Journal of Infrastructure Preservation and Resilience

RESEARCH Open Access

Multiclass anomaly detection in imbalanced structural health monitoring data using convolutional neural network

Mengchen Zhao[1], Ayan Sadhu[2*] and Miriam Capretz[1]

Abstract

Structural health monitoring (SHM) system aims to monitor the in-service condition of civil infrastructures, incorporate proactive maintenance, and avoid potential safety risks. An SHM system involves the collection of large amounts of data and data transmission. However, due to the normal aging of sensors, exposure to outdoor weather conditions, accidental incidences, and various operational factors, sensors installed on civil infrastructures can get malfunctioned. A malfunctioned sensor induces significant multiclass anomalies in measured SHM data, requiring robust anomaly detection techniques as an essential data cleaning process. Moreover, civil infrastructure often has imbalanced anomaly data where most of the SHM data remain biased to a certain type of anomalies. This imbalanced time-series data causes significant challenges to the existing anomaly detection methods. Without proper data cleaning processes, the SHM technology does not provide useful insights even if advanced damage diagnostic techniques are applied. This paper proposes a hyperparameter-tuned convolutional neural network (CNN) for multiclass imbalanced anomaly detection (CNN-MIAD) modelling. The hyperparameters of the proposed model are tuned through a random search algorithm to optimize the performance. The effect of balancing the database is considered by augmenting the dataset. The proposed CNN-MIAD model is demonstrated with a multiclass time-series of anomaly data obtained from a real-life cable-stayed bridge under various cases of data imbalances. The study concludes that balancing the database with a time shift window to increase the database has generated the optimum results, with an overall accuracy of 97.74%.

Keywords: Structural health monitoring, Anomaly detection, Convolutional neural network, Imbalanced dataset, Hyperparameter tuning

Introduction

Civil infrastructure is rapidly aging worldwide due to increasing natural calamities, population, and operational loads. Structural health monitoring (SHM) system [7, 13] is a valuable data-driven technology that involves the integration of sensors, data transmission, data analysis, and decision-making to protect these structures. With the advent of next-generation sensors, SHM has been an emergent and powerful diagnostic tool for damage detection and disaster mitigation of large-scale structures [24, 34, 45]. In the past several decades, SHM systems have been widely applied to different civil infrastructures, including bridges, tunnels, railways, and buildings. Without an accurate and timely SHM, it may result in a substantial and costly repair of the infrastructure when there is significant damage identified in the structure [5, 30, 41]. Therefore, it is critical to detect any damages at the earliest possible stage to avoid substantial service interruption or potential safety risks. The lifespan of a civil infrastructure could also be lengthened with proactive detection and proper maintenance [42].

*Correspondence: asadhu@uwo.ca

[2] Department of Civil and Environmental Engineering, Western University, London, Ontario, Canada
Full list of author information is available at the end of the article

However, being exposed to outdoor environments, SHM systems of civil infrastructure face significant implementation challenges from environmental and measurement noises, sensor faults, data acquisition malfunctions, and operational conditions during the data collection stage [1]. Sensors are also affected by poor installation, inadequate maintenance, lack of protective coverings, and delayed replacement [23]. This often leads to the collection of anomalous data, requiring anomaly detection (AD) techniques such that only good quality data is used for structural diagnostics and decision-making.

AD is an essential step for data cleaning and preparation purposes of SHM before any diagnostic algorithms are implemented. The lifetime of SHM sensors is typically much shorter than that of the infrastructure [25], resulting in the normal aging and the need for the replacement of sensors. Installed sensors are often exposed to various weather conditions [10] and accidental incidences, which cause the sensors to lose connectivity, resulting in a hindrance to the measurement for key feature extraction. Sensor malfunctioning and faulty signals may potentially cause destabilization of the entire system and provide misleading information about the current condition of the structures [48, 50]. Without proper data cleaning processes, the SHM technology is unable to provide valuable insights even if advanced damage diagnostic techniques have been applied. With the rapid growth of computational power, algorithm improvements, and data collection, various machine learning (ML) and deep learning (DL) methods [44] have been applied to the realm of SHM and AD.

AD techniques and algorithms are often proposed for binary classification and studied on in-lab or simulated data, aiming to separate anomalies from normal data. Steiner et al. [46] focused on detecting anomalous data due to sensor faults in wireless sensor networks. They proposed support vector regression for preserving the constrained resources of wireless sensor nodes. The proposed method was tested on a prototype wireless SHM system on a four-story shear frame structure for automated and decentralized sensor fault detection and isolation. Zhu et al. [56] explored a temperature-driven moving principal component analysis method for AD. The proposed method calculated the covariance matrix within a pre-selected window size instead of the whole time series. Using the simulated case studies, it was concluded that the proposed method was more sensitive than the traditional moving principal component analysis, and it was able to detect anomalies during the expected period without any delay. Yang et al. [53] proposed a model to test a dataset obtained from a three-story laboratory structure. The proposed model was an end-to-end-trainable deep scenario based on a deep support vector data description. The proposed model was to map most of the data network representation into a hypersphere and define an anomaly score based on the distance of the point to the center of the hypersphere. Data that lie far away from the center or outside the hypersphere was identified as anomaly data.

Maes et al. [27] studied the effect of linear regression and linear principal component analysis models for AD in tunnel monitoring. The techniques were applied to a dataset obtained from an in-situ monitoring campaign in a tunnel. It was concluded that linear regression analysis was not suited as an AD tool for the tunnel-soil system due to a strong temporal dependence. The simulated numerical analysis showed the need for a sufficiently long training period, as well as the need for visual inspections and analysis coupled with linear principal component analysis. Wedel and Marx [52] studied the transient relationship between the air temperature and the bridge temperature with regression methods. The regression model was tested with long-term monitoring data of bridges in Germany. The research concluded that the ML methods could be used to detect sensor faults by comparing real measured values and predicted behavior. Sensor compensation via predicted sensor measurement was possible, assuming isolated and local sensor fault.

AD can be viewed as a data cleaning process for data compression and data reconstruction. Ni et al. [31] focused on finding an efficient unsupervised method for SHM data compression. They proposed a one-dimensional convolutional neural network (1D-CNN) with Adam Algorithm and mini-batch gradient descent for AD. Data were defined as abnormal when the measurements showed unacceptable deviations from the true values of the measured variables in either time or frequency domain. However, the proposed model was only useful for binary classification, namely the normal and anomalous categories. Jeong et al. [21] proposed a bidirectional recurrent neural network for sensor data reconstruction based on spatiotemporal correlation. For a system with N sensors, one was treated as the targeted output data, while sensors with high spatial correlations with the output sensor were treated as input sensors, resulting in an N combination. The reconstruction error was used as an indicator to detect potential anomalies of the output sensor. The faulty sensor was isolated based on the idea that the faulty sensor only causes local effects, and the proposed algorithm resulted in a higher computational cost for binary classification. Recently, Mao et al. [28] argued that supervised learning methods of AD had two unresolved challenges: imbalanced dataset and incompleteness of anomalous patterns for the training dataset. They proposed to use deep convolutional generative adversarial networks with autoencoders. The input for the network was transformed Gramian Angular Field images from time-series data, while the output was classified as either normal or anomalous data. Sarmadi and Karamodin [37] also explored unsupervised learning since it only required information on a single known structural state. They introduced an adaptive Mahalanobis-squared distance and one-class kNN rule to formulate a new multivariate distance. The proposed method improved the conventional Mahalanobis-squared distance technique for non-Gaussian or heavy-tailed distribution. The algorithm was tested in a lab truss structure.

Similar to machine learning techniques, researchers also explored various deep learning techniques for AD. Bao et al. [4] explored data visualization by converting the time-series signals into images. This was achieved by splitting the data into sections and plotting it in grayscale images. They trained a deep neural network with greedy layer-wise pre-training and a fine-tuning stage. The proposed model was tested on acceleration data that were divided into six patterns: missing, minor, outliers, squares, trends, and drifts. It resulted in a total accuracy of 87% for one-year test data using a real-life cable-stayed bridge. Tang et al. [47] presented a two-dimensional CNN (2D CNN) with a combination of time-domain response and frequency-domain response as input images. The frequency-domain response was obtained through Fast Fourier Transform. Using a balanced dataset, the proposed model was verified on a long-span cable-stayed bridge, which achieved an overall accuracy of 93%. Arul and Kareem [2] presented to use a relatively new time-series representation named “Shapelet Transform” in combination with a Random Forest classifier to autonomously identify anomalies in SHM data. Jana et al. [20] proposed CNN for detecting the presence of a sensor fault and convolutional autoencoder to reconstruct sensor data based on its identified type. The proposed model was tested using simulated and experimental datasets to demonstrate its performance. However, the model was designed to train data with single fault types using a simulated balanced dataset. Liu et al. [26] proposed to use a generative adversarial network and CNN-based AD technique. The model contains a three-channel input by combining time-series data with its fast Fourier transform and Gramian angular field output. The proposed model used a generative adversarial network for addressing class-imbalance issues, followed by CNN for classification tasks.

