Smart Structures and Systems, *Vol. 29, No. 1 (2022) 181-193*
https://doi.org/10.12989/sss.2022.29.1.181

CNN based data anomaly detection using multi-channel imagery for structural health monitoring

Shaik Althaf V. Shajihan [a], Shuo Wang*, Guanghao Zhai [b] and Billie F. Spencer Jr. [c]

Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

(*Received April 28, 2021, Revised August 14, 2021, Accepted September 24, 2021*)

Abstract. Data-driven structural health monitoring (SHM) of civil infrastructure can be used to continuously assess the state of a structure, allowing preemptive safety measures to be carried out. Long-term monitoring of large-scale civil infrastructure often involves data-collection using a network of numerous sensors of various types. Malfunctioning sensors in the network are common, which can disrupt the condition assessment and even lead to false-negative indications of damage. The overwhelming size of the data collected renders manual approaches to ensure data quality intractable. The task of detecting and classifying an anomaly in the raw data is non-trivial. We propose an approach to automate this task, improving upon the previously developed technique of image-based pre-processing on one-dimensional (1D) data by enriching the features of the neural network input data with multiple channels. In particular, feature engineering is employed to convert the measured time histories into a 3-channel image comprised of (i) the time history, (ii) the spectrogram, and (iii) the probability density function representation of the signal. To demonstrate this approach, a CNN model is designed and trained on a dataset consisting of acceleration records of sensors installed on a long-span bridge, with the goal of fault detection and classification. The effect of imbalance in anomaly patterns observed is studied to better account for unseen test cases. The proposed framework achieves high overall accuracy and recall even when tested on an unseen dataset that is much larger than the samples used for training, offering a viable solution for implementation on full-scale structures where limited labeled-training data is available.

Keywords: convolutional neural network (CNN); data anomaly detection; sensor-fault identification; structural health monitoring

1. Introduction

Structural health monitoring (SHM) provides a useful tool for ensuring the integrity and safety of civil infrastructure, as well as detecting the associated evolution of damage and estimating performance (Ou and Li 2009). Data acquisition is the foundation of the entire SHM system in that the reliability and validity of the collected data is of great significance for the effectiveness of the subsequent processing and assessment (Sohn *et al*. 2002). Huge amounts of data are produced during long-term monitoring, e.g., the SHM system used for the Sutong Bridge in China, which has 785 sensors, produces 2.5 TB of data each year (Tang *et al*. 2019); as a result, multiple types of data anomalies are inevitable, which present significant challenges for practical applications of SHM in the field.

Since the late 1990s, researchers have started to explore sensor data anomaly detection methods, initially focusing on the data structure itself. For example, Friswell and Inman (1999) proposed a sensor validation approach making use of the natural data redundancy for cases when a model of the structure is available. Ibarguengoytia *et al*. (2001) proposed an algorithm utilizing a Bayesian network for the detection of a fault in a set of sensors and can achieve intelligent sensor validation in real time environments. Kerschen *et al*. (2004) proposed a procedure based on principal component analysis (PCA), which can perform detection, isolation, and reconstruction of a faulty sensor. Kullaa (2010) studied sensor fault detection, identification, and correction using the minimum mean square error (MMSE) estimation with the spatial and spatiotemporal correlation between the variables. Hernandez-Garcia and Masri (2014) describe a statistical monitoring approach using latent-variable techniques to detect and identify faulty sensors, and evaluate the performance on a cable-stayed bridge in Los Angeles, California. Yi *et al*. (2016) suggested solving the problem of shift detection in health-monitoring data using the CUSUM chart developed from statistical theory to reduce the risk of false alarms and missed detections in a bridge deformation monitoring system. Yang and Nagarajaiah (2016) harnessed the data structure itself to recover the randomly missing structural vibration responses from the available, incomplete data; they investigated the performance of sparse representation versus low-rank structure in terms of recovery accuracy and computational time under different data missing rates on a few structural vibration response data sets (Nagarajaiah and Yang 2017). Yi *et al*. (2017) provide a comprehensive review of the traditional sensor

*Corresponding author, Ph.D. Student,
E-mail: shuow2@illinois.edu
[a] Ph.D. Student
[b] Ph.D. Student
[c] Newmark Endowed Chair in Civil Engineering

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

validation and data anomaly detection methods. The above data anomaly detection methods mainly focused on the data structure itself and are usually not sufficient to handle multiple types of data anomalies within one framework. Also, the large variations of extracted features from massive SHM data make the data anomaly detection techniques prone to being over-processed or under- processed (Tang *et al*. 2019). These drawbacks indicate that the robustness and efficiency of conventional signal processing techniques are not adequate to handle multiple-types data anomaly detection for massive data, which is an urgent requirement in SHM practice.

Neural networks have recently been proposed to address the problem of data anomaly detection. For example, Fu *et al*. (2019) identified sensor faults in a wireless smart sensor network (WSSN) in a decentralized fashion using Artificial neural networks (ANN). The approach also recovered the faulty data with estimated corrected values for three fault types, and validation was performed on data from the Jindo Bridge, South Korea. These strategies utilized the raw sensor data in the time domain for fault diagnosis. Smarsly and Law (2014) proposed an ANN-model trained based on the time-domain correlation between multiple-sensors in the network and were able to identify drift and bias type faults in the data. Li *et al*. (2019) applied multiple hypothesis test with a generalized likelihood estimator to detect sensor faults and evaluated on the sensor data acquired from a long-span bridge. Oh *et al*. (2020) proposed a structural response recovery method using a convolutional neural network to restore missing strain structural responses when they cannot be collected due to a sensor fault, data loss, or communication errors. Ni *et al*. (2020) proposed a DL-based approach using a CNN and Autoencoder in-parallel for data anomaly detection and compression. Zhang and Lei (2021) model the anomaly detection problem directly as time series classification problem using a CNN. However, these methods relied upon the raw time-series representations, which are computationally intensive, especially for high-sampling rates.

To provide a more scalable and efficient representation of the data, Tang *et al*. (2019) proposed a data anomaly detection method based on transforming the time histories to images. Subsequently, the approach employed a convolutional neural network (CNN) using dual-channel (time and frequency) images of the raw data, improving upon the deep neural network (DNN) based approach proposed by Bao *et al*. (2019). Tang *et al*. (2019) demonstrated their approach using acceleration data for a long-span cable-stayed bridge, identifying six types of anomalies. Acceptable performance with an accuracy of 93.5% on one-year of testing data. Note that anomalies are intrinsically temporal events, with normal data often changing suddenly to anomalous data, and occasionally returning again to normal data. However, the extraction of the frequency-domain channel employed by Tang *et al*. uses a finite time Fourier transform, which loses all temporal information about the signal. Moreover, their network often has difficulty in identifying rare anomalies in the data, due to the inevitable bias in the training data.

This paper proposes an approach that accommodates the temporal nature of data anomalies, while at the same time improving identification of rare anomalies. First, the 1D time series is transformed to an image and decomposed into 3-channels: time, spectrogram, and probability density function (PDF). The images generated from the 1-D input signal are processed and fed into a CNN designed for data anomaly identification. The proposed CNN architecture improves on the efficiency of feature leaning with the use of grouped convolution layers. The spectrogram and PDF channels generalize and encode the change in signal characteristics, which otherwise remain ambiguous in original time-domain representation. To demonstrate the efficacy of the proposed approach, a dataset consisting of a one-month acceleration time record for a long-span cable-stayed bridge in China is examined. The CNN designed employing the proposed approach is trained on 1) a small-sample size balanced subset of the full dataset, 2) imbalanced full dataset, 3) imbalanced full dataset with median frequency balancing. The proposed method achieves high-overall accuracy even when tested on a largely unseen dataset that is larger than the relatively small sample size used for training. Moreover, the results also show a reduced effect of bias with improved recognition of rare fault types in data. Thereby demonstrating the significant potential of the proposed method for improving data-anomaly identification for full-scale structures where labeled-training data is limited.

