Engineering Structures 292 (2023) 116473

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct

A novel multi-level data fusion and anomaly detection approach for infrastructure damage identification and localisation

Hao Wang [a,*], Giorgio Barone [b], Alister Smith [a]

[a] School of Architecture, Building and Civil Engineering, Sir Frank Gibb Building, Loughborough University, Loughborough LE11 3TU, United Kingdom
[b] Formerly School of Architecture, Building and Civil Engineering, Sir Frank Gibb Building, Loughborough University, Loughborough LE11 3TU, United Kingdom

ARTICLE INFO

Keywords:
Data fusion
Machine learning
Anomaly detection
Damage identification
Damage localisation
Infrastructure health monitoring
Smart infrastructure

ABSTRACT

In this paper, a new damage identification and localisation framework utilising multi-level data fusion and anomaly detection techniques is introduced. The performance of the framework is presented via case studies of a simply supported bridge and a continuous bridge, whose data were numerically generated through a surrogate model. Accelerations, deflections and bending moments obtained at multiple sensor locations when the bridge was subjected to a moving vehicle were used as the input. A damage-sensitive feature was established via coupling principal component analysis and Mahalanobis distance, allowing for initial data dimensionality reduction and information integration. Anomaly detection using a deep convolutional autoencoder was performed to identify the presence of single and multiple damages on the bridge. It is demonstrated that the proposed approach is independent of the mass and speed of moving vehicles, and is applicable to different types of bridges (i.e. simply supported bridge and continuous bridge) with only minor modifications to the data augmentation/preparation processes. The accuracy of damage identification for the simply supported bridge is shown to be consistently greater than 93%, even for extremely small damage severity (e.g. 1% reduction in local stiffness), and multiple damages. While the damage identification accuracy for the continuous bridge is greater than 94.9% for the majority of the tested cases. The proposed damage localisation approach can accurately determine damage location(s) for both single and multiple damage cases. It is shown how the proposed framework can be adopted for early damage detection to enable reliable decision-making and maximise structural safety.

1. Introduction

Infrastructure is vital to the functioning of society. Damage and performance deterioration can endanger the safety and serviceability of infrastructure, causing significant economic losses [1]. Global challenges, such as climate change and growing population, are accelerating the rate of deterioration of existing infrastructure, increasing susceptibility to damage. Hence, it is critical to monitor the health condition of infrastructure assets throughout their lifecycle to provide information crucial to maintenance programmes and safety–critical decision-making processes.

As a result of the recent technological advances in sensors, and data analysis and interpretation approaches, modern infrastructure is often monitored using complex systems of sensors providing real-time data for health condition assessment [2–4]. Traditional vibration-based techniques exploit the inherent structural characteristics (e.g. mass, stiffness and damping) to detect damages via system identification and/or model updating [5–7]. System identification relies on the identification accuracy of higher-order dynamic characteristics, while model updating draws inferences from analytical or numerical models, which are typically idealised and based on significant assumptions.

In the past decade, a suite of automated damage identification algorithms has been developed alongside machine learning (ML) techniques because of their efficiency and ability to handle large datasets [8–10]. These damage identification algorithms employ ML techniques to recognise changes in monitoring data patterns between undamaged and damaged states, which enables the presence of damage to be identified. The most widely used ML techniques in infrastructure health monitoring can be broadly divided into two categories: supervised and unsupervised learning. Supervised learning techniques model the relationships between the input features and their corresponding target

* Corresponding author at: School of Architecture, Building and Civil Engineering, Sir Frank Gibb Building, Loughborough University, Loughborough LE11 3TU, United Kingdom.

E-mail addresses: H.Wang4@lboro.ac.uk, Hao.Wang@atkinsglobal.com (H. Wang).

https://doi.org/10.1016/j.engstruct.2023.116473
Received 18 November 2022; Received in revised form 17 May 2023; Accepted 9 June 2023
Available online 7 July 2023
0141-0296/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Nomenclature

Symbol	*Meaning*
AEs	Autoencoders
ANN	Artificial neural network
CNN	Convolutional neural networks
Conv1D	1D convolution layers
Conv1D_Transpose	Transposed 1D convolution layers
DCAE	Deep convolutional autoencoder
FP	False positive
FN	False negative
MAE	Mean absolute error
MD	Mahalanobis distance
ML	Machine learning
MDR_i	The individual value of the MD ratio
$\bar{MDR}^2$	The square of the average MD ratio
MDR_T	The transformed MD ratio.
Nadam	Nesterov-accelerated adaptive moment estimation
P	Projection of the data matrix **X**
PCA	Principal component analysis
PCs	Principal components
ReLU	Rectified linear unit
SNR	Signal-noise ratio
TN	True negative
TP	True positive
W	Eigenvectors matrix
a & b	Weight parameter and bias
C	Covariance matrix
$\mathbf{C}^{-1}$	The inverse of the covariance matrix of **y**
$f(\cdot)$	Activation function
k	The k-th input data
p	The p-th neuron of a neural network layer
$\mathbf{u_i}$ & $\mathbf{v_i}$	Coordinates of data points
V	Eigenvectors of the covariance matrix
$x_{i\alpha}$&$x_{i\beta}$	Variables at the α-th and v-th column of β-th row of the matrix $\mathbf{X}_{m\times n}$
$\bar{\mathbf{x}}$	Mean of the data matrix **X**
$\mathbf{X_{m\times n}}$	Matrix of time-series data
y	Feature matrix obtained from PCA
$\bar{y}$	Mean vectors of the matrix **y**
$\tilde{\mathbf{y}}$	Median of the matrix **y**
$\hat{\mathbf{z}}_\mathbf{j}$	The reconstruction of the *j* th input
$\mathbf{z_j}$	The actual value of the *j* th input
Λ	Eigenvalues of the covariance matrix

