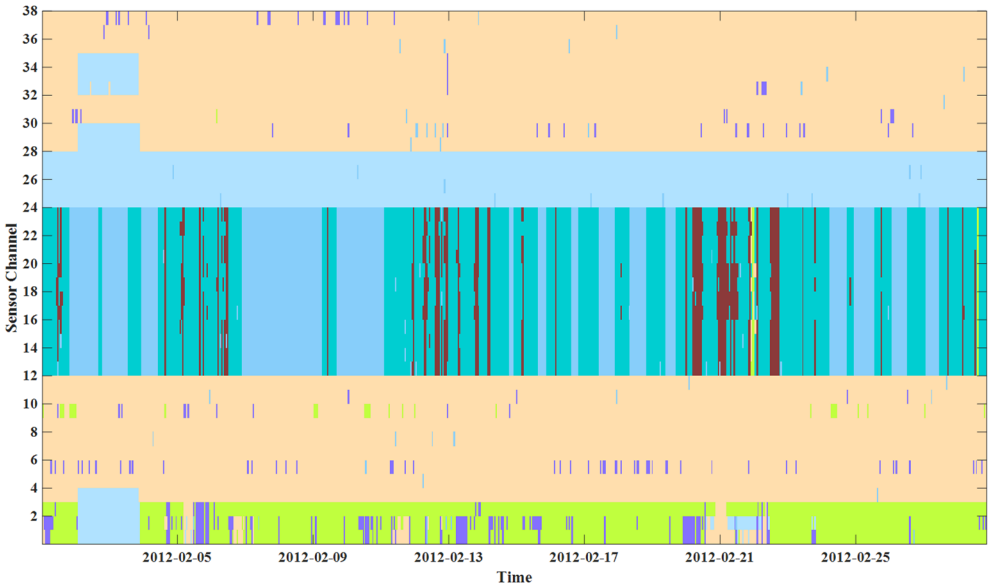
Fig. 8 Confusion matrix of classification results in January 2012

Labelled \ Predicted	Normal	Missing	Minor	Outlier	Square	Trend	Drift	Recall
Normal	12795	0	26	61	7	2	6	99.2%
Missing	3	2962	0	0	2	0	0	99.8%
Minor	21	0	1617	8	0	0	4	98.0%
Outlier	38	2	13	276	2	0	0	83.4%
Square	22	1	0	1	3191	0	0	99.3%
Trend	0	0	5	0	0	4393	160	96.4%
Drift	0	0	0	0	0	86	745	89.7%
Precision	99.4%	99.9%	97.4%	80.0%	99.7%	98.0%	81.4%	98.2%

Fig. 9 Confusion matrix of classification results in February 2012



(a) Anomaly detection results



(b) Actual anomaly labels

Fig. 10 Classification results in February 2012

and recall of 88.1%. This is mainly caused by the misclassification between "Normal" and "Outlier" patterns. To be specific, "Normal" samples account for 76.7% of FP samples of "Outlier", and nearly 1/8 of "Outlier" samples are misclassified as "Normal". The precision of "Drift" pattern is also unsatisfactory, which is 80.0% only. The reason is that a relatively large number of samples of "Trend" pattern are misclassified as "Drift" pattern, which account for 20.3% of the predicted "Drift" samples. Another issue corresponding to the poor performance of "Outlier" and "Drift" patterns is that the samples of these two patterns are limited. In other words, a small number of misclassified samples would have a significant influence on the performance indices of the classes with insufficient samples.