2. Methodology

This section first discusses the data pre-processing technique, followed by a description of the framework for CNN-based data anomaly identification. The overall workflow for the proposed approach is provided in Fig. 1.

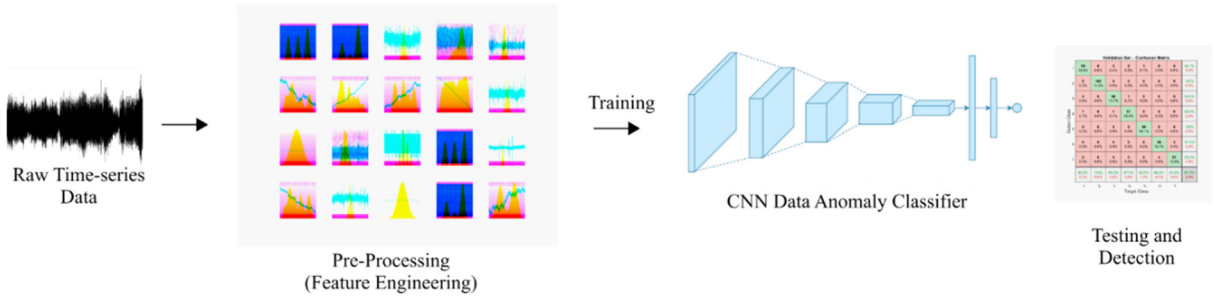


Fig. 1 Workflow of the proposed data anomaly detection framework

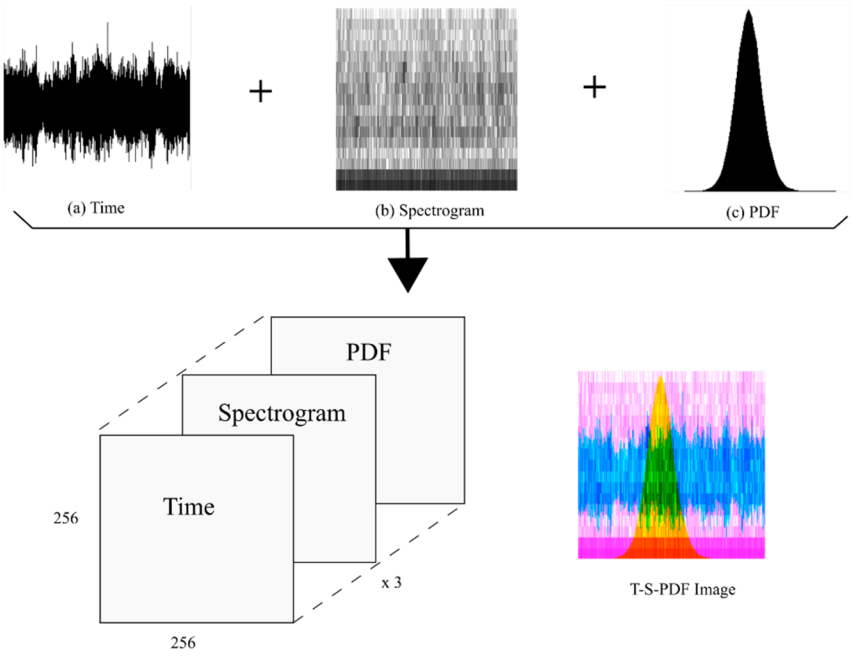


Fig. 2 Proposed feature engineering workflow

2.1 Data pre-processing

Pre-processing of the raw time-series data is required to reduce the dimensionality of the dataset, while retaining key features enabling anomaly detection. The processed data consists of a 3-channel image input comprised of time-domain (T), spectrogram (S), and statistical (PDF) features. A single input image, denoted T-S-PDF image, is produced by stacking the three channels, with cyan representing the time channel, magenta representing the spectrogram channel, and yellow represents the PDF channel, as shown in Fig. 2.

The details of pre-processing steps for each of the 3-channels are described in the following subsections.

2.1.1 Time channel

The time-domain channel is an image derived from the one-hour segments of time-series data for each sensor, with the x-axis being time and the y-axis representing the amplitude of the structural response, as shown in Fig. 2(a). Each image is stored with a resolution of 256 × 256 and a depth of 8-bits, indicating that each pixel takes a value ranging from 0-255. Note that antialiasing filters are applied to the images before being stored. The image dimensionality is chosen by considering the trade-off between features retained by the image and the associated computational cost. The image generation process is automated to create labeled and processed data that will serve as the first input channel to the CNN.

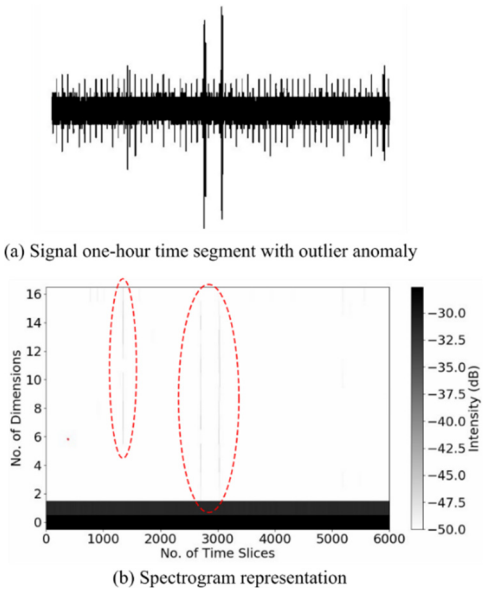
2.1.2 Spectrogram channel

Extracting features from the time-domain signal using a frequency-domain channel was first proposed by Tang *et al*. (2019). Their approach employed the Fourier transform of the entire one-hour time segment of the data, which loses all temporal features within that hour. Because anomalies are intrinsically temporal events, this approach may lack sensitivity to the various anomalies. To address this problem, the proposed approach employs the short-time Fourier transform, termed herein the spectrogram channel, which can accommodate shifts in the frequency and phase of the signal as a function of time and is more appropriate for exploring anomalous changes in features of the data with time.

This study uses a window size of 32 sampling points, combined with a Hamming window and 75% overlap, which yields a low-resolution representation of the changes in signal characteristics as a function of time. To illustrate the capabilities of the spectrogram in capturing temporal variations, consider the segment of time history data given in Fig. 3(a), where several outlier faults are seen. These outliers are clearly identified in the spectrogram, as shown in Fig. 3(b). In contrast, the features of the anomalies could go undetected when the entire one-hour segment of data is considered as in Tang *et al*. (2019).

2.1.3 PDF channel

The probability density function (PDF) of the time-signal amplitude is used as the third channel in the proposed approach. Using only the first two channels described above, fault types can be misclassified, due to inadequate representation of anomaly features. Tang *et al*. (2019) reported that this problem was particularly acute for “drift” and “trend” anomalies. Here a statistics-based PDF channel is introduced to provide additional distinguishing features that can reduce misclassification between various fault types. To this end, the Kernel density estimation (KDE) is used to calculate the PDF of each one-hour time segment, which is then represented as a grayscale image, as illustrated in Fig. 4. Here, in general, the signal with the “trend” type fault is seen to have a multi-modal and flatter PDF, compared to the “drift” type, which has a PDF that is



(a) Signal one-hour time segment with outlier anomaly

(b) Spectrogram representation

Fig. 3 Spectrogram channel representation of time-signal with outlier fault type

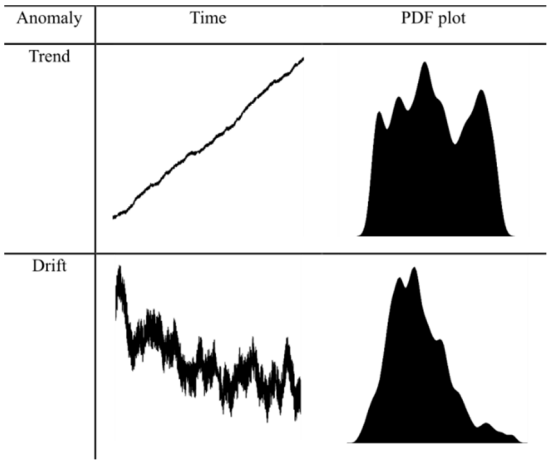


Fig. 4 Typical PDF feature channel for Trend and Drift type faults

closer to unimodal. While this PDF plot may not delineate between all fault types, it does provide additional global features of the 1-D time signal; therefore, the PDF is used as the third channel of the input data.