labels/variables. The major limitation of supervised learning techniques is that they cannot provide information beyond the learned knowledge. In the context of infrastructure health monitoring, supervised learning techniques require an equal or similar number of samples from both undamaged and damaged cases [11]. However, damage information is typically unavailable, and labelling data can be expensive and time-consuming.

By contrast, unsupervised learning techniques attempt to identify the structure of the input features without any information about target labels/variables [11]. These techniques can draw inferences on unseen data using the knowledge learned during the training process. Anomaly detection is one of the most popular unsupervised learning techniques in the infrastructure health monitoring community. Anomaly detection algorithms are usually trained solely using data from the undamaged state, and identify damage as data points deviating from expected behaviour. A variety of techniques can be used for anomaly detection, such as distance metrics [12], control charts [13,14], data reconstruction [15] and one-class algorithms [16].

Autoencoders (AEs) are neural networks with bottleneck structures, and they are one of the most widely used anomaly detection techniques, due to their abilities in feature extraction and dimensionality reduction [15,17]. AEs reconstruct input features to identify the presence of damage, which is achieved by identifying data samples as anomalies when the reconstruction error exceeds a threshold that was automatically derived during the training process. AEs implemented by convolutional layers have sparsely connected convolutional neurons among the neural networks, which considerably improves the computational efficiency compared with the fully connected ones [18]. The features extracted via convolutional neural networks (CNN) have been proven to be highly sensitive to the presence of damages [13,19].

In recent years, data fusion techniques that enable the integration of information from multiple sources to improve damage diagnostic procedures have attained an increasing amount of interest [20,21]. Health monitoring systems typically comprise multiple types of sensors to acquire a suite of parameters at different spatial and temporal resolutions. Data fusion techniques refine data acquired from multiple sources during a series of data processing stages: data-level, feature-level and decision-level [22]. Data-level fusion synthesises raw data from multiple sources to achieve more useful information. The feature-level fusion process results in an improved representation (i.e. feature) of the raw data. Decision-level fusion results in a unique decision, considering multiple hypotheses, which improves decision-making performance and robustness [23–26]. Data fusion can be achieved using various techniques, such as dimensionality reduction [27], feature extraction [28], clustering techniques [29] and distance metrics [30].

Principal component analysis (PCA) is the most popular unsupervised learning technique for dimensionality reduction. PCA and its variants have been used for several purposes in infrastructure health monitoring, such as data cleaning [31,32], feature extraction [33], data fusion [34] and damage detection [35]. PCA is a statistical technique that projects the input data into a lower-dimensional space while retaining the variance present in the original data as much as possible. Correlated variables that do not contribute to decision-making are removed, which improves the performance of the subsequent data analysis algorithms. However, it is notable that if the number of principal components (PCs) is not properly selected, some valuable information can be lost.

Calculation of the distance metrics among features is the simplest way to implement data fusion. Mahalanobis distance (MD) is one of the commonly-used metrics, which considers the correlations between variables as well as the differences in variances. MD gives less weight to highly correlated variables and variables with high variance, so that all characteristics are treated as equally important. In high-dimensional spaces, data points are normally sparse and all the data points are almost equidistant. Some distance measures, such as Euclidean distance, are meaningless in these scenarios. The combination of PCA and MD has been used in the infrastructure health monitoring community [12,33,36].

The main objective of this study was to develop an automated approach for bridge damage identification and localisation using accelerations, deflections and bending moments measured at multiple positions in response to moving vehicles. Numerical modelling of a simply supported bridge and a continuous bridge under simulated traffic loading were performed to generate the synthetic data. The developed approach combines multi-level data fusion and anomaly detection techniques to detect and locate damage(s). At the data level, PCA was used to fuse the raw data acquired from multiple sensors and extract initial features. Subsequently, feature-level fusion was implemented using MD as the damage-sensitive feature. A convolutional AE was trained using the MD calculated from data measured solely on the

undamaged structure, since these would be the only data available for a real bridge case (i.e. before damage has occurred). The presence of damage was then detected as anomalies in the reconstruction of unseen data via the trained AE. The novelty of this work is that the damage(s) are automatically identified and located under the operating condition (i.e. vehicle loading) of the bridges and in an unsupervised manner, while the approach can be applied to different types of bridges with only minor modifications to the data augmentation/preparation processes. The approach developed can successfully detect small damage severities (e.g. reductions in local stiffness as small as 1%). A variety of single and multiple damage scenarios were investigated to demonstrate the potential of the proposed approach.