Despite a large amount of work on AD using ML and DL, SHM data still has several implementation challenges associated with the availability of balanced training data. Success in implementing DL depends predominately on access to large amounts of data [12]. If the dataset is too small, the lack of sufficient image samples makes it difficult to converge in end-to-end learning [33]. Recently, AD techniques were also explored to imbalance datasets in different disciplines [8, 19, 51]. However, these studies were mostly focused on a single type of anomaly, which often results in poor accuracy for multiple anomalies [3]. To the authors’ knowledge, there exist limited studies on multiclass AD of SHM data using limited imbalanced time-series datasets. Besides the size and imbalance issues of the sensor data, each ML algorithm has a hyperparameter setting. The optimal hyperparameter helps in building a better ML model [32], while the tuning process to find the optimal combination of hyperparameters in the DL models improves the overall accuracy and minimizes the loss function effectively. This paper proposes a hyperparameter-tuned CNN for multiclass imbalanced anomaly detection (CNN-MIAD) modelling of time-series SHM data.

The proposed CNN-MIAD model is a novel approach to solving multiclass AD with a limited imbalanced time-series dataset. In the real world, it is impractical to label a large amount of SHM data. Thus, there may exist a limited input sample where the proposed CNN-MIAD model aims to include an overlapping sliding window to increase the size of the dataset. The performance of this model is compared and examined using the acceleration data from a real-life cable-stayed bridge in three cases with both balanced and imbalanced datasets of varying input sizes. The inclusion of the random search hyperparameter tuning strategy has further optimized the model performance.

The paper is structured as follows: Section 2 presents the proposed CNN-MIAD model with an overview of the CNN methodology and hyperparameter tuning processes; Section 3 illustrates the proposed CNN-MIAD model on imbalanced anomaly data of a real-life cable-stayed bridge using data augmentation; Section 4 discusses the analysis and results of the proposed study. Section 5 presents the key conclusions and outcomes of this research.

Background

Convolutional neural network

Deep learning (DL) can automatically learn abstract features of the original data and classify them effectively, avoiding the shortcoming of requiring handcrafted features designed by engineers [55]. CNN is a type of DL method that has seen rapid progress in recent decades due to the developments in computing power, the advent of large amounts of labelled data, and supplemented by improved algorithms in many disciplines [35, 44]. It is suited for structured data such as images and mainly comprises the convolutional layers, pooling layers and fully connected layers. CNNs have been highlighted in computer vision, image recognition, and classification tasks, which are inspired by the visual cortex of animals [9].

During the training of a convolutional model, the model learns the spatial relationships between features within the target dataset, which may then be applied to test data for classification or recognition tasks. The 2D convolution function is used for the input image in each convolutional layer to produce a tensor of outputs. The convolution is the sum of the products between each image pixel and kernel pixel. Each convolutional layer is followed by a pooling layer to reduce the spatial size of an input array as a form of down-sampling. Adding a pooling layer to the model does not only save the spatial information of pixels but also reduces computational costs [36]. Scherer et al. [38] empirically proved that max-pooling operation was vastly superior for capturing invariance in the data and could lead to improved generalization and faster convergence when compared to a subsampling operation. The last few layers in a CNN model are the fully-connected layers, which require a flattened pooling layer or convolutional layer as the input. The neurons in the fully-connected layers provide a global connection to every neuron in the preceding layer, whereas the neurons in the convolutional layer are only connected to the neighbouring neurons based on the kernel size. The fully connected layers are used to get the probabilities of the input being in a particular class for classification tasks.

Hyperparameter tuning

Every machine learning (ML) system has hyperparameters, and it is critical to set these hyperparameters to optimize performance [17]. Examples of hyperparameters include the learning rate, batch size, and the number of neurons for the hidden layers. The learning rate of the model has a strong impact on the stability and efficiency of training. Choosing a learning rate that is too large results in the instability and divergence of the objective function, whereas choosing a learning rate that is too small results in slow learning and inefficiency [54]. Batch size defines the number of inputs that will be propagated through the network each time. Batch normalization, as one of the common regularization strategies, aims to deal with noise data, the limited size of the training data, and the complexity of classifiers to avoid overfitting [49]. Using a smaller batch size requires less memory and results in faster training; however, setting the batch size too small will result in less accuracy for the estimate of the gradient. Deciding the number of neurons in the hidden layers is important as too few neurons will result in the underfitting of the model, whereas too many neurons may result in overfitting and increase the time for training [16].

However, these hyperparameters cannot be directly estimated from data, and there exist no analytical formulas to calculate their appropriate values [22]. This leads to the need for hyperparameter optimization. To tune the hyperparameters, grid search is the most popular method, which allows the user to specify a finite set of hyperparameter combinations. As the number of tuning hyperparameters increases, the required number of function evaluations grows exponentially due to the full factorial design [29]. This results in wasted computational resources and inefficiencies as every combination of hyperparameter values has to be examined. An alternative to grid search is the random search algorithm, where a user only specifies the search space as a boundary of hyperparameter values. As the name suggests, the combinations of hyperparameters are randomly sampled within the domain. Bergstra and Bengio [6] proved that randomly chosen trials are more efficient for hyperparameter optimization than grid search. There also exists Bayesian optimization algorithm, which contains two key components: the probabilistic surrogate model consisting of a prior distribution that models the unknown objective function and an acquisition function that is optimized for deciding where to sample next [39]. This global optimization strategy requires a time-consuming procedure and results in a high computational cost to perform more computation on the next iterations. It is also usually unclear for a given practical problem what an appropriate choice is for the covariance function and its associated hyperparameters [43]. Therefore, due to the proven efficiency of the random search algorithm, the proposed method incorporates random search as the hyperparameter tuning method.

Proposed approach

Data imbalance and augmentation

2D CNN models use images or image-like matrix input. To convert time-series data into image representation, the sensor readings are plotted against time. This conversion of time-series data into image representation serves the purpose of changing dimensions for data preprocessing. The time frame for each image should be kept constant to maintain the consistency of the input. Each image in the whole database is converted to grayscale with a matching dimension to the required CNN model input. The dataset is randomly split into a training set, validation set, and testing set based on a 70–20-10 ratio. Both the training set and the validation set are used to train the CNN model with hyperparameter tuning processes. The best combination of hyperparameters of a model is determined by the highest training accuracy and lowest validation loss. After generating the best-trained model with the appropriate hyperparameters, the testing set is used to evaluate the accuracy of the fully trained CNN model.

However, a dataset is described as imbalanced when at least one category has relatively fewer samples than the other categories in classification problems. For classification tasks, the DL algorithm favours a balanced dataset as it provides equal information for each class, otherwise, the minority observations are likely treated as noise and ignored in the process. In some cases, most test data samples are classified into the majority group, resulting in the classification accuracy of the minority class tending to be much lower than that of the majority class [18]. This causes the imbalanced dataset to have higher false negatives on minor classes. The predictive output of classifiers trained with an imbalanced dataset is also biased because classifiers are less sensitive to the minority classes [14]. Furthermore, the class with the lowest number of samples is usually the class of interest from a learning task perspective [15]. Augmentation techniques can be used to address the imbalanced data issue, especially when the original dataset is too small [11, 40]. In time-series data, one way to address the data imbalance issue is through data augmentation using a sliding window. As illustrated in Fig. 1, time-series data with a non-overlapping sliding window requires more raw data points to generate the same amount of input samples. The number of raw data points required would further decrease with a shortened interval between each time shift. Using this shortened time shift strategy, the number of input samples for the minor classes will increase with the same amount of data points in time series data.

Proposed anomaly detection model

Figure 2 shows the architecture and parameters of the proposed CNN-MIAD model. The required input for the proposed model is grayscale images with an input size of 100×100 pixels obtained from the time-series of anomaly data. This is the first layer feeding into the CNN-MIAD model. The input then passes through three sets of convolution and max-pooling layers. The channel size increases with each convolution layer, whereas the dimension decreases due to the kernel size of 5×5. The kernel size for the max-pooling layers is 2×2, which further reduces the size of the convolved feature map. The default for the convolutional layer is a stride of one with no padding, whereas for the max-pooling layer is a stride of 2 with no padding. The tensor is then fed into a fully connected layer with a rectified linear unit (ReLU), where the number of neurons is one of the tuning hyperparameters. The output is then fed into the final layer, which uses a softmax classifier to generate the final output. The tuned hyperparameters in this model include the number of neurons in the second fully connected layer, the batch size, and the learning rate.