Finally, all three channels (i.e., time, spectrogram, PDF) are stacked into a single image that forms the input to the 2D CNN. Few illustrative examples of the stacked multi-channel input for the 7 fault classes (see Section 3.1) considered in this study is depicted in Fig. 5.

2.2 Framework of CNN for anomaly identification

The proposed CNN architecture for data anomaly detection and classification using the stacked 3-channel image as input is illustrated in Fig. 6. This section discusses the architecture's key features, training options, and performance metrics used to validate the model.

2.2.1 Key features

The proposed CNN architecture uses grouped convolution layers along with 2D convolution layers to get the output feature maps. The concept of grouped convolution was first introduced in the seminal paper, Alexnet by Krizhevsky *et al*. (2012). This approach has two main advantages: (1) Efficient training, which is achieved by dividing the task into several paths that can be handled separately in parallel on multiple GPUs; (2) More efficient models, as the model parameters decrease with respect to increase in the number of filter groups. Moreover, studies have also indicated that grouped convolution may provide a better model than a normal 2D convolution model due to the influence of the sparse filter relationship (Ioannou 2017). The key layers and activation functions used in the architecture are briefly discussed below:

(1) Leaky rectified linear unit layer (LReLU): The LReLU is an activation function that performs threshold operations and is an improvement over the standard ReLU. The ReLU is not continuously differentiable and can lead to dead neuron problems due to zero values for the gradient when the input is negative. The LReLU overcomes this problem by having a slight negative slope for inputs less than zero; the activation function of LReLU is represented as $max(\alpha x, x)$, where α is the negative slope for the input x.

(2) Cross-channel normalization: In addition to the dropout layer, Cross-channel normalization is used to carry out channel-wise normalization and capture local response before feeding the features into the FCNN with a SoftMax layer for classification.

(3) Batchnorm layer: This layer is used to normalize each input channel across a mini-batch and improve the stability of the model and training speed.

The proposed CNN architecture focuses on learning and extracting generalizable distinguishing features from the 3-channel input image for data anomaly identification. The use of grouped convolution layers in the proposed CNN architecture improves the efficiency of feature learning Note that the stability and efficiency of the network is dependent on the options used for training the network, as discussed in the next subsection.

2.2.2 Training options

Training options such as learning rate and mini-batch size are carefully designed so that training is done in a stable, valid, and efficient way. Here, the Adaptive moment estimation (ADAM) optimizer is used for training the CNN model. Based on trial-and-error, the learning rate is initialized to be 0.0002, which determines the step size at each iteration, while moving toward the minimum of the loss function. The trade-off here is that, if the learning rate is too low, the training will take an excessive amount of time, while if the learning rate is too high, the training is likely to jump over the optimal solution.

To address this issue, the learning rate schedule is set to

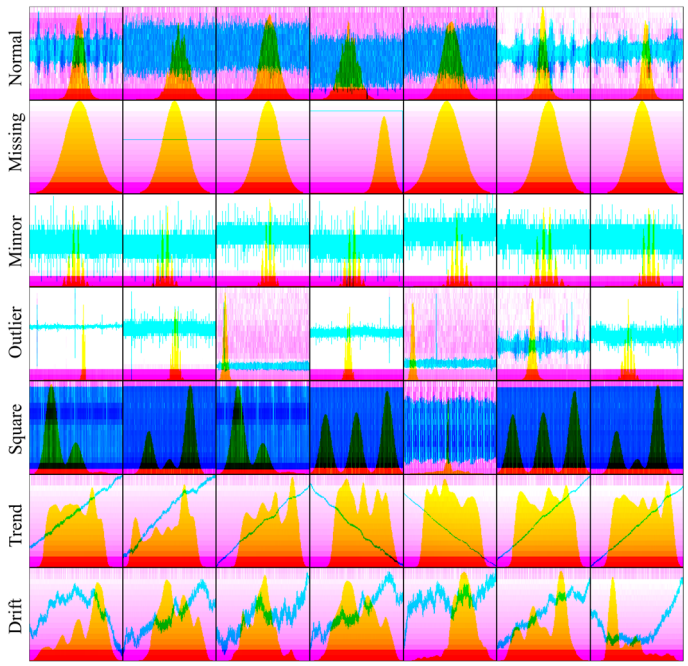


Fig. 5 Examples of stacked multi-channel input for 7 fault classes

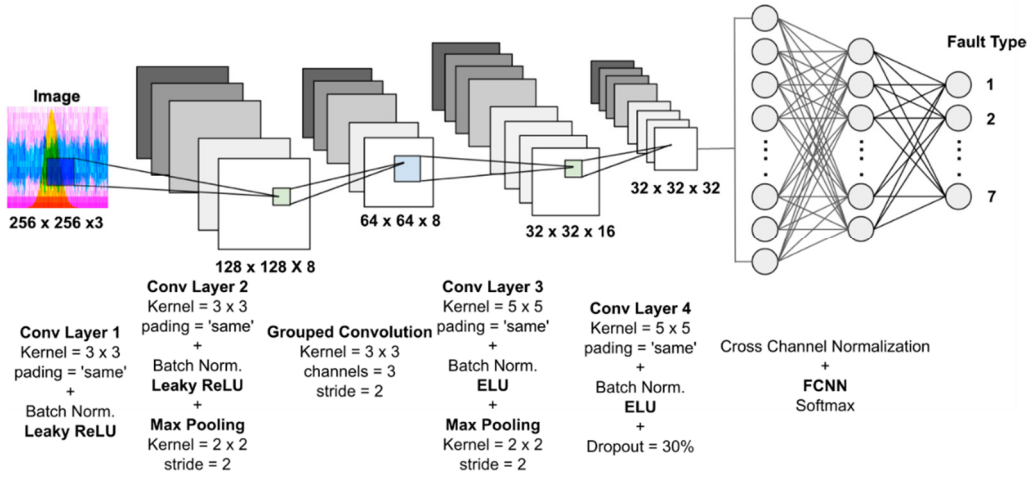


Fig. 6 Designed CNN architecture for data anomaly identification

be piecewise, i.e., the learning rate will be multiplied by a factor every time a certain number of epochs have passed, so that the global learning rate can be lowered during training. This piecewise learning rate design will let the training start from a reasonably large step size and be more and more cautious as the optimum is approached, thus reducing calculation time.

If the mini-batch size does not evenly divide the number of training samples, then the training data that does not fit into the final complete mini-batch for each epoch will be discarded. For this reason, the training data is shuffled before each training epoch to avoid discarding the same data in every epoch; the validation data will also be shuffled before each validation. The validation frequency, i.e., the number of iterations between evaluations of the validation metrics, is chosen to be 10. Based on trial and error, the mini-batch size for each training iteration in this study is set to be 38. Training-progress is assessed by monitoring the training accuracy and loss, as well as the validation accuracy and loss, during the training process, enabling real-time detection of problems such as overfitting.