2. Surrogate model of a simply supported bridge

Fig. 1 shows the numerical model of a simply supported bridge with a 30 m span and with a moving vehicle, simulated using the commercial finite element software SAP2000. The span was divided into 60 equal-sized finite elements. The model parameters are listed in Table 1.

Accelerations, deflections and bending moments were selected as inputs for the anomaly detection algorithm. Acceleration, deformation and strain are the most commonly-used parameters in infrastructure health monitoring [37,38]. Bending moments can be obtained from strain measurements when the cross-section geometry and elastic properties are known, and they have been used for damage detection in the research literature (e.g. influence line method) [39]. Accelerations, deflections and bending moments at the nodes corresponding to nine equidistant sensor locations (Fig. 1) were measured at a sampling rate of 200 Hz.

Time histories of the structural response of the bridge when subjected to a moving vehicle with constant velocity have been simulated (Fig. 1). The vehicle speed was varied in the range of 20 to 80 km/h, with intervals of 2 km/h, for a total of 31 possible cases. A time window of 15 seconds was employed for data acquisition during each event, to capture the structural responses while the vehicle is passing over the bridge, as well as during damped oscillations.

Table 1
Parameters of the surrogate model.

Material	C30/37 Concrete
Density	2500 kg/m^3
Young's Modulus	32 GPa
Damping Ratio	0.05
Cross-Section Area	0.4 m^2
Second Moment of Area	0.0333 m^4
Element Type	Beam Element
Type of Analysis	Time-History Analysis

A variety of single and multiple damage scenarios were simulated. Each scenario is characterised by localised damage on the bridge, which resulted in discontinuities in stiffness. Damage was simulated using a reduced second moment of area in one or more selected elements. Table 2 provides a summary of the cases investigated, showing the

Table 2
Cases used in this study.

Case	Vehicle Mass (kN)	Damaged Element(s)	Relative Damage Location (RDL = X/L)	Damage Severity
Baseline	10	–	–	–
I	20	–	–	–
II	10	7	0.1~0.12	a) 1%; b) 5%; c) 30%
III	10	13	0.2~0.22	a) 1%; b) 5%; c) 30%
IV	10	19	0.3~0.32	a) 1%; b) 5%; c) 30%
V	10	37	0.6~0.62	a) 1%; b) 5%; c) 30%
VI	10	13 & 37	0.2~0.22 & 0.6~0.62	a) 1% & 1%; b) 5% & 5%; c) 30% & 30%
VII	10	13 & 37	0.2~0.22 & 0.6~0.62	a) 30% & 1%; b) 1% & 5%

Note: X is the distance of the damage location from the left-hand side support; L is the length of the simply supported bridge.

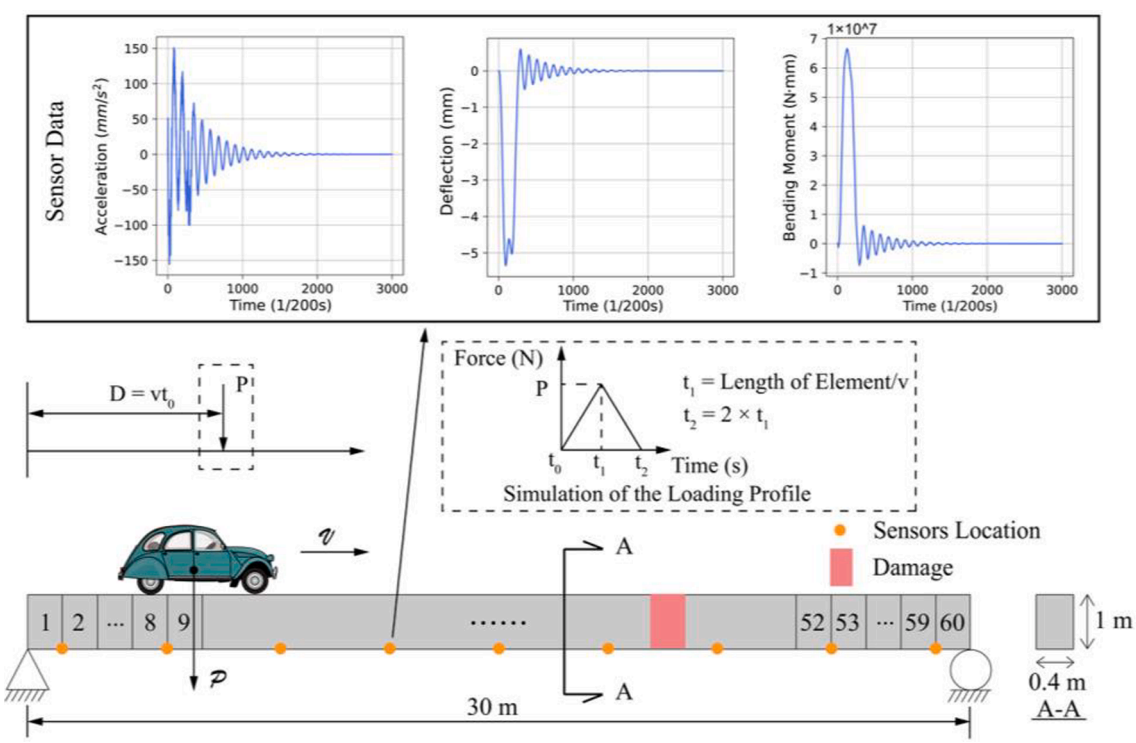

Fig. 1. Surrogate model of the simply supported bridge under a moving vehicle with a constant speed.