3.5 Large-scale test

In order to further evaluate the anomaly detection performance of the proposed method on the actual long-term SHM data, a blind acceleration dataset collected in February 2012 is utilised, which contains 26448 samples in total. Fig. 9 shows the confusion matrix of the results. Most anomaly patterns can be detected with high precision and recall scores, and the overall accuracy is 98.2%. These results are consistent with the small-scale test in January 2012. The precision and recall scores of "Outlier" and "Drift" patterns are also not satisfactory. The precision of the former drops from 87.7% to 80.0%, and the latter changes from 80.0% to 81.4%. Their recalls also decrease from 88.1% and 97.2% to 83.4% and 89.7%. The dropping is understandable because of the increased test samples. For "Outlier" pattern, samples of "Normal" pattern misclassified as "Outlier" account for 69.1% of FPs, and nearly 11.5% of "Outlier" samples are misclassified as "Normal". For "Drift" pattern, samples of "Trend" pattern misclassified as "Drift" account for 94.1% of FPs, and all FNs of "Drift" pattern belong to "Trend" pattern.

Fig. 10 shows the anomaly detection results and actual ground-truth labels in February 2012 in terms of sensor channel and time for better comparison. These two subfigures are almost similar, indicating that the proposed method has excellent anomaly detection performance. The anomalies in February 2012 are distributed in an explicit cluster manner: "Minor" pattern is mainly distributed in Channels 1–3; "Trend", "Missing" and "Drift" patterns occur extensively in Channels 13–24; and "Square" pattern is the major anomaly in Channels 25–28. Data quality of other sensor channels is satisfactory, with only a few scattered anomalies distributed randomly. Besides, there is a clear correlation between the occurrence of data anomalies and time. Overall, sensor data anomaly shows a clear spatiotemporal cluster characteristic, which can be further studied in the future.

4. Discussions

4.1 The model performance using focal loss function and cross-entropy loss function

The focal loss function is adopted to tackle the class imbalance issue in this study. Table 3 compares the model performance using the focal loss function and cross-entropy loss function. The data in February 2012 is used. The overall accuracies are 98.2% and 98.1%, which are almost identical. The precision and recall scores of the model adopting the focal loss function are slightly better than those adopting the cross-entropy loss function. It is possibly because of the co-existing pattern issue, which will be discussed in detail in Section 4.4.

4.2 The performance of ResNet variants

There are several variants of ResNet. Table 4 compares the model performance adopting ResNet18, ResNet34 and ResNet50 for data in February 2012. The anomaly detection performance does not show a clear improvement with the increased complexity of the network, while some indices even decrease. It is possibly because of the simplicity of the classification task and the limited number of labelled samples compared to the model complexity in this study. Besides, the increase of the network depth increases the training time. Therefore, ResNet18 is appropriate for anomaly detection in this study.

4.3 The effect of different-scale training data on the model performance

Table 5 compares the model performance using different-scale training data, namely, 5-, 10- and 21-day training set. Their overall accuracy is 97.1%, 97.6% and 98.2%, which indicates that the performance improves with the increase in the training data. The precision and recall scores also show a similar trend, especially for "Outlier" and "Drift" patterns. Nevertheless, the increasing training data will inevitably take more training time. Adjusting the image size, batch size and complexity of the backbone model might be considered to optimise the training efficiency in the future.

Table 3 The model performance of focal loss function and cross-entropy loss function