2.2.3 CNN performance metrics

Statistical performance metrics are beneficial for the effective evaluation of CNN predictions. This research uses the confusion matrix, depicted in Fig. 7, which allows visualization of the performance of a supervised learning algorithm. The precision of the positive class is defined as the ratio of true positive divided by the sum of true positive and false positive; The recall of the positive class is defined as the ratio of true positive, divided by the sum of true positive and false negative; Accuracy is a term representing the overall performance, indicating how often is the classifier correct, and is defined as the ratio of correct

		Actual Condition	
		Actual Condition Positive	Actual Condition Negative
Predicted Condition	Predicted Condition Positive	**True Positive (TP)**	False Positive (FP)
	Predicted Condition Negative	False Negative (FN)	**True Negative (TN)**

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{Recall} = \frac{TP}{TP+FN} \quad \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Fig. 7 Confusion matrix and CNN performance metrics

predictions (i.e., true positive and true negative) to all predictions. The confusion matrix provides a concise assessment of the efficacy of the proposed classification model.

2.2.4 Data imbalance handling

Canonical machine learning algorithms assume that each class in the data is represented similarly (Krawczyk 2016); however, in many applications, imbalanced data, characterized as having more data from certain classes than others, is very common. The resulting bias becomes a challenge for machine learning algorithms, as classification rules that predict the small classes tend to be undiscovered or ignored (Im *et al*. 2020). Consequently, data belonging to the small classes will be misclassified more often than those belonging to the prevalent classes.

A straightforward method to deal with data imbalance is to manually generate a balanced dataset by deleting samples in the overrepresented classes; however, this approach will essentially reduce the amount of data available for training.

Another option is to apply median frequency balancing (Kampffmeyer *et al*. 2016) in the classification layer. The classification layer computes the cross-entropy loss for classification tasks with mutually exclusive classes. Median frequency balancing transforms the classification task into a weighted classification task, where a class weight inversely proportional to the size of each class is applied in the cross-entropy loss function. Specifically, the weight of each class is defined as the ratio of the median of class frequencies divided by the frequency of this class. The modified loss function is given by

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} w_c l_c^{(n)} ln(\hat{p}_c^{(n)}) \quad (1)$$

N is the number of samples in a mini-batch, C is the set of all classes c, f_c is the weight for class c, f_c is the frequency of class $c$computed on the entire training set, $\hat{p}_c^{(n)}$ is the softmax probability of sample n being in class c, and $l_c^{(n)}$ is the one-hot ground truth label of sample n for class c. The above modification implies that larger classes in the training set have a weight smaller than 1, and smaller classes have a weight larger than 1, thus reducing imbalance and associated data bias. Both manually balancing of the dataset and using median frequency balancing will be explored in the following case study.

3. Case study – Long-span bridge

The performance of the proposed methodology for data anomaly identification is demonstrated through a case study using sensor data obtained from a long-span bridge in China (Tang *et al*. 2019). This section discusses the processing of the dataset and the training, validation, and testing results for the CNN.

3.1 Data preparation

In this study, the dataset used contains one-month of acceleration measurements for 38 sensors installed on a long-span cable-stayed bridge in China. The details of the locations and installation directions of sensors are illustrated in Fig. 8. The dataset contains 7 patterns, including the normal acceleration signal and six types of data anomalies (i.e., Normal, Missing, Minor, Outlier, Square, Trend, and Drift). The characteristic of each pattern and the percentage of the dataset containing the respective patterns is given in Table 1. Note that Normal data accounts for 48% of the total dataset, while rare anomaly types, such as Outlier and Drift, account for less than 3% each. Therefore, data imbalance-handling strategies mentioned in Section 2 will be beneficial for the training of this highly imbalanced dataset. The characteristic of the seven data

Table 1 Fault type and count in the one-month dataset (Bao *et al*. 2021)

Fault type	Class	Description	Count	Percentage (%)
Normal	1	The time response is normal oscillation curve; frequency response is peak-like (may differ between bridges)	13575	48.02
Missing	2	Most/all the time response is missing, which makes the time and frequency response zero	2942	10.41
Minor	3	Relative to normal sensor data, the amplitude is very small in the time domain	1775	6.28
Outlier	4	One or more outliers appear in the time response	527	1.86
Square	5	The time response is like a square wave	2996	10.60
Trend	6	The data has an obvious trend in the time domain and has an obvious peak value in the frequency domain	5778	20.44
Drift	7	The vibration response is non-stationary, with random drift	679	2.40

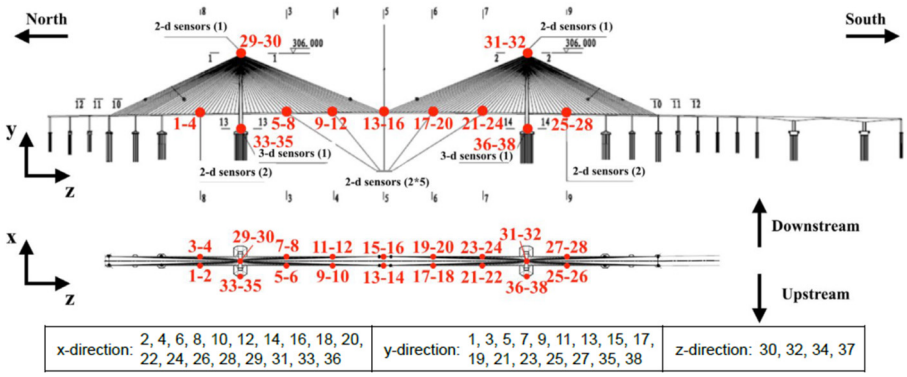


Fig. 8 The bridge and placement of accelerometers on the deck and towers (Tang *et al*. 2019)

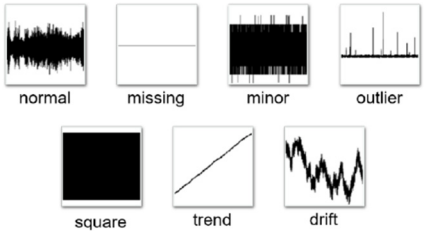


Fig. 9 Example for each data pattern (duration of each time series: 3600s) (Tang *et al*. 2019)

patterns can be seen from the time-domain representation of the acceleration record shown in Fig. 9. The one-month acceleration record of each sensor is split into 744 one-hour-duration segments. The ground truth label of the data pattern for each segment is manually generated by the committee of IPC-SHM 2020 and provided to competition participants (Bao *et al*. 2021).

Time, spectrogram, and PDF channels are generated for each one-hour-duration segment using the proposed data pre-processing method, described in Section 2. Next, a 3-channel image is generated by stacking the time, spectrogram, and PDF channels together. Subsequently, an image datastore is created by sorting the 28,272 (744 one-hour-duration segments for each of the 38 sensors) 3-channel images into seven directories according to their corresponding data pattern labels. Thereafter, the image dataset is split into the training set, validation set, and test set at a specified ratio, e.g., 70%: 20%: 10%, applied identically to each labeled category. The developed datastore is then fed into the designed CNN model for training and evaluation.

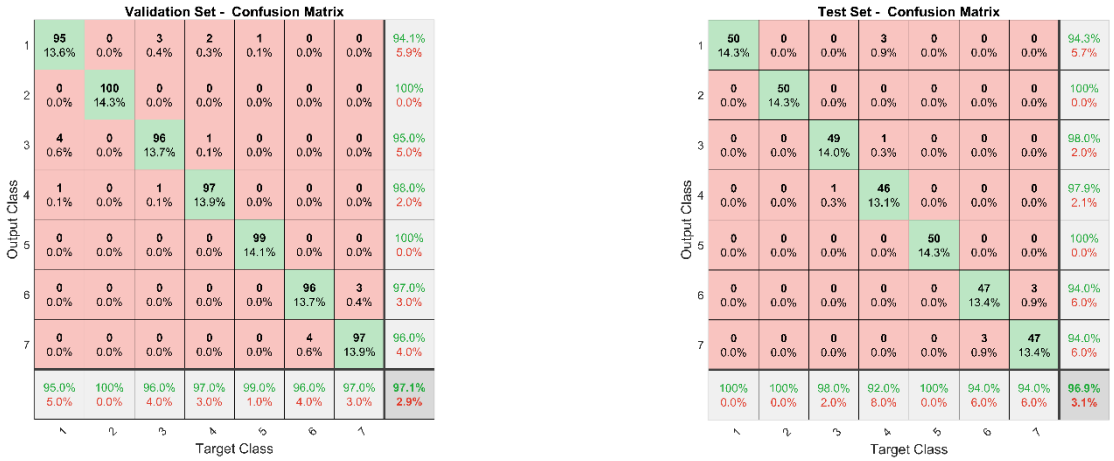
3.2 CNN training and validation

The designed CNN model for data anomaly identification is trained following the training options introduced in section 2.2.2 and is evaluated using the performance metrics described in section 2.2.3. To illustrate the efficacy of median frequency balancing on alleviating imbalance and associated data bias, the following subsections consider the dataset as three separate cases: 1) perfectly balanced dataset, 2) imbalanced full-dataset, and 3) imbalanced full-dataset with median frequency balancing.