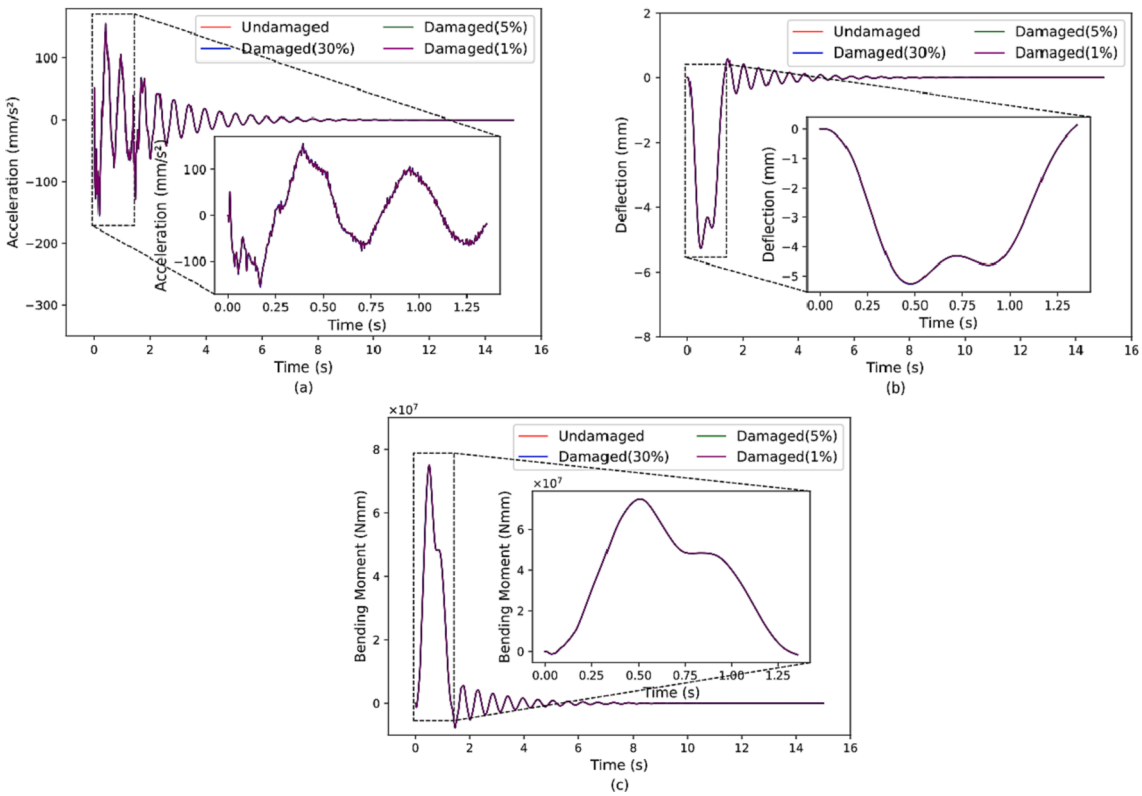

Fig. 2. Example time-series data generated from the sensor at mid-span under a moving vehicle: (a) acceleration; (b) deflection; and (c) bending moment.

vehicle mass, damage locations and damage severities (i.e. reduction in the local second moment of area as a percentage of the original). Fig. 2 shows the samples of the time-series data generated from the sensor at the mid-span. It is notable that even the effects of 30% of damage severity in a single element cannot be identified by naked eye from the raw data.

3. Damage identification & localisation methodology

3.1. Data-Level fusion – PCA

PCA is an unsupervised learning algorithm that projects the data onto a low-dimension subspace retaining the maximum possible variance from the original data, and having as reference orthogonal axes (Principal Components (PCs)) the directions of maximum variance of the original data [40].