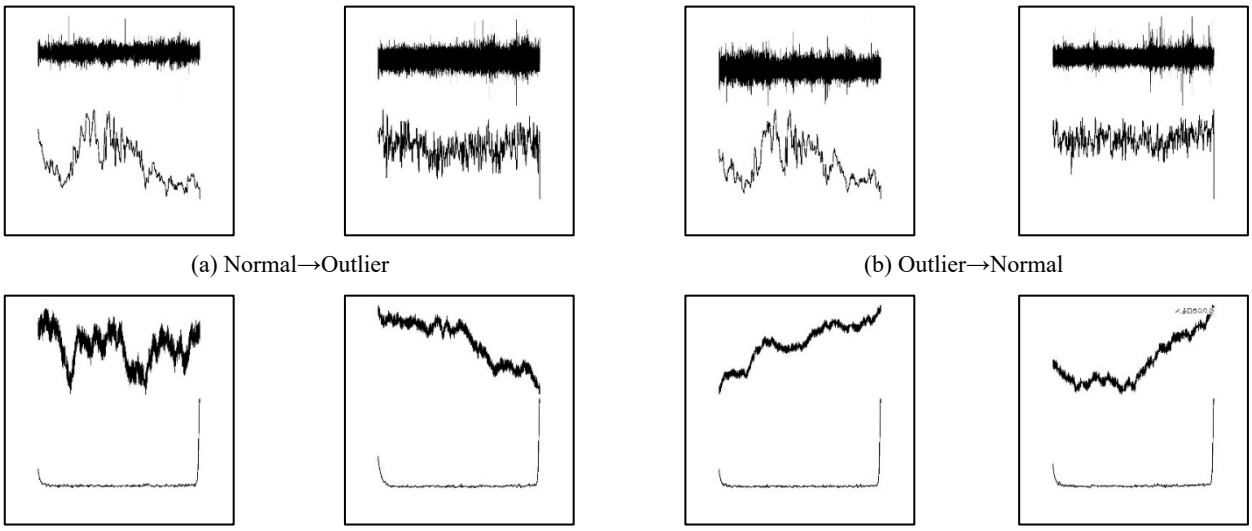
Loss function \ Pattern		Normal	Missing	Minor	Outlier	Square	Trend	Drift
Focal loss function	Precision	99.4%	99.9%	97.4%	80.0%	99.7%	98.0%	81.4%
	Recall	99.2%	99.8%	98.0%	83.4%	99.3%	96.4%	89.7%
Cross-entropy loss function	Precision	99.1%	99.9%	98.4%	80.7%	99.7%	96.5%	85.2%
	Recall	99.4%	99.8%	95.9%	81.9%	99.6%	97.6%	81.0%

Table 4 The performance of ResNet variants

Model	Pattern	Normal	Missing	Minor	Outlier	Square	Trend	Drift
ResNet18	Precision	99.4%	99.9%	97.4%	80.0%	99.7%	98.0%	81.4%
	Recall	99.2%	99.8%	98.0%	83.4%	99.3%	96.4%	89.7%
ResNet34	Precision	99.1%	100%	98.4%	79.4%	99.8%	97.1%	83.7%
	Recall	99.4%	99.8%	95.6%	80.4%	99.6%	97.1%	85.6%
ResNet50	Precision	99.1%	100%	98.3%	79.2%	99.5%	96.8%	82.6%
	Recall	99.3%	99.9%	96.3%	78.3%	99.4%	97.0%	84.1%

Table 5 The model performance using different-scale training data

Dataset	Pattern	Normal	Missing	Minor	Outlier	Square	Trend	Drift
5-day	Precision	99.4%	99.9%	89.8%	67.7%	99.7%	92.8%	90.3%
	Recall	98.2%	99.4%	95.5%	61.9%	99.3%	98.8%	69.7%
10-day	Precision	99.5%	99.8%	94.4%	64.3%	99.6%	95.9%	84.9%
	Recall	98.3%	99.8%	95.9%	84.9%	99.8%	97.7%	78.3%
21-day	Precision	99.4%	99.9%	97.4%	80.0%	99.7%	98.0%	81.4%
	Recall	99.2%	99.8%	98.0%	83.4%	99.3%	96.4%	89.7%



(a) Normal→Outlier (b) Outlier→Normal

(c) Trend→Drift (d) Drift→Trend

Fig. 11 Common misclassification examples

4.4 The possible reasons for the false classification results

The overall accuracy of our proposed method on the small-scale and large-scale tests are 98.3% and 98.2%, which are acceptable in practice. Nevertheless, the precision and recall rates of "Outlier" and "Drift" patterns are relatively low, which are mainly caused by the following four types of misclassified patterns: "Normal" misclassified as "Outlier", "Outlier" misclassified as "Normal", "Trend" misclassified as "Drift" and "Drift" misclassified as "Trend".