Performance metrics

Confusion matrices are a way of visualizing the performance of a classification model, where the predicted

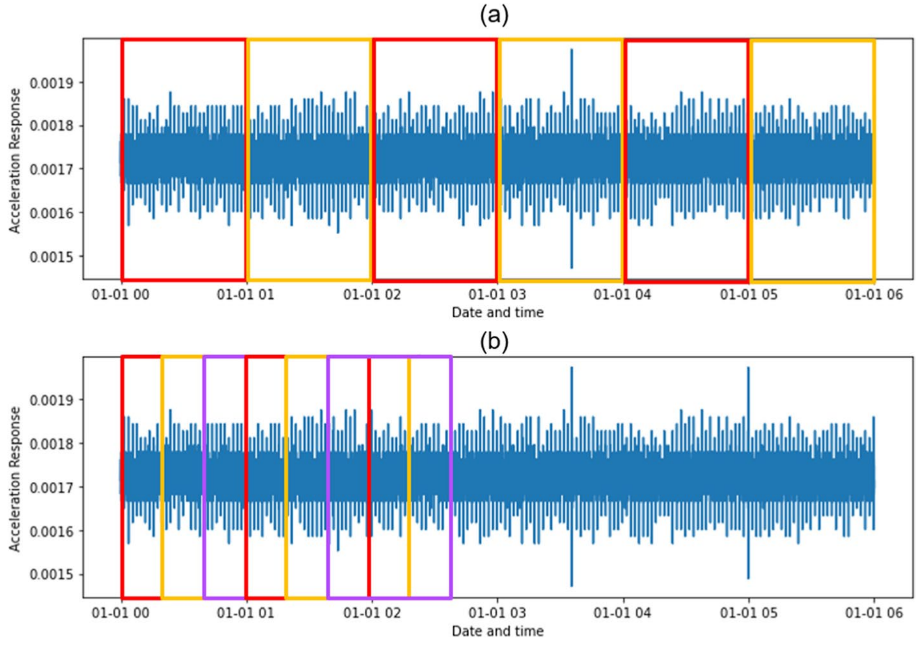


Fig. 1 Comparison between the number of data points needed between (**a**) non-overlapping hourly time shift and (**b**) shortened interval

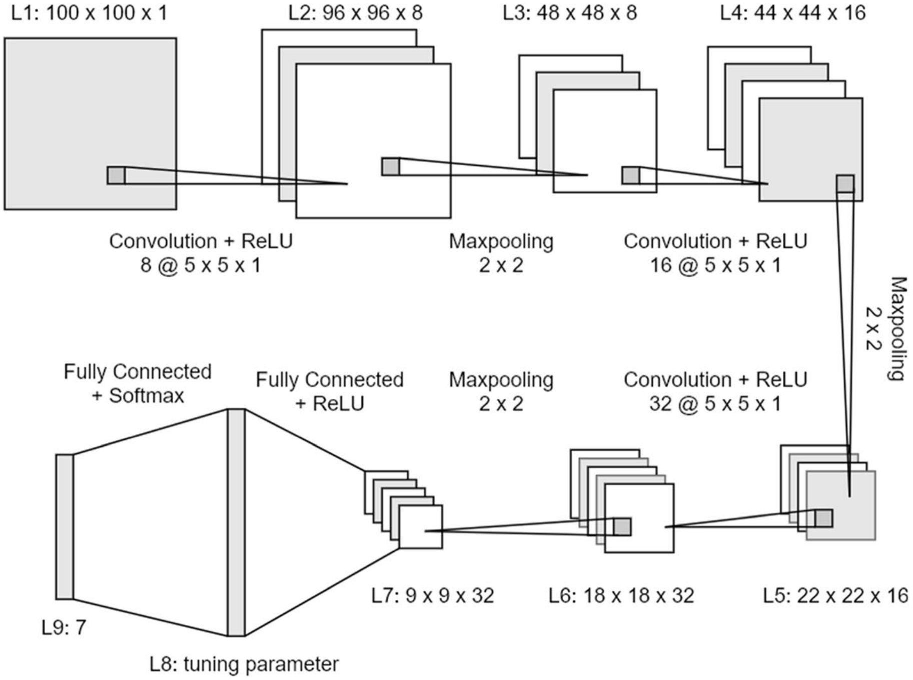


Fig. 2 The proposed CNN-MIAD model

labels are plotted against the actual labels of a dataset. By doing this, it can be seen how accurately a model is able to classify a particular class, especially for a multiclass classification problem. For this study, the confusion matrix with detailed analysis on the accuracy, precision, recall and *F1* score is used to analyze the performance of the model for various input datasets. The definitions of these metrics are defined in Eq. 1.

$$\begin{cases} \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \\ \text{Precision} = \frac{TP}{TP+FP}, \\ \text{Recall} = \frac{TP}{TP+FN} \\ F1 = 2 \times \frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \times 100\% \end{cases} \tag{1}$$

Where, true positive (*TP*) means a sample is correctly classified into its own category; true negative (*TN*) means a sample does not belong to a particular class and has been correctly identified in the other category; false positive (*FP*) means a sample has been misclassified from one of the other categories to the target category; false negative (*FN*) means a sample has been misclassified to one of the other categories from the target category. Accuracy is a basic metric representing the ratio of correct predictions and the total number of predictions. Precision returns the ratio between the *TP* and all the predicted positives, measuring a classifier's exactness. On the other hand, recall returns the ratio between the TP and all the labelled positives, providing a measure of how accurately the model is able to identify the relevant data. *F1* score is a harmonic mean of the precision and recall scores. These metrics are used to report the performance of the proposed CNN-MIAD model as the last step. Figure 3 provides a summary of the workflow of the proposed CNN-MIAD method to perform multiclass AD using imbalanced data.

Description of the dataset

The proposed CNN-MIAD model is analyzed with a dataset that consists of one month of acceleration data for a long-span cable-stayed bridge in China (IPC-SHM 2020). The dataset contains 38 acceleration sensors, and their locations are illustrated in Fig. 4. Each sensor collects data at a sampling frequency of 20 Hz. The data collected by the sensors are labelled into seven different classes: normal, missing, minor, outlier, square, trend, and drift. A detailed description of each of the seven classes is listed in Table 1. As part of the data preprocessing, the time series data is converted into images. Figure 5 shows an example of the input image for each class, the x-axis for each image represents the time, and the y-axis represents the acceleration. Each image has a one-hour time duration in the original database. The dimension for each of the original images is 875 × 656 pixels.

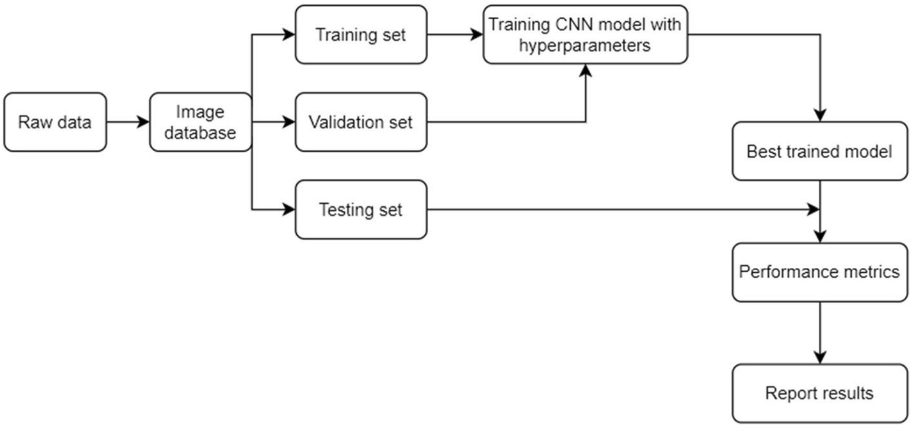
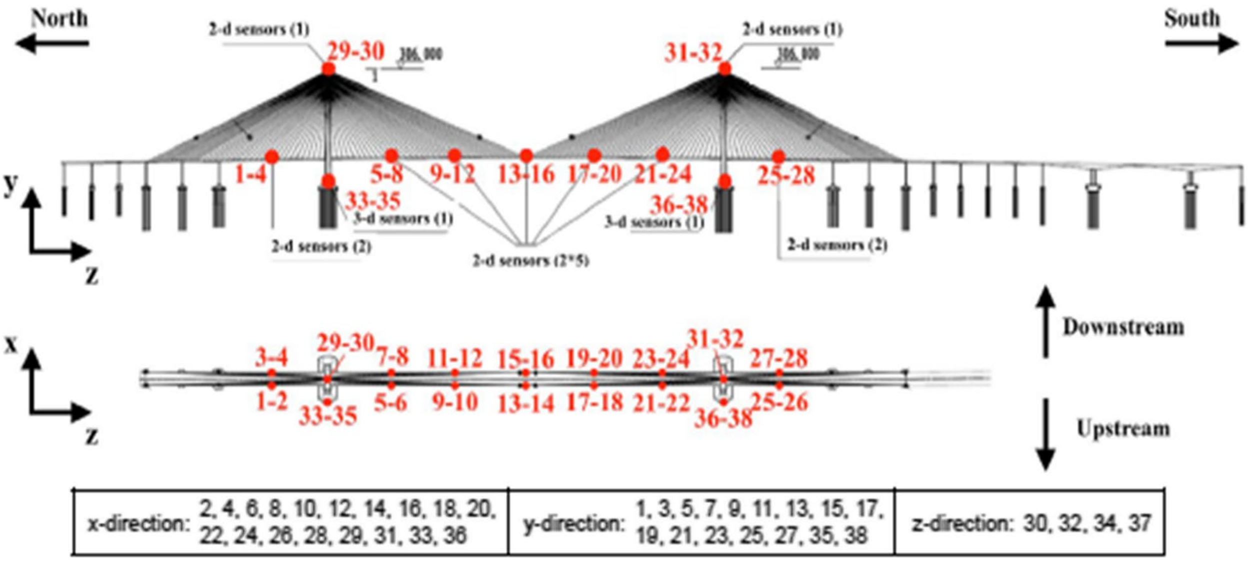