Prior to performing the PCA, each type of time-series data (accelerations, deflections and bending moments) acquired from nine different locations was merged into a unique dataset in matrix form, resulting in three matrices (one each for accelerations, deflections and bending moments). Standardisation of the data was not required at this stage because each matrix contained the same type of information, although collected at different positions on the bridge. Each matrix took the form:

$$\mathbf{X}_{m\times n} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

where m is the number of data samples and n is the number of sensors (i.e. each type of data was acquired from 9 sensor locations in this study). Following the formation of these matrices, the first step to perform the PCA was calculating the sample covariance matrix:

$$\mathbf{C} = \frac{1}{m-1}\sum_{i=1}^{m}\left(x_{i\alpha} - \bar{x}_i\right)\left(x_{i\beta} - \bar{x}_i\right)^T \tag{2}$$

where $x_{i\alpha}$ and $x_{i\beta}$ are variables at the α-th and β-th column of i-th row of the matrix $\mathbf{X}_{m\times n}$, and $\bar{\mathbf{x}}$ is the array of the data mean of the matrix $\mathbf{X}_{m\times n}$:

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x_i} \tag{3}$$

As the covariance matrix is symmetric, it satisfies the following condition:

$$\mathbf{CV} = \mathbf{V\Lambda} \tag{4}$$

where $\mathbf{V}$ and $\mathbf{\Lambda}$ denote the eigenvectors and eigenvalues of the covariance matrix $\mathbf{C}$, respectively. The calculation of eigenvectors and eigenvalues was implemented by eigendecomposition. As all elements in Eq. (4) are in the matrix form, the eigendecomposition is expressed as:

$$\mathbf{C} = \mathbf{V\Lambda V}^T \tag{5}$$

Subsequently, the eigenvectors and corresponding eigenvalues were sorted in descending eigenvalue order. A matrix $\mathbf{W}$ is formed by selecting t eigenvectors with the largest eigenvalues, in which t is less than n. The projection of the raw data, $\mathbf{P}$, is achieved by the dot product of the eigenvectors matrix $\mathbf{W}$ and the raw data matrix $\mathbf{X}$, as expressed below:

$$\mathbf{P} = \mathbf{WX} \tag{6}$$

where the matrix $\mathbf{P}$ represents the projection of the raw data in a lower-dimensional space.

3.2. Feature-level fusion – MD

Feature-level fusion is an essential step to provide further refinement of the features established in data-level fusion (e.g. via PCA), to enhance sensitivity to relevant parameters (e.g. structural damage in this study). The feature-level fusion was performed here by calculating the MD using the features obtained through the application of PCA (i.e. projected data

along with PCs). The mathematical expression of MD is defined as:

$$\mathrm{MD} = \sqrt{(\mathbf{y} - \bar{\mathbf{y}})^{\mathrm{T}} \mathbf{C}^{-1} (\mathbf{y} - \bar{\mathbf{y}})} \tag{7}$$

where **y** denotes the row vector of the feature matrix obtained from the data-level fusion, $\bar{y}$ are the mean vectors of the samples (columns of the feature matrix) and $\mathbf{C}^{-1}$ represents the inverse of the covariance matrix of the features. Based on this definition, the MD represents a distance metric measuring the distance among observations in multivariate space.

It is notable that the median values ($\tilde{\mathbf{y}}$) of the projected data were used in place of the mean to provide a robust estimation of the centre of the dataset [41], because the acquired time-series data consist of a large number of zero observations within the fixed time window. The modified expression of MD is:

$$\mathrm{MD} = \sqrt{(\mathbf{y} - \tilde{\mathbf{y}})^{\mathrm{T}} \mathbf{C}^{-1} (\mathbf{y} - \tilde{\mathbf{y}})} \tag{8}$$

3.3. Deep convolutional auto encoder (DCAE)

Ordinarily, AE is an artificial neural network (ANN) that consists of an encoder and a decoder, in which the former attempts to learn a compressed representation of the input data by reducing its dimensionality, while the latter uses the learned knowledge to reconstruct the original input [42]. An AE trained for anomaly detection should accurately reconstruct data representing 'normal' behaviour, but have very poor performance when attempting to reconstruct data if anomalies are involved (i.e. unexpected behaviour or, in this study, 'damage') [43]. The reconstruction error was adopted as the anomaly indicator, that is, samples with reconstruction error exceeding an automatically defined threshold derived during the training process, are classified as anomalies (i.e. damage). A simple AE structure is shown in Fig. 3.

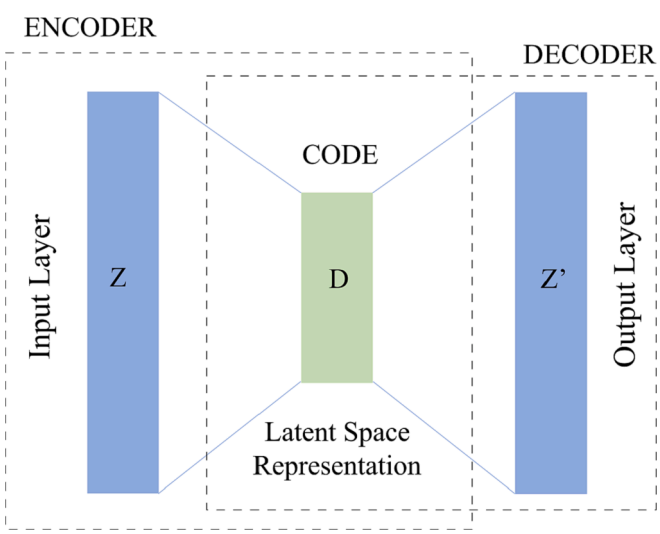

Fig. 3. An illustration of an Auto-Encoder.