Fig. 11 displays typical images of these misclassified samples, with two examples for each misclassified pattern, from which possible reasons for these false classifications can be derived. Figs. 11(a) and (b) present the misclassification between "Normal" and "Outlier" patterns. These four images have similar graphic features but are manually labelled with different patterns. The reason for the ambiguity of labelling is that the spikes in the two figures are "moderate". By contrast, if the spikes exceed the normal range in an extravagant manner, then the relevant sample no doubt belongs to "Outlier" pattern. Take the classification results in February 2012 as an example. After a careful check, among all 61 samples belonging to "Normal" misclassified as "Outlier", 48 samples show the ambiguous characteristic; among all 38 samples belonging to "Outlier" misclassified as "Normal", 71.1% are ambiguous samples.

Similarly, ambiguous labelling also occurs between "Trend" and "Drift" because several samples are "trending with drifting". This coexisting-pattern feature is widespread in the falsely classified samples. In other words, these samples exist more than one pattern. However, all samples are only labelled with a single pattern during the manually labelling stage, resulting in the poor detection performance of our model on those samples.

Two feasible solutions can be investigated in the future to address this problem:

(1) Multi-label classification. Different from multi-class classification, which means that multiple labels represent different classes but each sample should have one label only, multi-label classification indicates that one sample can have multiple labels. For the ambiguous anomaly samples, the multi-label classification might be adopted, such as "Missing and Outlier", "Minor and Outlier" or "Trend and Drift". Multi-label classification is a more complex task, and the inter-relationship between labels may be considered to facilitate the learning process.

(2) Data pre-processing before labelling. The final purpose of anomaly detection is to guarantee the quality of SHM data. Therefore, before manually labelling the raw data, mature data processing tools such as removing outliers or detrending can be carried out first to remove easily identified anomalies. Thus, the occurrence of multi-label samples is expected to be decreased, and a pure multi-class classification task can be conducted to automatically identify "stubborn" anomalies.

5. Conclusions

Data anomaly detection is an essential data pre-processing step for an automatic and reliable SHM. In the actual SHM applications, the majority of collected data are unlabelled, and the anomaly distribution is usually severely imbalanced, which pose a huge challenge to the effective detection of multiple anomalies. In this study, we propose a new anomaly detection method to overcome these constraints. TL is adopted to utilise ResNet18 as the feature extractor to avoid the massive labelling effort. The focal loss function effectively reassigns the training direction and corrects the classification bias. Dual-information image input, SPP and k-fold cross-validation further enhance the feature extraction and representation of the model.

Two sets of acceleration data of a real cable-stayed bridge are used to evaluate the performance of the proposed method. The overall accuracy of the classification results is 98.2% and 97.9%, showing excellent anomaly detection performance. Coexisting pattern is the main reason for the poor detection performance of "Outlier" and "Drift" patterns, for which two possible solutions are suggested. Semi-supervised learning or unsupervised learning can be investigated in the future to better utilise the limited labelled data and massive unlabelled data. Domain adaptation strategies, such as domain adversarial training, may be investigated to expand the deployment capability of the proposed model on detecting anomalous data from other types of sensors or other SHM systems.

Acknowledgments

The authors would like to thank the organisers of the 1st International Project Competition for SHM (IPC-SHM, 2020) for generously providing excellent opportunities during the COVID-19 and invaluable data from an actual structure. Special thanks go to Professor Hui Li and Professor Billie F. Spencer Jr., Co-Chairs of IPC-SHM, 2020. This research is also supported by the Key-Area Research and Development Program of Guangdong Province (Project No. 2019B111106001) and National Key Research and Development Program (Project No. 2019YFB1600700).