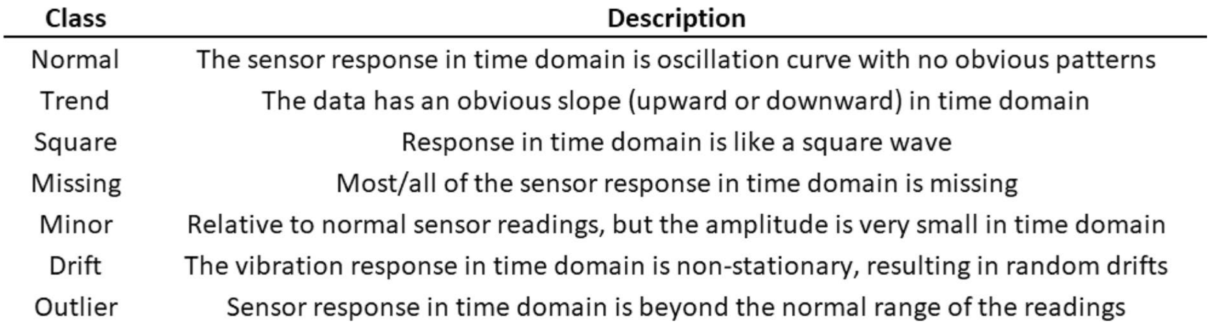
Fig. 3 The flowchart of the proposed CNN-MIAD method



x-direction:	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 29, 31, 33, 36	y-direction:	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 35, 38	z-direction:	30, 32, 34, 37

Fig. 4 The bridge and placement of the accelerometers on the deck and towers (IPC-SHM 2020)

Table 1 Description of each data class



Class	Description
Normal	The sensor response in time domain is oscillation curve with no obvious patterns
Trend	The data has an obvious slope (upward or downward) in time domain
Square	Response in time domain is like a square wave
Missing	Most/all of the sensor response in time domain is missing
Minor	Relative to normal sensor readings, but the amplitude is very small in time domain
Drift	The vibration response in time domain is non-stationary, resulting in random drifts
Outlier	Sensor response in time domain is beyond the normal range of the readings

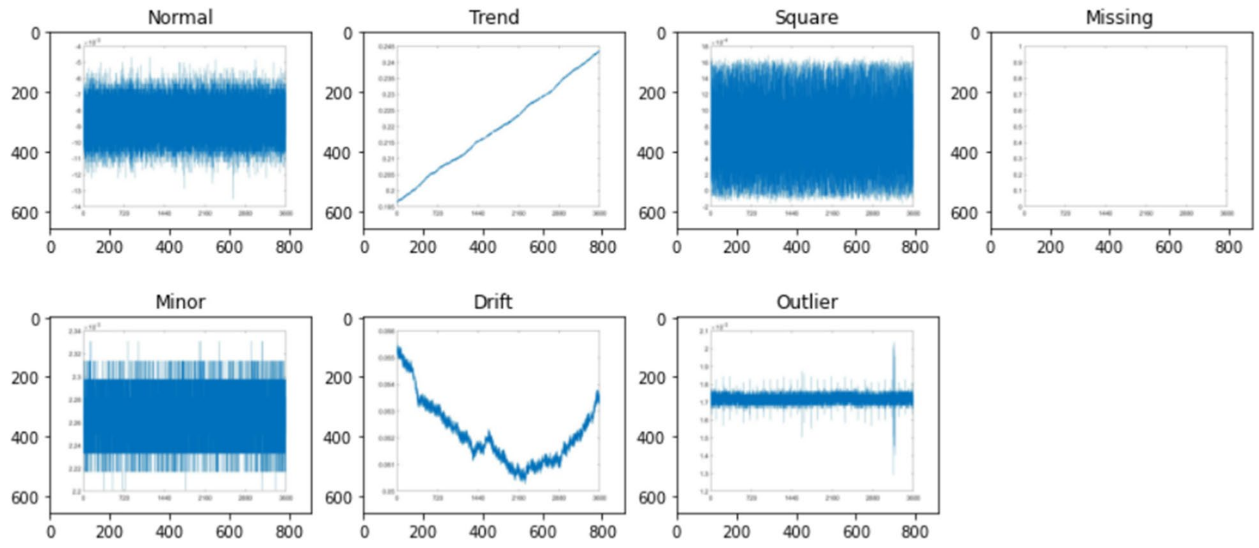


Fig. 5 Typical multiclass anomaly data

Table 2 Quantity and percentage for each data class of the selected benchmark data

Pattern	C1 Unbalanced dataset		C2 Balanced dataset		C3 Augmented balanced dataset	
	Quantity	%	Quantity	%	Quantity	%
Normal	13563	48.04	526	14.3	3000	14.3
Trend	5766	20.42	526	14.3	3000	14.3
Square	2992	10.60	526	14.3	3000	14.3
Missing	2933	10.39	526	14.3	3000	14.3
Minor	1775	6.29	526	14.3	3000	14.3
Drift	679	2.40	526	14.3	3000	14.3
Outlier	526	1.86	526	14.3	3000	14.3
Total	28234	100	3682	100	21000	100