The goal during training of the AE is to minimise the reconstruction error so that the predicted data are as close as possible to the original data which include no anomalies. In this study, the reconstruction error was defined using the mean absolute error (MAE) loss function, which is one of the most common metrics used to measure prediction accuracy for continuous samples. The loss function is:

$$\mathrm{MAE} = \frac{1}{\mathrm{w}} \sum_{\mathrm{j}=1}^{\mathrm{w}} \left| \hat{\mathbf{z}}_{\mathrm{j}} - \mathbf{z}_{\mathrm{j}} \right| \tag{9}$$

in which $\hat{\mathbf{z}}_{\mathbf{j}}$ represents the reconstruction of the j-th input and $\mathbf{z}_{\mathbf{j}}$ is the actual value of the j-th input.

In an ANN, an activation function needs to be assigned to each of the neurons to map the inputs of the neuron to its corresponding output, as expressed below:

$$\theta^{(\mathrm{p})} = f\left(\sum_{\mathrm{k}=1}^{1} \mathrm{a}_{\mathrm{k}}^{(\mathrm{p})} \rho_{\mathrm{k}}^{(\mathrm{p})} + \mathrm{b}^{(\mathrm{p})} \right) \tag{10}$$

where p corresponds to the p-th neuron of a neural network layer, k represents the k-th input data, a is the weight parameter, b is a bias, and the $\mathrm{f}(\cdot)$ denotes the activation function. In this work, the rectified linear unit (ReLU) activation function was used for all the neural network layers. The ReLU is expressed as:

$$f(\mathrm{q}) = \begin{cases} 0 & \text{if } \mathrm{q} < 0; \\ q & \text{otherwise.} \end{cases} \tag{11}$$

A common issue with ANN-based algorithms is computational efficiency. Hence, convolution layers were employed in the proposed AE as they can increase the model efficiency by reducing the number of parameters in comparison to conventional fully-connected networks [44]. Convolution is a linear mathematical operation that integrates two sets of information into a new one, which is the integral of the product of these two sets of information after one is reversed and shifted. The convolution layer conducts the convolution operation on the input data in a discrete, rather than continuous way by sliding short arrays (filters) over the input. In other words, the convolution layer performs the element-wise product between each variable of the filter and the input at the location of each input variable and sums them to attain the output

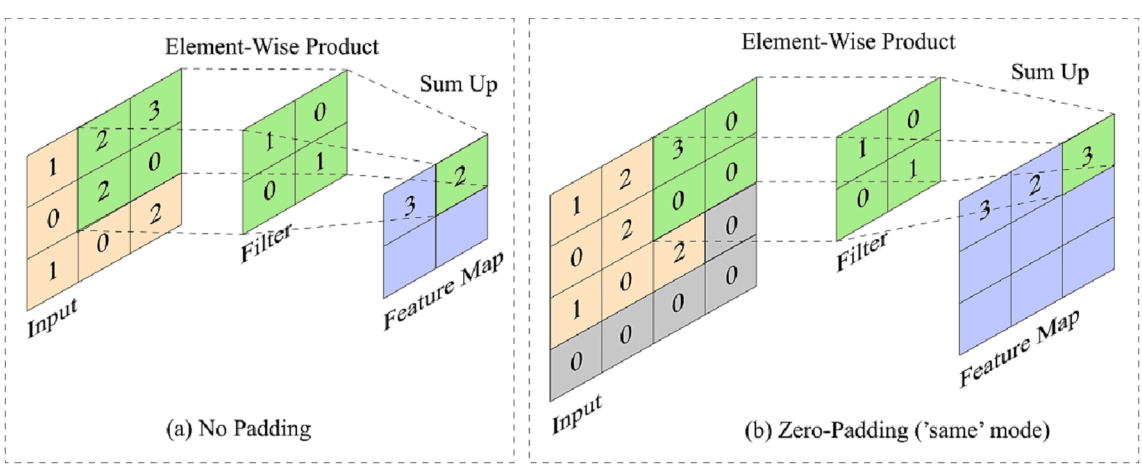

Fig. 4. Convolution layer with (a) no padding; (b) zero-padding.

Table 3
Structure of the proposed DCAE.

Layers	Input Shape	Output Shape	Number of Filters	Filter Size	Activation	Padding
Conv1D_1	1 × 120 × 1	1 × 120 × 512	512	1 × 7	ReLU	Same
Conv1D_2	1 × 120 × 512	1 × 120 × 256	256	1 × 7	ReLU	Same
Conv1D_3	1 × 120 × 256	1 × 120 × 128	128	1 × 7	ReLU	Same
Conv1D_4	1 × 120 × 128	1 × 120 × 64	64	1 × 7	ReLU	Same
Conv1D_5	1 × 120 × 64	1 × 120 × 32	32	1 × 7	ReLU	Same
Conv1D_6	1 × 120 × 32	1 × 120 × 16	16	1 × 7	ReLU	Same
Conv1D_Transpose_1	1 × 120 × 16	1 × 120 × 32	32	1 × 7	ReLU	Same
Conv1D_Transpose_2	1 × 120 × 32	1 × 120 × 64	64	1 × 7	ReLU	Same
Conv1D_Transpose_3	1 × 120 × 64	1 × 120 × 128	128	1 × 7	ReLU	Same
Conv1D_Transpose_4	1 × 120 × 128	1 × 120 × 256	256	1 × 7	ReLU	Same
Conv1D_Transpose_5	1 × 120 × 256	1 × 120 × 512	512	1 × 7	ReLU	Same
Conv1D_Transpose_6	1 × 120 × 512	1 × 120 × 1	1	1 × 7	–	Same

Note: Dropout layers are employed after each convolution layer with a dropout rate of 0.2.

variables in the corresponding positions, namely the feature map [45]. Weights are equally applied across all input data. These shared weights enable the identification and extraction of local feature patterns, since the convolution filters scan the input data across all possible spatial positions.