3.2.1 Perfectly balanced dataset

In this case, a perfectly balanced dataset is created by truncating the original one-month dataset based on the fault type with the minimum number of samples, which is the Outlier fault type with 527 samples. Specifically, a subset of the original dataset is generated by randomly sampling each of the seven classes with a limit of 500 samples. Then the training, validation and test dataset are split from the created perfectly balanced dataset with a ratio of 70%: 20%: 10%. The Confusion matrix for the validation set and the test set obtained by training on the designed CNN is depicted in Fig. 10. The number of correct predictions for each class with the corresponding percentage in the total dataset is listed in the diagonal of the matrix, and all the incorrect predictions are listed outside the diagonal. The precision for all 7 classes is listed in the last column while the recall in the last row, with the overall accuracy shown at the bottom right corner. An accuracy of 97.1% and 96.9% is achieved for the validation and test set, respectively, which indicates the good overall prediction capability of the trained CNN; this trained model is denoted as Model-A. The precision and recall, which are the reliability indicators regarding the classification results and ground truth labels, are highlighted in grey alongside the confusion matrix. Note that for both validation and test dataset, these indicators are observed to be consistently more than 90% for each of the classes in the balanced dataset, which demonstrates that the proposed CNN model has a strong learning ability, giving good performance even with such a small training dataset.

Further insight into the learning process of the CNN model can be inferred by visualizing the activations, defined as the outputs of different layers for a given input. A convolutional layer of a CNN usually contains a certain number of filters, and the activations of the convolutional layer will have the same number of channels of output, represented as grayscale images. Strong positive activation is indicated by white pixels, and strong negative activation is indicated by black pixels, whereas mostly gray-colored channels indicate weak activation. Note that the pixel position in the activation image directly corresponds to the same location in the input image. Therefore, features learned by CNN can be investigated by observing which areas the convolutional layers activate and comparing with the corresponding areas in the original images. Fig. 11 shows the filter activations for a sample input image from



(a) Validation dataset (b) Test dataset

Fig. 10 Confusion matrix for perfectly balanced dataset (Model-A)

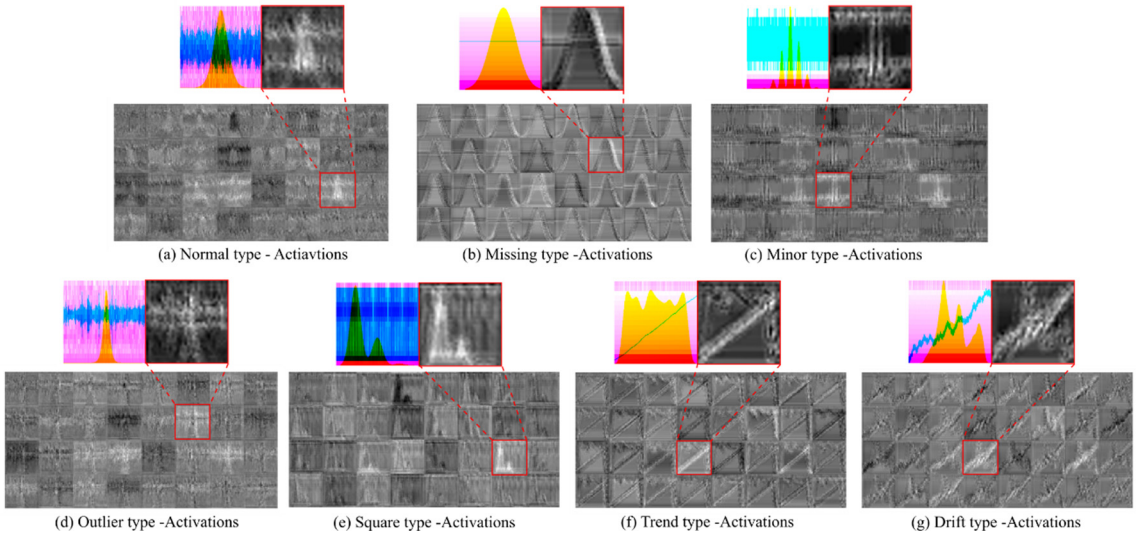


Fig. 11 Filter activations for different fault types

each of the seven fault classes by the last convolutional layer (the deepest convolutional layer containing 32 filters, see in Fig. 6) of the CNN model and the activations are represented by 32 channels of 32 × 32 grayscale images. Most convolutional neural networks learn to detect features like color and edges in their first convolutional layer. In deeper convolutional layers, the network learns to detect more complicated features, because later layers build up their features by combining features of earlier layers. Therefore, complex features such as the multi-modal PDF shape in class Trend are extracted by the last convolutional layer, as shown in Fig. 11(f). In each fault case, the strongest activation channel among the 32 is highlighted in red in Fig. 11 and compared with the corresponding input image. The strong influence of the PDF channel towards activation is clearly demonstrated by the high intensity of pixels observed at the same positions, indicating the important benefit PDF channel provides towards the classification task.

3.2.2 Imbalanced full-dataset

In this case, the full dataset is used for training and testing the CNN model; the training is performed for 10 Epochs until the loss and accuracy curves indicate saturation with an increase in iterations, as observed in Fig. 12. This trained model is denoted as Model-B. The corresponding confusion matrix for the validation and test datasets is shown in Fig. 13. The effect of imbalance in the dataset can be seen from the inferior precision and recall achieved for the Outlier fault type (class 4) in both the validation and test datasets. For the validation dataset shown in Fig. 13(a), 21.9% of class 4 gets misclassified as the 2 larger classes, class 1 and 3, resulting in a relatively lower recall of 78.1% for class 4; 21.9% of samples classified to be class 4 are actually from class 1 and 3, resulting in the precision for class 4 being as low as 78.1%. Similarly for test dataset shown in Fig. 13(b), 81.1% of class 4 gets misclassified as the 2 dominant classes, class 1 and 6, as well as 5.7% misclassified as class 2, 3 and 7, resulting in the recall for class 4 being as low as 13.2%; 41.7% of samples classified to be class 4 are actually from