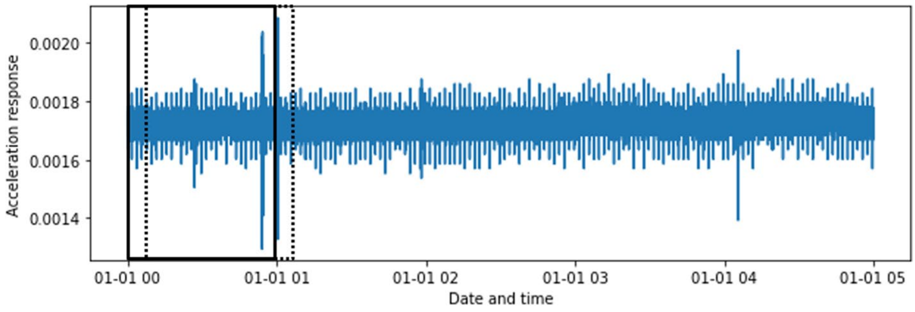


Fig. 6 Illustration of the time shift for C3

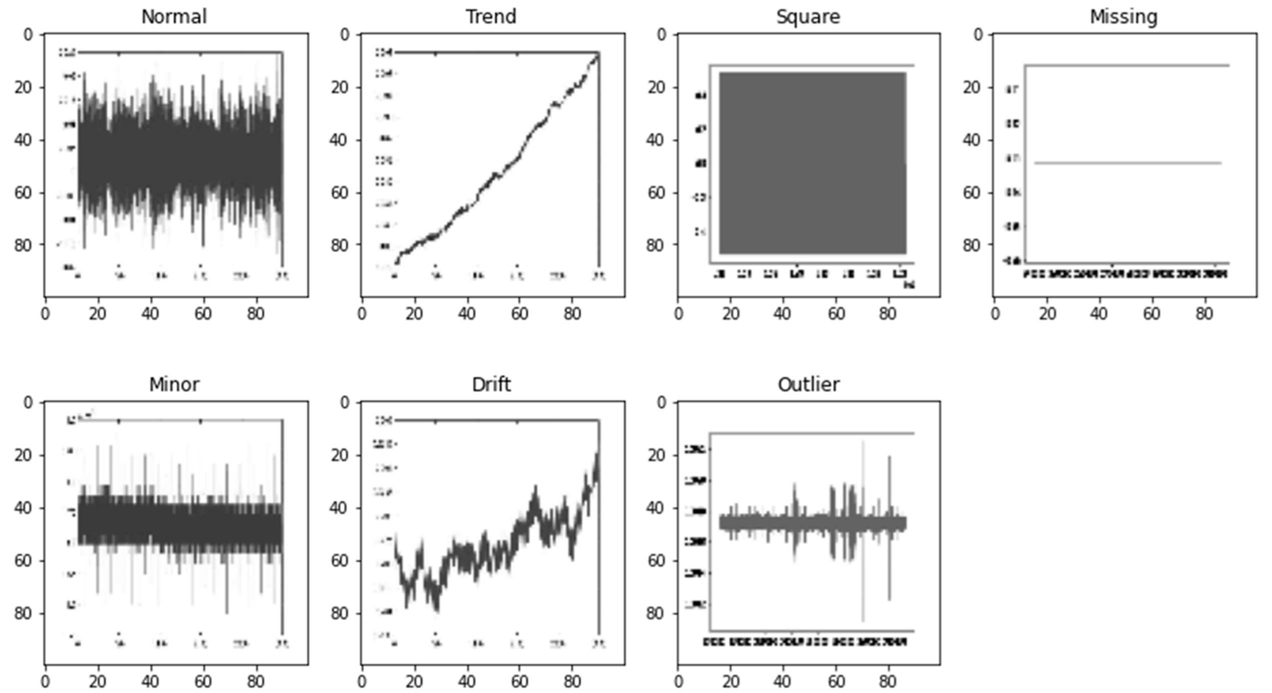
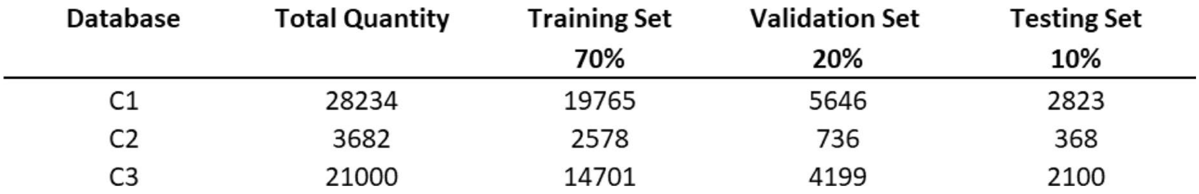


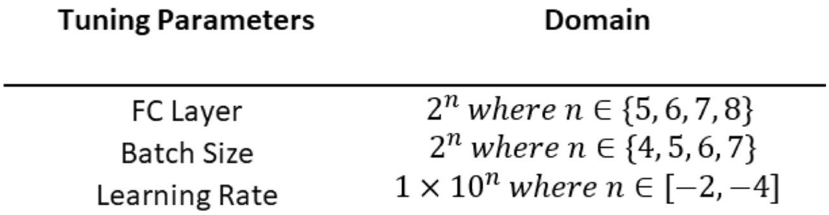
Fig. 7 Image samples for each class after preprocessing

Table 3 Ratio and quantity for the training, validation, and testing sets



Database	Total Quantity	Training Set	Validation Set	Testing Set
		70%	20%	10%
C1	28234	19765	5646	2823
C2	3682	2578	736	368
C3	21000	14701	4199	2100

Table 4 Domain for the hyperparameters



Tuning Parameters	Domain
FC Layer	$2^n \; where \; n \in \{5,6,7,8\}$
Batch Size	$2^n \; where \; n \in \{4,5,6,7\}$
Learning Rate	$1 \times 10^n \; where \; n \in [-2,-4]$

Data preparation and augmentation

The original dataset is imbalanced, with nearly 50% of the data in the Normal class and only ~2% data belonging to the Outlier class. Although this is always the case in practice, data augmentation techniques can be implemented as a preprocessing step before feeding the input to the proposed CNN-MIAD model. As part of the data preparation process after the collection of raw data, three cases have been created, as shown in Table 2, to verify the effects of data imbalance. Case 1 (C1) uses the

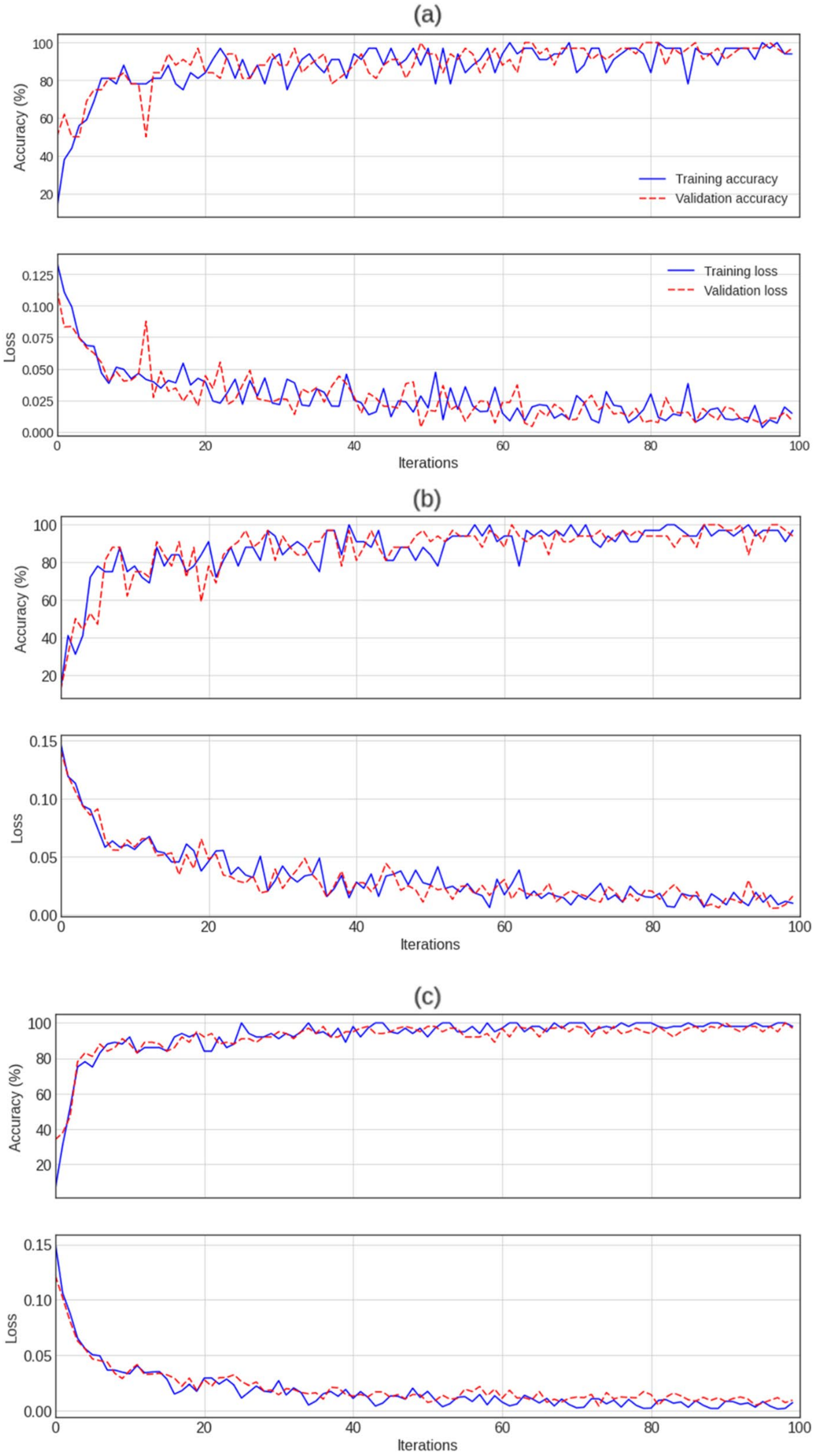
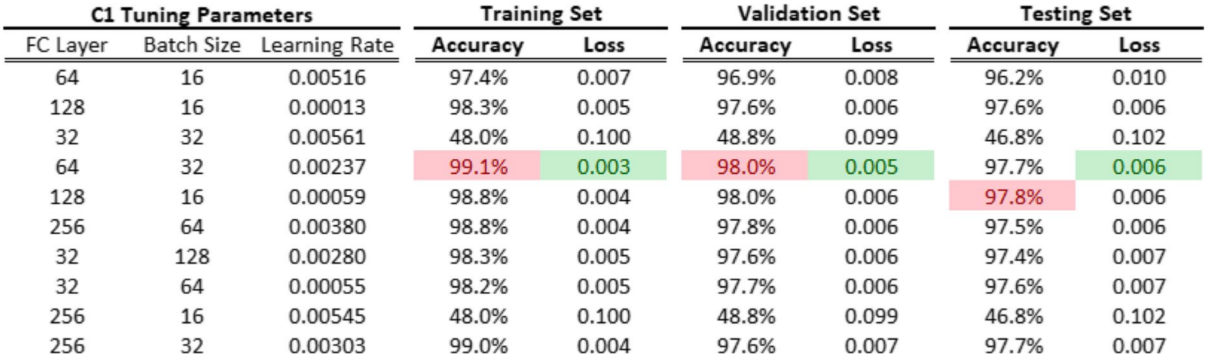