References

An, Y.H., Chatzi, E., Sim, S.H., Laflamme, S., Blachowski, B. and Ou, J.P. (2019), "Recent progress and future trends on damage identification methods for bridge structures", *Struct. Control Health Monitor.*, **26**(10). https://doi.org/10.1002/stc.2416

Bao, Y. and Li, H. (2020), "Machine learning paradigm for structural health monitoring", *Struct. Health Monitor.*, **20**(4), 1353-1372. https://doi.org/10.1177/1475921720972416

Bao, Y., Tang, Z., Li, H. and Zhang, Y. (2019a), "Computer vision and deep learning–based data anomaly detection method for structural health monitoring", *Struct. Health Monitor.*, **18**(2), 401-421. https://doi.org/10.1177/1475921718757405

Bao, Y.Q., Chen, Z.C., Wei, S.Y., Xu, Y., Tang, Z.Y. and Li, H. (2019b), "The State of the Art of Data Science and Engineering in Structural Health Monitoring", *Engineering*, **5**(2), 234-242. https://doi.org/10.1016/j.eng.2018.11.027

Bengio, Y. and Grandvalet, Y. (2004), "No unbiased estimator of the variance of k-fold cross-validation", *J. Mach. Learn. Res.*, **5**, 1089-1105.

Cha, Y.J., Choi, W. and Buyukozturk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aided Civ. Infrastruct. Eng.*, **32**(5), 361-378. https://doi.org/10.1111/mice.12263

Chang, P.C., Flatau, A. and Liu, S. (2003), "Health monitoring of civil infrastructure", *Struct. Health Monitor.*, **2**(3), 257-267. https://doi.org/10.1177/1475921703036169

Fan, G., Li, J. and Hao, H. (2019), "Lost data recovery for structural health monitoring based on convolutional neural networks", *Struct. Control Health Monitor.*, **26**(10). https://doi.org/10.1002/stc.2433

Fan, G., Li, J. and Hao, H. (2020), "Dynamic response reconstruction for structural health monitoring using densely connected convolutional networks", *Struct. Health Monitor.*, **20**(4), 1373-1391. https://doi.org/10.1177/1475921720916881

Fu, Y.G., Peng, C., Gomez, F., Narazaki, Y. and Spencer, B.F. (2019), "Sensor fault management techniques for wireless smart sensor networks in structural health monitoring", *Struct. Control Health Monitor.*, **26**(7). https://doi.org/10.1002/stc.2362

He, K., Zhang, X., Ren, S. and Sun, J. (2015), "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", *IEEE Trans. Pattern Anal. Mach. Intell.*, **37**(9), 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hernandez-Garcia, M.R. and Masri, S.F. (2014), "Application of statistical monitoring using latent-variable techniques for detection of faults in sensor networks", *J. Intell. Mater. Syst. Struct.*, **25**(2), 121-136. https://doi.org/10.1177/1045389x13479182

Hou, R. and Xia, Y. (2020), "Review on the new development of vibration-based damage identification for civil engineering structures: 2010–2019", *J. Sound Vib.*, **491**, 115741. https://doi.org/10.1016/j.jsv.2020.115741

Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T. and Keutzer, K. (2014), "Densenet: Implementing efficient convnet descriptor pyramids", arXiv preprint arXiv:1404.1869.

Ioffe, S. and Szegedy, C. (2015), "Batch normalisation: Accelerating deep network training by reducing internal covariate shift", *Proceedings of International Conference on Machine Learning*, pp. 448-456.

Jeong, S., Ferguson, M., Hou, R., Lynch, J.P., Sohn, H. and Law, K.H. (2019), "Sensor data reconstruction using bidirectional recurrent neural network with application to bridge monitoring", *Adv. Eng. Inf.*, **42**, 1009911. https://doi.org/10.1016/j.aei.2019.100991

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Adv. Neural Inform. Process. Syst.*, **25**, 1097-1105.