To implement a convolution layer, several hyperparameters must be defined prior to the training process, such as the size of filters, number of filters, padding and stride. The size and number of filters define the volume that performs convolutions on the input data. The stride is the number of samples by which the filter moves after each convolution operation. As the filter is contained in the input, the size of the feature map is normally smaller than the input (Fig. 4a). Padding, typically zero-padding, is used to address this issue by adding a specific number (zeroes in zero-padding) to the boundaries of the input (Fig. 4b).

Since the output of the feature-level fusion was structured as a single-dimensional array, 1D convolution layers (Conv1D) were employed in the DCAE encoder. While for the decoder, transposed 1D convolution layers (Conv1D_Transpose) were used to perform convolution in the opposite direction. Dropout layers were employed after each convolution layer with a dropout rate of 0.2, that is, 20% of neurons were randomly selected and ignored during the training process. An early-stopping strategy was employed to terminate the training process when performance on a validation dataset degrades a preselected number of consecutive epochs. Both dropout and early stopping avoided overfitting the model. The optimisation was conducted by using the Nadam (Nesterov-accelerated Adaptive Moment Estimation) optimiser which combines the Adam algorithm and Nesterov momentum [46]. The optimisation occurs over a number of epochs, in which the entire dataset is passed forward and backward through the DCAE. An initial learning rate of 0.001 was set for the algorithm, and adaptively reduced between successive epochs. The initial number of epochs was set as 120. The proposed DCAE was implemented with Keras V2.7.0 and Python 3.8.5, and the calculations were performed by GPU computing on a NVIDIA Quadro P1000 graphics card. The details of the DCAE are summarised in Table 3.

3.4. Damage identification approach

The proposed damage identification approach (Fig. 5) involves the following levels:

i) Accelerations, deflections and bending moments are monitored at nine equidistant sensor locations on the bridge to capture the structural responses during the passage of the vehicle, as well as the responses during damped oscillations. The vehicle is assumed to pass over the bridge at a constant speed. A fixed 15-second time window is employed for data acquisition during each vehicle passing event, at the sampling rate of 200 Hz.
ii) A batch of data (i.e. at least 15 events of a vehicle passing over the bridge) is collected initially to build up the baseline DCAE model. The trained DCAE model is then employed to assess the health condition of the bridge using a new batch of data (i.e. at least one event of a vehicle passing over the bridge).
iii) At the data-level fusion stage, each type of data (i.e. accelerations, displacement and bending moment) is fused individually via PCA, extracting three features for each event from the raw data. Notably, 95% of the variance presented in accelerations and bending moments, and 99.9% of variance presented in deflections were maintained in the extracted features. The extracted PC features from each event are then structured into matrix form for use in the subsequent step.
iv) At the feature-level fusion stage, MD is calculated to integrate the PC features into a 1-dimensional feature, in order to improve the sensitivity to damage. MDs are calculated individually for each event based on the distribution of the data from baseline events.
v) Prior to carrying out the training and anomaly detection, several data augmentation processes were conducted, including data alignment, scaling, transformation and moving average. Data alignment aims to align the MDs of different speeds with respect to the position of the moving vehicle on top of the bridge. Min-Max scaling was used to scale the aligned MDs into the range 0 to 1. Subsequently, a data transformation process was used to enlarge the difference between the 'normal' data (i.e. undamaged) and the anomalies (i.e. damaged). This was achieved by detrending the scaled MDs, and then multiplying each scaled MD by the 20th power of its own standard deviation. Furthermore, a moving average of each aligned MD was computed, reducing the size of the array. Finally, all the resulting data were scaled, dividing them by their minimum values and sequentially placed into a single 1D array.
vi) The resulting 1D array of baseline events was used to train the DCAE neural networks. The maximum reconstruction error from the training process is automatically defined as the threshold for anomaly detection. When facing the unseen data (i.e. event(s) to be assessed), data points are classified as anomalies, if the reconstruction error exceeds the threshold, indicating the presence of damage.