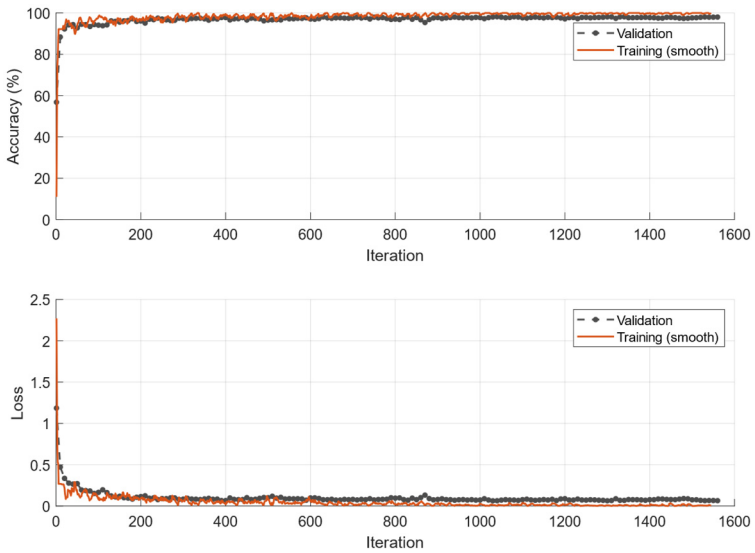
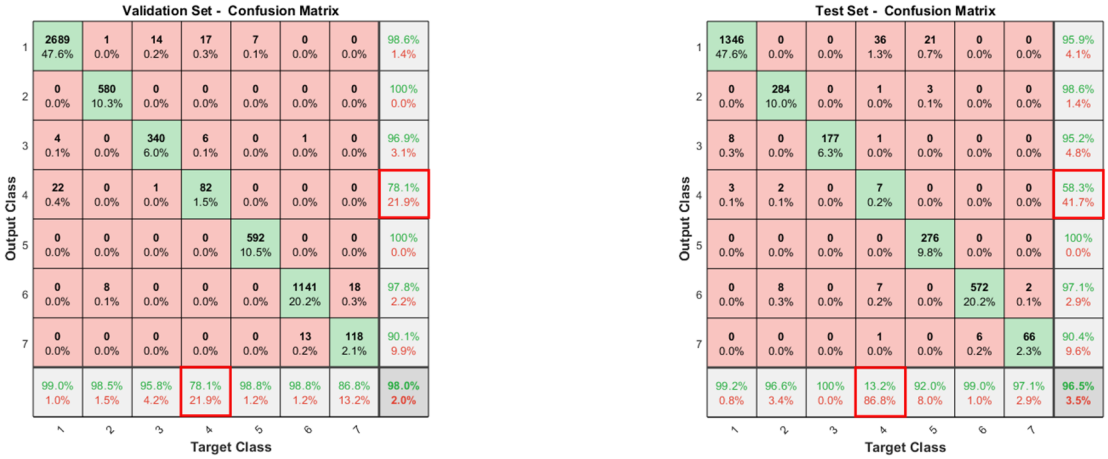


Fig. 12 Accuracy and Loss curve of the training progress



(a) Validation dataset (b) Test dataset

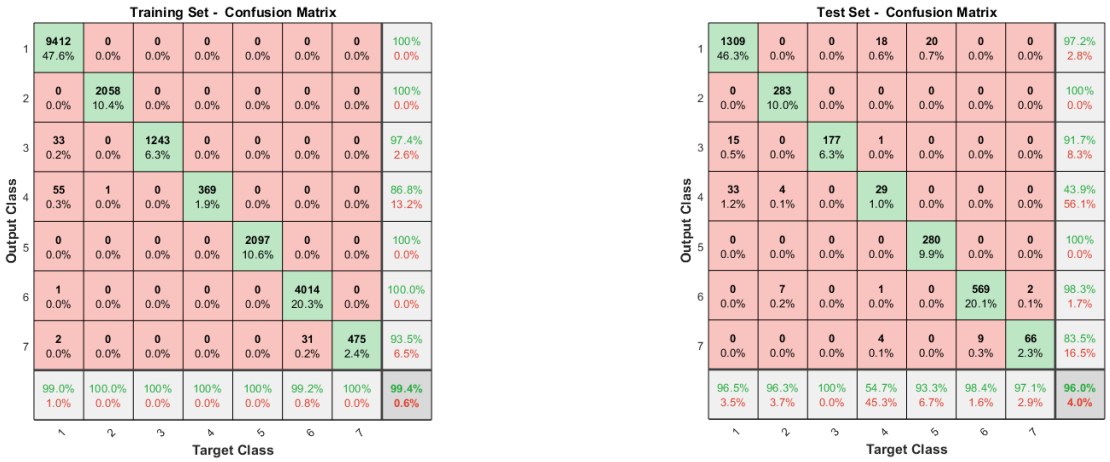
Fig. 13 Confusion matrix for imbalanced full dataset (Model-B)

class 1 or 2, resulting in the precision for class 4 being as low as 58.3%. Note that in the test dataset, both the recall and the precision for class 4 are of lower values than those in validation dataset, indicating overfitting. The inferior performance for class 4 recall and precision implies that the class with the fewest number of samples remains under-trained, even though a high-overall accuracy of 98% for validation dataset and 96.5% for test dataset are achieved.

3.2.3 Imbalanced full-dataset with median frequency balancing

Utilizing the full dataset while accounting for the imbalance in classes is achieved by using the concept of median frequency balancing introduced in Section 2.2.4. The class weight in the loss function updates is chosen to be inversely proportional to the size of each class. Two types of median frequency balancing approaches are implemented: 1) class weights based on class proportions in the current batch and 2) class weights based on class proportions in the full dataset. For our dataset, the latter weight balancing approach performs better than the former in terms of recall, accuracy, and overall stability during training. The trained model using the second balancing approach is designated as Model-C. Fig. 14 depicts the confusion matrices of Model-C for the training and test dataset. However, even with median frequency balancing, the CNN model is still unable to learn class 4 as well as the other categories, which is seen from the relatively lower precision of 86.8% for the training dataset and 43.9% for the test dataset. Note that the recall for class 4 is 100% in the training dataset, while 54.7% in the test dataset, indicating the presence of overfitting.

Model C, employing median frequency balancing, improved the recall from 13.2% to 54.7% for class 4, as observed from the highlights shown on the confusion matrices in Figs. 13(b) and 14(b), indicating the efficacy of median frequency balancing method on reducing imbalance and associated data bias.



(a) Training dataset (b) Test dataset

Fig. 14 Confusion matrix for imbalanced full dataset with mean frequency balancing (weight from the full dataset) (Model-C)

Table 2 Overall accuracy statistics for 10 trials

	Training	Validation	Testing
Average accuracy (%)	99.98	95.87	95.07
Standard deviation	0.06	0.84	0.48

Table 3 Overall accuracy statistics for 10-fold cross-validation

	Training	Validation	Testing
Average accuracy (%)	99.99	95.83	95.40
Standard deviation	0.03	0.48	0.87

4. Discussion

4.1 Cross-validation tests

To test the robustness of the model for different training and test sets, two set of experiments are performed using the perfectly balanced dataset case. A subset of the full dataset is generated by randomly sampling each of the seven classes with a limit of 500 samples. Then the training, validation and test dataset are split in the ratio of 70%: 20%: 10%.

In the first set of experiments, the random sampling of 500 samples per class is performed 10 times. Thereafter, in the samples are split in a sequence, the first 70% as training, followed by 20% for validation, and the last 10% for testing. The standard deviation and average overall accuracy achieved for the 10 trials are tabulated in Table 2.

In the second set of experiments, a 10-fold cross-validation study is performed. Cross-validation limits the number of samples to reduce the prediction bias in assessing general performance of the model tested on unseen data. For a 10-fold cross-validation, first, the dataset is split evenly into 10 groups. Then, one group is retained for testing, while the remaining 9 groups are used for the training set. Similarly, the process is repeated 10 times with each of the 10 groups used exactly once for testing. Thereafter, the ten results are averaged to produce a single estimate. The 10-fold cross-validation results are presented using a boxplot in Fig. 15, indicating the recall, precision, and overall accuracies. The standard deviation and average overall accuracy for the 10-fold cross-validation is tabulated in Table 3.