Kullaa, J. (2011), "Distinguishing between sensor fault, structural damage, and environmental or operational effects in structural health monitoring", *Mech. Syst. Sig. Process.*, **25**(8), 2976-2989. https://doi.org/10.1016/j.ymssp.2011.05.017

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M. and Kolesnikov, A. (2020), "The open images dataset v4", *Int. J. Comput. Vis.*, 1-26. https://doi.org/10.1007/s11263-020-01316-z

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, **521**(7553), 436-444. https://doi.org/10.1038/nature14539

Lei, X., Sun, L. and Xia, Y. (2020), "Lost data reconstruction for structural health monitoring using deep convolutional generative adversarial networks", *Struct. Health Monitor.*, **20**(4), 2069-2087. https://doi.org/10.1177/1475921720959226

Li, J., Hao, H., Xia, Y. and Zhu, H.-P. (2014), "Damage detection of shear connectors in bridge structures with transmissibility in frequency domain", *Int. J. Struct. Stab. Dyn.*, **14**(02), 1350061. https://doi.org/10.1142/s0219455413500612

Li, L.L., Liu, G., Zhang, L.L. and Li, Q. (2019), "Sensor fault detection with generalised likelihood ratio and correlation coefficient for bridge SHM", *J. Sound Vib.*, **442**, 445-458. https://doi.org/10.1016/j.jsv.2018.10.062

Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017a), "Focal loss for dense object detection", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980-2988.

Lin, Y.Z., Nie, Z.H. and Ma, H.W. (2017b), "Structural damage detection with automatic feature-extraction through deep learning", *Comput.-Aided Civ. Infrastruct. Eng.*, **32**(12), 1025-1046. https://doi.org/10.1111/mice.12313

Liu, G., Li, L.L., Zhang, L.L., Li, Q. and Law, S.S. (2020), "Sensor faults classification for SHM systems using deep learning-based method with Tsfresh features", *Smart Mater. Struct.*, **29**(7). https://doi.org/10.1088/1361-665X/ab85a6

Mao, J.X., Wang, H. and Spencer, B.F. (2020), "Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders", *Struct. Health Monitor*, **20**(4), 1609-1626. https://doi.org/10.1177/1475921720924601

Mohtasham Khani, M., Vahidnia, S., Ghasemzadeh, L., Ozturk, Y.E., Yuvalaklioglu, M., Akin, S. and Ure, N.K. (2019), "Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines", *Struct. Health Monitor.*, **19**(5), 1440-1452. https://doi.org/10.1177/1475921719883202

Ni, F.T., Zhang, J. and Noori, M.N. (2020), "Deep learning for data anomaly detection and data compression of a long-span suspension bridge", *Comput.-Aided Civ. Infrastruct. Eng.*, **35**(7), 685-700. https://doi.org/10.1111/mice.12528

Oh, B.K., Glisic, B., Kim, Y. and Park, H.S. (2019), "Convolutional neural network-based wind-induced response estimation model for tall buildings", *Comput.-Aided Civ. Infrastruct. Eng.*, **34**(10), 843-858. https://doi.org/10.1111/mice.12476

Pan, S.J. and Yang, Q.A. (2010), "A Survey on Transfer Learning", *IEEE Trans. Knowl. Data. Eng.*, **22**(10), 1345-1359. https://doi.org/10.1109/Tkde.2009.191

Ramachandran, P., Zoph, B. and Le, Q.V. (2017), "Searching for activation functions", arXiv preprint arXiv:1710.05941.

Rao, A.R.M., Kasireddy, V., Gopalakrishnan, N. and Lakshmi, K. (2015), "Sensor fault detection in structural health monitoring using null subspace–based approach", *J. Intell. Mater. Syst. Struct.*, **26**(2), 172-185. https://doi.org/10.1177/1045389x14522534

Rawat, W. and Wang, Z. (2017), "Deep convolutional neural networks for image classification: A comprehensive review", *Neural Comput.*, **29**(9), 2352-2449. https://doi.org/10.1162/NECO_a_00990