Fig. 8 Results of the proposed training using the optimally-tuned model: **a** C1, **b** C2 and **c** C3

Table 5 Hyperparameter tuning for C1



C1 Tuning Parameters			Training Set		Validation Set		Testing Set	
FC Layer	Batch Size	Learning Rate	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
64	16	0.00516	97.4%	0.007	96.9%	0.008	96.2%	0.010
128	16	0.00013	98.3%	0.005	97.6%	0.006	97.6%	0.006
32	32	0.00561	48.0%	0.100	48.8%	0.099	46.8%	0.102
64	32	0.00237	99.1%	0.003	98.0%	0.005	97.7%	0.006
128	16	0.00059	98.8%	0.004	98.0%	0.006	97.8%	0.006
256	64	0.00380	98.8%	0.004	97.8%	0.006	97.5%	0.006
32	128	0.00280	98.3%	0.005	97.6%	0.006	97.4%	0.007
32	64	0.00055	98.2%	0.005	97.7%	0.006	97.6%	0.007
256	16	0.00545	48.0%	0.100	48.8%	0.099	46.8%	0.102
256	32	0.00303	99.0%	0.004	97.6%	0.007	97.7%	0.007

original unbalanced full dataset as the model input. Since the least number of images for an input class is 526 (i.e., class Outlier), Case 2 (C2) uses a balanced dataset with randomly-selected 526 images in each class as the input dataset. However, this modification of balancing the dataset has significantly decreased the number of input images from 28,234 to 3682. To analyze the effect of the number of samples in the input dataset, Case 3 (C3) uses an augmented dataset with 3000 images in each class, resulting in a number of 21,000 input images. An illustration of the augmentation is included in Fig. 6, in which the accelerometer response for a single sensor is plotted against time. The hourly timeframe can only be joined together if it is consecutive, and the pattern label remains the same. The augmentation for C3 uses a sliding window of five minutes instead of one hour for the classes with less than 3000 images originally. Random selection is implemented to choose 3000 images from each class in the newly generated database, aiming to reduce the possibility of correlation and selection bias.

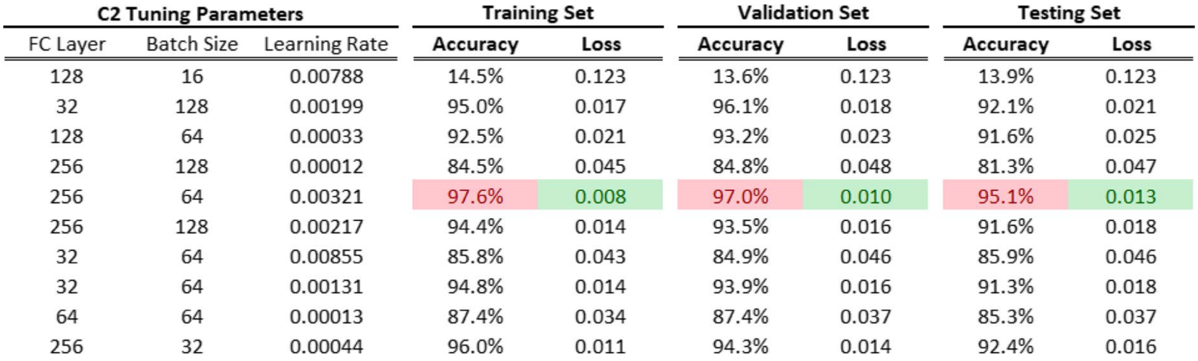
Before the original images are processed into the CNN-MIAD model, each image is resized from 875 × 656 pixels to 100 × 100 pixels and converted from RGB to greyscale, as shown in Fig. 7. This resizing of the image maintains the consistency of the entire database. Each database is split into training, validation, and testing sets based on a 70–20-10 split ratio, respectively. The ratio and quantity of images for each of the subsets in the database have been included in Table 3. In summary, the data preparation and augmentation steps include the conversion of time-series data into image representations, generating three cases using overlapping sliding windows and a random selection, changing dimensions, and splitting into training, validation, and testing sets.

Results

Hyperparameter tuning

The proposed CNN-MIAD model is trained with the aforementioned dataset in three cases, respectively. The random search algorithm is applied for the hyperparameter tuning process. The tuning parameters include the number of neurons in the second fully connected layer, the batch size, and the learning rate. The domain for each of the hyperparameters is included in Table 4. For each of the three cases, ten different hyperparameters are randomly generated. The trial with the highest training accuracy and lowest validation loss is defined as the best hyperparameter combination. The training process of the proposed CNN-MIAD model with the best hyperparameter combination for each of the three cases is shown in Fig. 8. The detailed results of the tuning process for C1, C2, and C3 are included in Tables 5, 6, and 7, respectively.

Table 6 Hyperparameter tuning for C2



C2 Tuning Parameters			Training Set		Validation Set		Testing Set	
FC Layer	Batch Size	Learning Rate	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
128	16	0.00788	14.5%	0.123	13.6%	0.123	13.9%	0.123
32	128	0.00199	95.0%	0.017	96.1%	0.018	92.1%	0.021
128	64	0.00033	92.5%	0.021	93.2%	0.023	91.6%	0.025
256	128	0.00012	84.5%	0.045	84.8%	0.048	81.3%	0.047
256	64	0.00321	97.6%	0.008	97.0%	0.010	95.1%	0.013
256	128	0.00217	94.4%	0.014	93.5%	0.016	91.6%	0.018
32	64	0.00855	85.8%	0.043	84.9%	0.046	85.9%	0.046
32	64	0.00131	94.8%	0.014	93.9%	0.016	91.3%	0.018
64	64	0.00013	87.4%	0.034	87.4%	0.037	85.3%	0.037
256	32	0.00044	96.0%	0.011	94.3%	0.014	92.4%	0.016

Table 7 Hyperparameter tuning for C3

C3 Tuning Parameters			Training Set		Validation Set		Testing Set	
FC Layer	Batch Size	Learning Rate	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
256	128	0.00341	99.1%	0.004	98.3%	0.005	97.7%	0.006
32	16	0.00059	99.2%	0.003	98.5%	0.005	97.5%	0.007
64	16	0.00116	99.3%	0.003	98.6%	0.004	98.0%	0.006
32	128	0.00140	98.5%	0.005	98.1%	0.006	96.9%	0.008
64	16	0.00074	99.5%	0.003	98.5%	0.005	97.7%	0.006
128	32	0.00055	99.3%	0.003	98.4%	0.005	97.5%	0.006
32	64	0.00097	98.4%	0.005	98.0%	0.006	97.1%	0.008
128	64	0.00014	97.3%	0.008	97.3%	0.009	96.2%	0.011
256	32	0.00051	99.6%	0.002	98.6%	0.004	98.1%	0.005
32	64	0.00043	97.6%	0.007	97.3%	0.008	95.9%	0.010

Performance evaluation

To evaluate the performance of the proposed CNN-MIAD model, five repetitions are generated using the best-tuned parameters for each of the three cases. As shown in Table 8, the overall accuracies and losses are calculated using the mean value for all five trials in each case. The training and testing accuracies are greater than 90% in all fifteen trials, meaning that the proposed model is free of underfitting. On the other hand, the testing accuracies and losses are close to the training results for each of the respective trials, meaning that the proposed model is not overfitting the training set. To further understand the effects of balancing the database and the number of input image samples on the model output, the confusion matrix against the fifth trial of the testing set of each case has been included in Fig. 9. Using the confusion matrix for each case, the precision, recall and *F1* score for C1, C2, and C3 are included in Table 9.