The high overall accuracies for the test set with a small standard deviation for the 10-fold cross-validation verifies

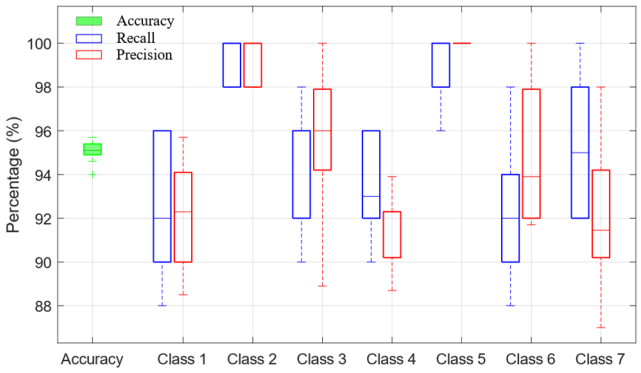


Fig. 15 10-fold cross-validation

the efficacy of the model against bias in data. Also, the range of values observed for recall and precision signifies the robustness of the CNN model in fault classification for different training and test sets.

4.2 Ablation experiments

An ablation study is performed to understand the significance of using multi-channel input and the contribution of each component to the overall model. Three cases are considered, in which only a single-channel from the three channels (i.e., time, spectrogram, and PDF) is used to create a small perfectly balanced dataset for training, in an approach similar to that used for Model-A (see Section 3.2.1). The same CNN architecture is re-used for a fair comparison of performance against Model-A trained with multi-channel input. The single-channel under consideration is stacked three times to match the input dimension (i.e.,

Table 4 Performance comparison with ablation study

Input channel	Accuracy (%)	Minimum recall (%)	Minimum precision (%)
Time	91.7	82.0 – class 3	81.8 – class 1
Spectrogram	88.6	68.0 – class 4	82.4 – class 1
PDF	91.7	86.0 – class 4	84.9 – class 1
Multi-channel (Model-A)	96.9	92.0 – class 4	94.0 – class 7

256 × 256 × 3) of the 2D CNN architecture. Table 4 compares the performance on the test dataset for the single-channel trained networks with the multi-channel trained Model-A.

For all the metrics compared in Table 4, it can be seen that Model-A with multi-channel input consistently performs better than any of the single-channel trained networks. Using only time-channel as input, the minimum precision occurs for Class 1 which gets misclassified with Class 3, while using only spectrogram or PDF channel the misclassification occurs with class 4. Whereas, the use of multi-channel (Model-A) resolves this misclassification to a large extent, with both the minimum precision and recall above 92%. The results demonstrate the significance of the proposed multi-channel input approach with a 2D CNN for the anomaly classification task.

4.3 Evaluation of model-A

The performance of the proposed fault detection architecture has been shown to improve upon previously reported approaches for rare faults, while performing equally well for more common fault types. Key to this improvement was inclusion of the PDF channel to extract global features, in addition to a time and spectrogram channel. Moreover, training on a small perfectly balanced dataset is shown to achieve high overall accuracy, even when tested on a much larger unseen dataset, by including the three channels as input to the CNN. These points are discussed in more detail in the following paragraphs.

Typically, CNNs learn to detect global features such as color and edges in their first convolution layers. As the layers get deeper, more complex and nuanced features are learned, while building upon the features extracted in previous layers. Sufficient features to distinguish between rare fault types are not learnt in the previously proposed single-channel (Bao *et al*. 2019) and dual-channel (Tang *et al*. 2019) approaches, as indicated by the relatively large drop in recall values for the rare classes. Fig. 16 represents the maximum (strongest) activation channel for each of the successive convolution layers in the CNN model trained for a perfectly balanced dataset (Model-A) using a representative input image for each fault types. Recall that time, spectrogram, and PDF channels in the input image are represented in the image by pixels that are cyan, magenta, and yellow, respectively. Convolution layer 1 activation shows that the first layer primarily learns the edges and color of the time-signal channel and the spectrogram channel of the input image. Also, after a series of channel transformations followed by the grouped convolution layer (before Conv Layer 3) clearer fusing of features is noticed. Strong activation patterns (i.e., indicated by bright white pixels) aligning with the shape of the PDF channel are observed in the deeper layers. Note that the gray-colored pixel regions correspond to weak-activations in the channel, and it has a lesser influence towards the classification. For the rare Drift fault type the activations primarily from the PDF channel allow differentiation from the Trend fault type. However, Drift and Trend share a similar pattern for the spectrogram channel, and often contrasting traits in the time-domain channel are also minimal, indicating chances of misclassification. Nevertheless, the deeper layers detect and learn the shape patterns observed in the PDF channel, which is the main distinguishing feature among these two classes, thus improving the performance of CNN on fault types Drift and Trend.

The performance of Model-A (i.e., trained using the small perfectly balanced dataset) appears to be significantly better than Model-C (i.e., trained using the full imbalanced dataset with median frequency balancing), as can be seen by comparing the associated confusion matrices in Figs. 10(b) and 14(b), respectively. This comparison, though, is not entirely fair, because Model-C was tested on a much larger dataset. Therefore, Model-A is now tested on 10% of the full dataset, so that Model-A and Model-C can be compared against each other fairly (i.e., using test datasets of the same size). The confusion matrix for Model-A tested on this larger dataset is shown in Fig. 17. Comparing the confusion matrices in Figs. 17 and 14(b), Model-A and Model-C are seen to achieve similar overall accuracy of 95.9% and 96%, respectively. Although only trained on a small perfectly balanced dataset, Model-A has a high score in recall for all classes, which indicates that Model-A can learn all the classes effectively and correctly classify them with respect to the ground truth labels. In contrast, Fig. 14(b) shows that Model-C only achieves 54.7% recall for the rare fault class, type 4, indicating that even median frequency balancing does not allow rare classes to be fully learned. Comparing with some of the previous works, Bao *et al*. (2019), using single-channel input to a DNN, achieved an accuracy of 87% with a minimum recall of 47.9% for Class 7. At the same time, Tang *et al*. (2019), using dual-channel input to a CNN, achieved an accuracy of 94.1% on a large testing set with a minimum recall of 57.1% for Class 7. Our results, from Fig. 17, show an accuracy of 95.9% with a significant improvement in minimum recall to 92.4% for Class 6.

However, the performance of Model-A is not perfect. For example, the performance for Model-A may be due, in part, to overlap between some of the randomly selected test samples from the 10% of the full dataset with the balanced training dataset. Additionally, Model-A has relatively poor performance in precision for classes 4 (49.5%) and 7 (63.6%) when tested on 10% of full dataset. The low precision for class 4 is due to some class 1 faults being misclassified as class 4; similarly, some class 6 faults are misclassified as class 7.

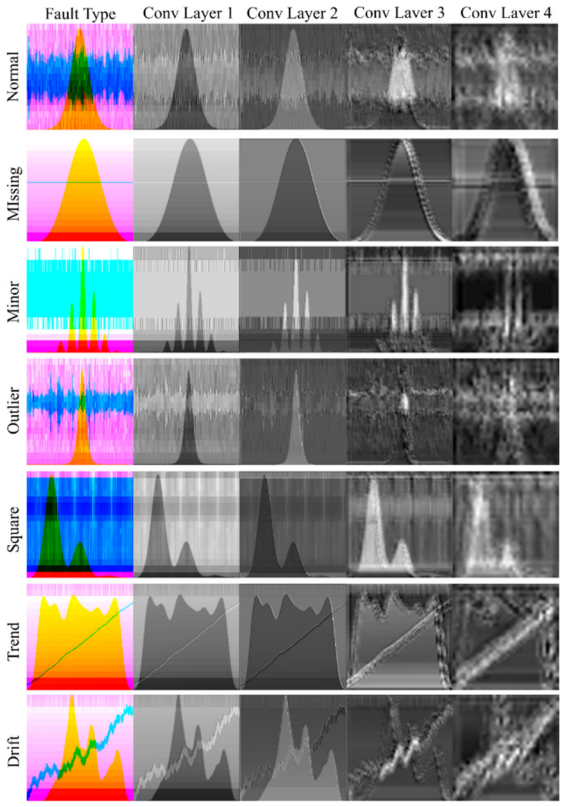


Fig. 16 Maximum activation channels of convolution layers for different fault types