Rodriguez, J.D., Perez, A. and Lozano, J.A. (2009), "Sensitivity analysis of k-fold cross validation in prediction error estimation", *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**(3), 569-575. https://doi.org/10.1109/TPAMI.2009.187

Sharifi, R., Kim, Y. and Langari, R. (2010), "Sensor fault isolation and detection of smart structures", *Smart Mater. Struct.*, **19**(10), 105001. https://doi.org/10.1088/0964-1726/19/10/105001

Smarsly, K. and Law, K.H. (2014), "Decentralised fault detection and isolation in wireless structural health monitoring systems using analytical redundancy", *Adv. Eng. Software*, **73**, 1-10. https://doi.org/10.1016/j.advengsoft.2014.02.005

Spencer, B.F., Hoskere, V. and Narazaki, Y. (2019), "Advances in computer vision-based civil infrastructure inspection and monitoring", *Eng.*, **5**(2), 199-222. https://doi.org/10.1016/j.eng.2018.11.030

Sun, Z., Zou, Z.L. and Zhang, Y.F. (2017), "Utilisation of structural health monitoring in long-span bridges: Case studies", *Struct. Control Health Monitor.*, **24**(10). https://doi.org/10.1002/stc.1979

Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. (2017), "Inception-v4, inception-resnet and the impact of residual connections on learning", *Proceedings of the AAAI Conference on Artificial Intelligence.*

Tang, Z.Y., Chen, Z.C., Bao, Y.Q. and Li, H. (2019), "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring", *Struct. Control Health Monitor.*, **26**(1). https://doi.org/10.1002/stc.2296

Wu, R.T. and Jahanshahi, M.R. (2019), "Deep convolutional neural network for structural dynamic response estimation and system identification", *J. Eng. Mech.*, **145**(1). https://doi.org/10.1061/(asce)em.1943-7889.0001556

Xia, Y., Chen, B., Weng, S., Ni, Y.-Q. and Xu, Y.-L. (2012), "Temperature effect on vibration properties of civil structures: a literature review and case studies", *J. Civil Struct. Health Monitor.*, **2**(1), 29-46. https://doi.org/10.1007/s13349-011-0015-7

Xia, Y., Lei, X., Wang, P., Liu, G. and Sun, L. (2020), "Long-term performance monitoring and assessment of concrete beam bridges using neutral axis indicator", *Struct. Control Health Monitor.*, **27**(12). https://doi.org/10.1002/stc.2637

Yeum, C.M. and Dyke, S.J. (2015), "Vision-Based Automated Crack Detection for Bridge Inspection", *Comput.-Aided Civ.*

Infrastruct. *Eng*., **30**(10), 759-770. https://doi.org/10.1111/mice.12141

Yi, T.H., Li, H.N., Song, G.B. and Guo, Q. (2016), "Detection of shifts in GPS measurements for a long-span bridge using CUSUM chart", *Int*. *J*. *Struct*. *Stab*. *Dyn*., **16**(04), 1640024. https://doi.org/10.1142/S0219455416400241

Yu, Y., Wang, C.Y., Gu, X.Y. and Li, J.C. (2018), "A novel deep learning-based method for damage identification of smart building structures", *Struct*. *Health Monitor*., **18**(1), 143-163. https://doi.org/10.1177/1475921718804132

Zhang, A., Wang, K.C.P., Li, B.X., Yang, E.H., Dai, X.X., Peng, Y., Fei, Y., Liu, Y., Li, J.Q. and Chen, C. (2017), "Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network", *Comput*.*-Aided Civil Infrastruct*. *Eng*., **32**(10), 805-819. https://doi.org/10.1111/mice.12297

Zhang, Y.Q., Miyamori, Y., Mikami, S. and Saito, T. (2019), "Vibration-based structural state identification by a 1-dimensional convolutional neural network", *Comput*.*-Aided Civ*. *Infrastruct*. *Eng*., **34**(9), 822-839. https://doi.org/10.1111/mice.12447