All three cases are trained on a Linux server with 24 Intel(R) Xeon(R) E5–2630 v2 processor. The training time for each random search model is around 130 minutes, 65 minutes, and 110 minutes for C1, C2, and C3, respectively. As observed by comparing C1 and C2 results, a database with more data points will generate a higher accuracy. The number of data samples in C1 and C2 are 28,234 and 3682, respectively. The accuracy of the model decreased from 97.70% to 93.59% as the number of image samples decreased. The *F1* score for classes Normal and Minor has decreased drastically due to the smaller sample size for the training set. Although both C2 and C3 are balanced databases, the overall accuracy for C3 has increased from 93.59% in C2 to 97.74% due to the larger sample size. As seen through C1 and C3 results, although the overall accuracy is similar, the *F1* scores for the Outlier and Drift classes have been remarkably improved

Table 8 Overall accuracy and loss of three cases with the optimally-tuned hyperparameter

Database	Tuning Parameters			Trial	Training Set		Validation Set		Testing Set	
	FC Layer	Batch Size	Learning Rate		Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
C1	64	32	0.00237	1	99.1%	0.003	98.0%	0.005	97.7%	0.006
				2	98.8%	0.005	98.1%	0.006	97.6%	0.007
				3	98.4%	0.005	97.9%	0.006	97.6%	0.007
				4	98.9%	0.004	98.0%	0.006	97.6%	0.007
				5	99.2%	0.003	98.1%	0.006	98.0%	0.006
				Overall	98.9%	0.004	98.0%	0.006	97.7%	0.007
C2	256	64	0.00321	1	97.6%	0.008	97.0%	0.010	95.1%	0.013
				2	98.2%	0.008	96.3%	0.012	93.5%	0.017
				3	96.2%	0.011	94.0%	0.016	93.5%	0.017
				4	98.6%	0.006	95.8%	0.011	93.2%	0.016
				5	98.3%	0.007	95.4%	0.013	92.7%	0.016
				Overall	97.8%	0.008	95.7%	0.012	93.6%	0.016
C3	256	32	0.00051	1	99.6%	0.002	98.6%	0.004	98.1%	0.005
				2	99.2%	0.002	98.5%	0.004	97.7%	0.005
				3	99.4%	0.003	98.6%	0.005	97.9%	0.006
				4	99.2%	0.003	98.3%	0.004	97.5%	0.006
				5	99.1%	0.003	98.4%	0.005	97.6%	0.006
				Overall	99.3%	0.003	98.5%	0.004	97.7%	0.006

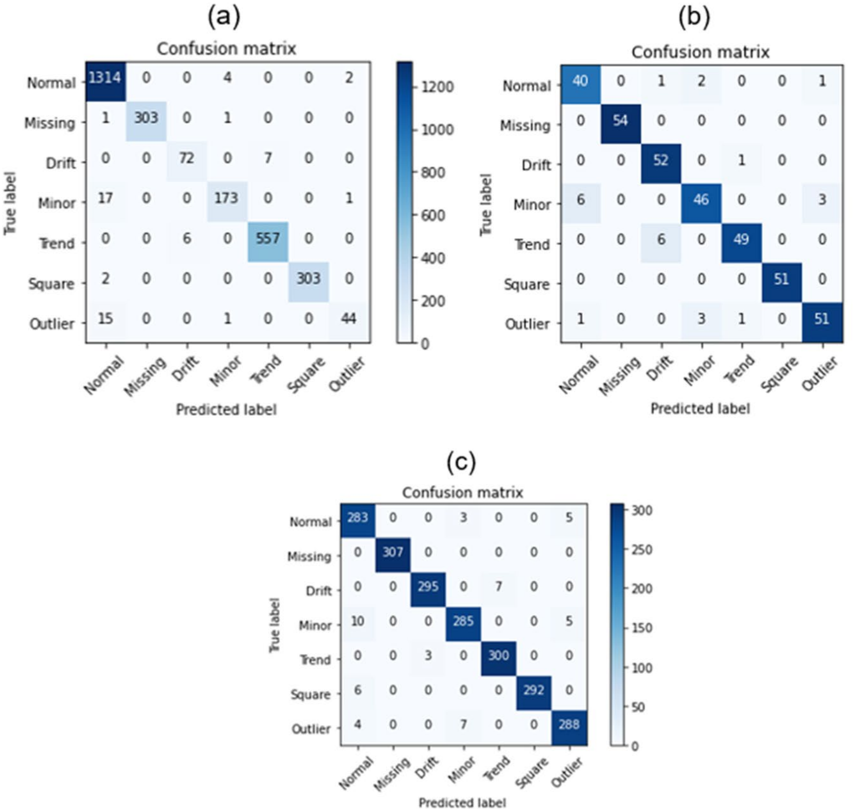


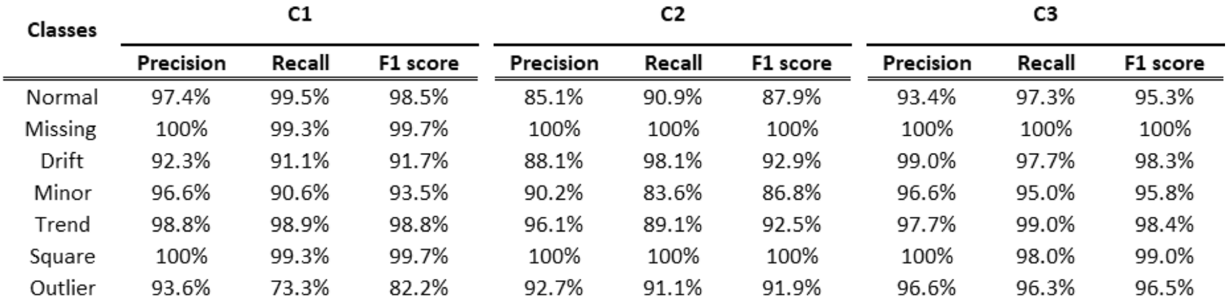
Fig. 9 Confusion matrix against the fifth trial testing set: **a** C1, **b** C2, and **c** C3

from 82.24% and 91.72% to 96.48% and 98.33%, respectively. Since the number of image samples in both cases is similar, this improvement proves that balancing the database could improve the performance of the classes with fewer image samples originally.

Conclusions

This paper proposes a deep learning algorithm for the AD tasks within the realm of SHM. The proposed CNN-MIAD model requires the conversion of time series data into greyscale images, which then pass through several layers with hyperparameter tuning techniques to generate the best-performing model. The proposed CNN-MIAD model is tested with three cases using a database generated by a long-span cable-stayed bridge in China. The three cases are the original unbalanced database, balancing the database using the smallest number of images in a class, and balancing the database with augmented samples. The results reveal that balancing the database allows to improve the F1 score of the classes with fewer image samples originally, and a database with more image samples will increase the overall accuracy of the model performance. It concludes that augmenting the balanced database leads to the best results, with a

Table 9 Precision, Recall and *F1* score for each case



Classes	C1			C2			C3		
	Precision	Recall	F1 score	Precision	Recall	F1 score	Precision	Recall	F1 score
Normal	97.4%	99.5%	98.5%	85.1%	90.9%	87.9%	93.4%	97.3%	95.3%
Missing	100%	99.3%	99.7%	100%	100%	100%	100%	100%	100%
Drift	92.3%	91.1%	91.7%	88.1%	98.1%	92.9%	99.0%	97.7%	98.3%
Minor	96.6%	90.6%	93.5%	90.2%	83.6%	86.8%	96.6%	95.0%	95.8%
Trend	98.8%	98.9%	98.8%	96.1%	89.1%	92.5%	97.7%	99.0%	98.4%
Square	100%	99.3%	99.7%	100%	100%	100%	100%	98.0%	99.0%
Outlier	93.6%	73.3%	82.2%	92.7%	91.1%	91.9%	96.6%	96.3%	96.5%

mean overall accuracy of 97.74% across five trials on the testing set.

The application for the proposed CNN-MIAD model is not restricted to the realm of SHM but could be applied to classification problems containing time-series data as raw data points. The complete flow of the model requires standardizing the input into a grayscale image database before passing it into the CNN-MIAD model. With sufficient raw data preprocessing, the CNN-MIAD model does not need modifications as the images are passing through. Incorporating a hyperparameter tuning process to categorize different classes satisfies the needs of a deep learning algorithm to achieve optimal results. Using the random search algorithm to tune the hyperparameters is more efficient by setting up the domain for each tuning parameter. The proven success of the CNN-MIAD model in SHM demonstrates the potential to expand its use in other imbalance dataset classification applications.

Acknowledgements

The authors would like to acknowledge the 1st International Project Competition for Structural Health Monitoring (IPC-SHM, 2020), organized by the Asia-Pacific Network of Centers for Research in Smart Structures Technology (ANCRiSST), Harbin Institute of Technology, the University of Illinois at Urbana-Champaign, and four other institutions, for sharing the benchmark data used in this paper. The authors would like to thank the Dean's award provided to the first author by the Faculty of Engineering at Western University. This research was also supported through the corresponding author's Early Researcher Award provided by the Ontario Ministry of Colleges and Universities.

Authors' contributions

MZ conceived the proposed study, conducted a detailed literature review for the paper and completed the proposed research. MZ participated in the research design, analyzed, and interpreted the data of the full-scale study. MZ drafted the manuscript, AS contributed to the write-up, and both AS and MC provided a detailed review of the manuscript. AS and MC supervised the entire research. All authors approved the manuscript for publication in the journal.

Funding

This research was financially supported by the Dean's award provided to the first author by the Faculty of Engineering at Western University. In addition, this research was supported by the corresponding author's Early Researcher Award provided by the Ontario Ministry of Colleges and Universities.

Availability of data and materials

The datasets used in this study were available open-source.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

All authors have consented to the publication of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Electrical and Computer Engineering, Western University, London, Ontario, Canada. [2]Department of Civil and Environmental Engineering, Western University, London, Ontario, Canada.