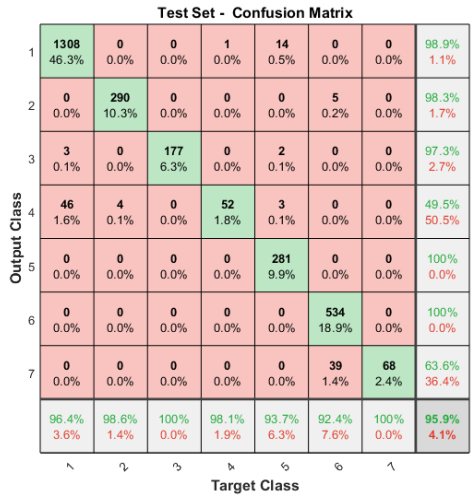


Fig. 17 Confusion matrix for Model-A tested on a larger dataset (10% of full dataset)

The low precision for class 4 is due to some class 1 faults being misclassified as class 4; similarly, some class 6 faults are misclassified as class 7. Intuitively, this result is because Model-A has seen only 500 samples of each class 1 and 6, whereas the full dataset has over 13000 and 5000 samples in classes 1 and 6, respectively. Nevertheless, these results demonstrate that Model-A, trained only on a balanced dataset of relatively small size, achieves performance levels comparable to models trained on a much-larger dataset.

5. Conclusions

A 3-channel image-based pre-processing approach incorporating the temporal nature of anomalies in a 1-D time-series signal has been proposed for data anomaly identification using a CNN. To improve fault identification for relative rare fault types, the input time-series data was converted into a 3-channel image composed of time, spectrogram, and probability density function (PDF), before feeding it into a 2D CNN. The proposed CNN architecture using grouped convolution layers enables efficient model learning. The spectrogram and PDF channels in the trained network well generalize the distinguishing features observed in various fault types, particularly for rare anomalies. The efficacy of the approach was validated using a dataset comprised of a one-month acceleration time record for a long-span cable-stayed bridge in China. The effect of imbalance in the dataset was studied considering three training models: (a) small-sized perfectly balanced subset of full-dataset, (b) imbalanced full dataset, (c) imbalanced full dataset with median frequency balancing. The model trained with a perfectly balanced dataset only using 500 samples in each class gives the overall best performance. The model achieves 97.1% and 96.9% accuracy on the validation and test set, with above 92% recall and precision on each class. Moreover, an accuracy of 95.9% was observed even when tested on a largely unseen dataset, with a high recall even on the rare fault classes. These results demonstrate the capacity of the proposed approach to achieve high performance levels, even though only trained on a limited balanced dataset, offering a viable solution for data-anomaly identification for full-scale structures, particularly when labeled training data is limited.

Acknowledgments

The authors would like to thank the organizers of the International Project Competition for SHM (IPC-SHM 2020), ANCRiSST, Harbin Institute of Technology (China), and University of Illinois at Urbana-Champaign (USA) for generously providing the data used in this study. We gratefully acknowledge the guidance and constructive criticism offered by Dr. Yasutaka Narazaki, Zhejiang University-UIUC Institute throughout this study. Additionally, the second and third authors acknowledge the partial support of this research by the China Scholarship Council.

References

Bao, Y., Tang, Z., Li, H. and Zhang, Y. (2019), "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., Li, J., Nagayama, T., Xu, Y., Spencer Jr., B.F. and Li, H. (2021), "The 1st International Project Competition for Structural Health Monitoring (IPC-SHM, 2020): A summary and benchmark problem", *Struct. Health Monitor.*, **20**(4), 2229-2239. https://doi.org/10.1177/14759217211006485

Friswell, M.I. and Inman, D.J. (1999), "Sensor validation for smart

structures", *J. Intell. Mater. Syst. Struct.*, **10**(12), 973-982. https://doi.org/10.1106/GVD2-EGPN-C5B1-DPNX

Fu, Y., Peng, C., Gomez, F., Narazaki, Y. and Spencer Jr., B.F. (2019), "Sensor fault management techniques for wireless smart sensor networks in structural health monitoring", *Struct. Control Health Monitor.*, **26**(7), e2362. https://doi.org/10.1002/stc.2362

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

Ibarguengoytia, P.H., Sucar, L.E. and Vadera, S. (2001), "Real time intelligent sensor validation", *IEEE Transact. Power Syst.*, **16**(4), 770-775. https://doi.org/10.1109/59.962425

Im, J., Park, H. and Takeuchi, W. (2020), "Advances in remote sensing-based disaster monitoring and assessment", *Remote Sensing*, **11**(18), 2181. https://doi.org/10.3390/rs11182181

Ioannou, Y. (2017), "A Tutorial on Filter Groups (Grouped Convolution)", A Shallow Blog about Deep Learning. https://blog.yani.ai/filter-group-tutorial/, Accessed 04/24/2021

Kampffmeyer, M., Salberg, A.-B. and Jenssen, R. (2016), "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Las Vegas, NA, USA, June.

Kerschen, G., Boe, P.D., Golinval, J.-C. and Worden, K. (2004), "Sensor validation using principal component analysis", *Smart Mater. Struct.*, **14**(1), 36-42. https://doi.org/10.1088/0964-1726/14/1/004

Krawczyk, B. (2016), "Learning from imbalanced data: open challenges and future directions", *Progress Artif. Intell.*, **5**, 221-232. https://doi.org/10.1007/s13748-016-0094-0

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Proceedings of the Advances in Neural Information Processing Systems*, Lake Tahoe, USA, December.

Kullaa, J. (2010), "Sensor validation using minimum mean square error estimation", *Mech. Syst. Signal Process.*, **24**(5), 1444-1457. https://doi.org/10.1016/j.ymssp.2009.12.001

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

Nagarajaiah, S. and Yang, Y. (2017), "Modeling and harnessing sparse and low-rank data structure: a new paradigm for structural dynamics, identification, damage detection, and health monitoring", *Struct. Control Health Monitor.*, **24**, e1851. https://doi.org/10.1002/stc.1851

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

Oh, B.K., Glisic, B., Kim, Y. and Park, H.S. (2020), "Convolutional neural network–based data recovery method for structural health monitoring", *Struct. Health Monitor.*, **19**(6), 1821-1838. https://doi.org/10.1177/1475921719897571

Ou, J. and Li, H. (2009), *Structural Health Monitoring of Civil Infrastructure Systems*, Chapter 15. Structural health monitoring research in China: trends and applications), Woodhead Publishing Limited, Sawston, Cambridge, UK.

Smarsly, K. and Law, K.H. (2014), "Decentralized 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

Sohn, H., Farrar, C.R., Hemez, F.M. and Czarnecki, J.J. (2002), "A Review of Structural Health Review of Structural Health Monitoring Literature 1996-2001", *Proceedings of the 3rd World Conference on Structural Control*, Como, Italy, April.

Tang, Z., Chen, Z., Bao, Y. 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), e2296. https://doi.org/10.1002/stc.2296

Yang, Y. and Nagarajaiah, S. (2016), "Harnessing data structure for recovery of randomly missing structural vibration responses time history: Sparse representation versus low-rank structure", *Mech. Syst. Signal Process.*, **74**, 165-182. https://doi.org/10.1016/j.ymssp.2015.11.009

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

Yi, T.-H., Huang, H.-B. and Li, H.-N. (2017), "Development of sensor validation methodologies for structural health monitoring: A comprehensive review", *Measurement*, **109**, 200-214. https://doi.org/10.1016/j.measurement.2017.05.064

Zhang, Y. and Lei, Y. (2021), "Data Anomaly Detection of Bridge Structures Using Convolutional Neural Network Based on Structural Vibration Signals", *Symm. Struct. Health Monitor.*, **13**(7), 1186. https://doi.org/10.3390/sym13071186