Received: 14 June 2022 Accepted: 19 July 2022
Published online: 03 August 2022

References

1. Almasri N, Sadhu A, Ray Chaudhuri S (2020) Towards compressed sensing of structural monitoring data using discrete cosine transform. ASCE J Comput Eng 34(1):04019041
2. Arul, M. and Kareem, A (2020) Data anomaly detection for structural health monitoring of bridges using Shapelet transform. ArXiv
3. Bao Y, Chen Z, Wei S, Xu Y, Tang Z, Li H (2019) The state of the art of data science and engineering in structural health monitoring. Engineering. 5(2):234–242
4. Bao Y, Tang Z, Li H, Zhang Y (2018) Computer vision and deep learning-based data anomaly detection method for structural health monitoring. Structur Health Monit 18(2):401–421
5. Barbosh M, Sadhu A, Sankar G (2021) Time-frequency decomposition assisted improved localization of proximity of damage using acoustic sensors. Smart Mater Struct 30(2):025021
6. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(10):281–305
7. Cawley P (2018) Structural health monitoring: closing the gap between research and industrial deployment. Struct Health Monit 17(5):1225–1244
8. Chen J, Pi D, Wu Z, Zhao X, Pan Y, Zhang Q (2021) Imbalanced satellite telemetry data anomaly detection model based on Bayesian LSTM. Acta Astronautica 180:232–242
9. Ciresan D, Meier U, Masci J, Gambardella L, Schmidhuber J (2011) Flexible, high-performance convolutional neural networks for image classification. International Joint Conference on Artificial Intelligence IJCAI–2011
10. Das S, Saha P (2018) A review of some advanced sensors used for health diagnosis of civil engineering structures. Measurement 129:68–90
11. Debnath, A., Waghmare, G., Wadhwa, H., Asthana, S. and Arora, A. (2021). Exploring generative data augmentation in multivariate time series forecasting : opportunities and challenges. MileTS'21: 7th KDD Workshop on Mining and Learning from Time Series
12. Delgado J, Oyedele L (2019) Deep learning with small datasets: using autoencoders to address limited datasets in construction management. Appl Soft Comput 112:107836
13. Dong C, Catbas N (2020) A review of computer vision-based structural health monitoring at local and global levels. Struct Health Monit 20(2):692–743
14. Duman E, Tolan Z (2021) Comparing popular CNN models for an imbalanced dataset of dermoscopic images. J Comput Science IDAP-2021:192–207
15. Galar, M., Fernandez, A., Barrenechea, E., Bustince H. and Herrera, F. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 42(4)
16. Heaton, J. (2008). Introduction to neural networks with Java. Heaton Research, Inc. Second Edition. Pg 158
17. Hutter, F., Kotthoff, L. and Vanschoren, J. (2019). Automated machine learning: methods, systems, challenges. The springer series on challenges in machine learning. Chapter 1
18. Islam A, Belhaouari SB, Rehman AU, Bensmail H (2022) KNNOR: an oversampling technique for imbalanced datasets. Appl Soft Comput 115:108288
19. Jain M, Kaur G, Saxena V (2022) A K-means clustering and SVM based hybrid concept drift detection technique for network anomaly detection. Expert Syst Appl 193:116510
20. Jana D, Patil J, Herkal S, Nagarajaiah S, Duenas-Osorio L (2022) CNN and convolutional autoencoder (CAE) based real-time sensor fault detection, localization, and correction. Mech Syst Signal Process 169:108723
21. Jeong S, Ferguson M, Law K (2019) Sensor data reconstruction and anomaly detection using bidirectional recurrent neural network. Sensors Smart Struct Technol Civil Mechan Aerospace Syst 2019:10970
22. Kuhn M, Johnson K (2016) Applied predictive modelling. Springer Nature
23. Kullaa J (2013) Detection, identification, and quantification of sensor fault in a sensor network. Mech Syst Signal Process 40:208–221

24. Kaartinen E, Dunphy K, Sadhu A (2022) LiDAR-based structural health monitoring: applications in civil infrastructure systems. Sensors. 22(8):4610
25. Li H, Ou J (2016) The state-of-the-art in structural health monitoring of cable-stayed bridges. J Civ Struct Heal Monit 6(1):43–67
26. Liu G, Niu Y, Zhao W, Duan Y, Shu J (2022) Data anomaly detection for structural health monitoring using a combination network of GANomaly and CNN. Smart Struct Syst 29(1):5362
27. Maes K, Salens W, Feremans G, Segher K, François S (2022) Anomaly detection in long-term tunnel deformation monitoring. Eng Struct 250:113383
28. Mao J, Wang H, Spencer B (2020) Toward data anomaly detection for automated structural health monitoring: exploiting generative adversarial nets and autoencoders. Struct Health Monit 20(4):1609–1626
29. Montgomery D (2013) Design and analysis of experiments, 8th edn. John Wiley & Sons, Inc. Chapter 5
30. Neves A, Gonzalez I, Leander J, Karoumi R (2017) Structural health monitoring of bridges: a model-free ANN-based approach to damage detection. J Civ Struct Heal Monit 7(5)
31. Ni F, Zhang J, Noori M (2019) Deep learning for data anomaly detection and data compression of a long-span suspension bridge. Comput Aid Civil Infrastruct Eng 35(7):685–700
32. Panda B. (2019). A survey on application of population-based algorithm on Hyperparameter selection
33. Phung VH, Rhee EJ (2018) A deep learning approach for classification of cloud image patches on small datasets. J Inform Commun Converg Eng 16(3):173–178
34. Ranyal E, Sadhu A, Jain K (2022) Road condition monitoring using smart sensing and artificial intelligence: a review. Sensors. 22(8):3044
35. Rawat W, Wang Z (2017) Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput 29(9):2352–2449
36. Sai B, Anita CS, Rajalakshmi D, Berlin MA (2021) A CNN-based facial expression recognizer. Materials Today 37(2):2578–2581
37. Sarmadi H, Karamodin A (2020) Novel anomaly detection method based on adaptive Mahalanobis-squared distance and one-class kNN rule for structural health monitoring under environmental effects. Mech Syst Signal Process 140:106495
38. Scherer, D., Müller, A. and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. 20th International Conference on Artificial Neural Networks
39. Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2016) Taking the human out of the loop: a review of Bayesian optimization. Proc IEEE 104(1)
40. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(60)
41. Singh P, Sadhu A (2022) A hybrid time-frequency method for robust drive-by modal identification of bridges. Eng Struct 266:114624
42. Singh P, Keyvanlou M, Sadhu A (2021) An improved time-varying empirical mode decomposition for structural condition assessment using limited sensors. Eng Struct 232:111882
43. Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. ArXiv
44. Sony S, Dunphy K, Sadhu A, Capretz M (2021) A systematic review of convolutional neural network-based structural condition assessment techniques. Eng Struct 226:111347
45. Sony S, Laventure S, Sadhu A (2019) A literature review of next-generation smart sensing technology in structural health monitoring. Struct Control Health Monit 26(3):e2321
46. Steiner, M., Fritz, H., Thiebes, L. and Dragos, K. (2018). Fault diagnosis in wireless structural health monitoring systems based on support vector regression
47. Tang Z, Chen Z, Bao Y, Li H (2018) Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. Struct Control Health Monit 26(1):e2296
48. Thiyagarajan, K., Kodagoda, S. and Nguyen, L. (2017). Predictive analytics for detecting sensor failure using autoregressive integrated moving average model. 12th IEEE Conference on Industrial Electronics and Applications
49. Tian Y, Zhang Y (2022) A comprehensive survey on regularization strategies in machine learning. Inform Fusion 80:146–166
50. Wang H, Li L, Song G, Dabney JB, Harman TL (2015) A new approach to deal with sensor errors in structural controls with MR damper. Smart Struct Syst 16(2)
51. Wang Y, Zhu Y, Lou G, Zhang P, Chen J, Li J (2021) A maintenance hemodialysis mortality prediction model based on anomaly detection using longitudinal hemodialysis data. J Biomed Inform 123:103930
52. Wedel F, Marx S (2022) Application of machine learning methods on real bridge monitoring data. Eng Struct 250:113365
53. Yang J, Yang F, Zhang L, Li R, Jiang S, Wang G, Zhang L, Zeng Z (2021) Bridge health anomaly detection using deep support vector data description. Struct Health Monit 20(6):170–178
54. Zeiler M (2012) Adadelta: an adaptive learning rate method. ArXiv
55. Zhang W, Li C, Peng G, Chen Y, Zhang Z (2018) A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. Mech Syst Signal Process 100:439–453
56. Zhu Y, Ni Y, Jin H, Inaudi D, Laory I (2019) A temperature-driven MPCA method for structural anomaly detection. Eng Struct 190:447–458

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.