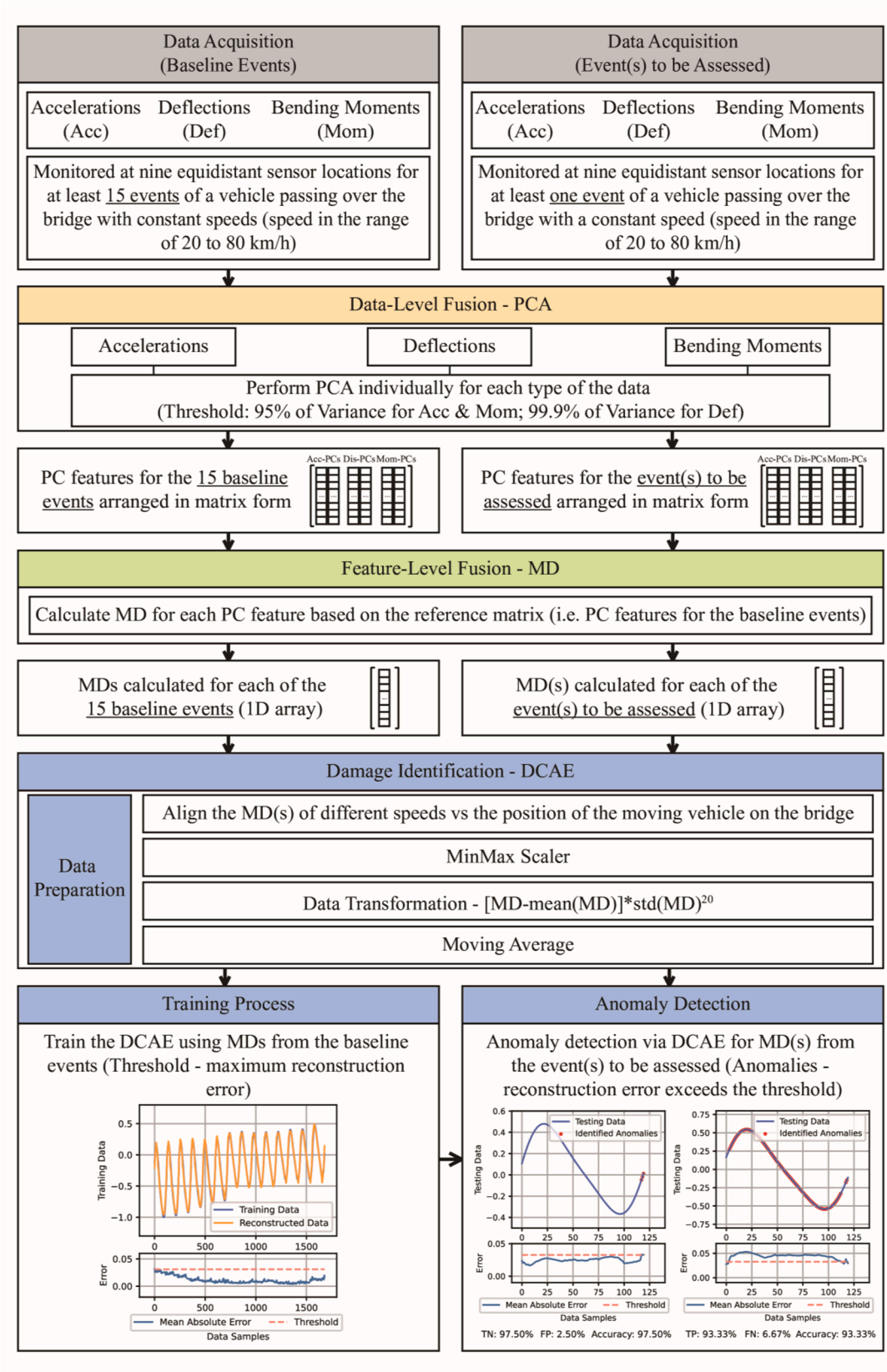

Fig. 5. Flowchart of the proposed multi-level data fusion and anomaly detection approach.

3.5. Damage localisation

In those cases in which the presence of damage is identified by the framework, a second step for the localisation of the damage can be performed. Damage localisation was implemented by following the same first three steps from the damage identification approach, with two additional processes: calculation of the MD ratio and data transformation. The former was implemented by dividing the MD from an undamaged event by the MD from the damaged event with the closest speed of the moving vehicle. The subsequent data transformation employed to transform the MD ratio is shown in Eq. (12):

$$\mathrm{MDR_T} = \left| \frac{\mathrm{MDR_i} - \bar{\mathrm{MDR}}^2}{\max\left|\mathrm{MDR_i} - \bar{\mathrm{MDR}}^2\right|} \right| \tag{12}$$

in which $\mathrm{MDR_i}$ represents the individual value of the MD ratio array; $\bar{\mathrm{MDR}}^2$ represents the square of the average MD ratio.

The position(s) of the peak(s) in MD ratio indicates the damage location(s), if the magnitude of the peak(s) is greater than 0.65. Since the damage is simulated by reducing the second moment of area of a specific finite element(s) (each with a length of 0.5 m), the region formed by the peak(s) (i.e. identified damage location(s)) and 0.25 m either side of the peak(s) is defined as the identified damaged region.

4. Case study

In this section, the performance of the proposed damage identification and localisation approaches are evaluated by using synthetic data generated from the surrogate finite element models, following the flow diagram illustrated in Fig. 5. The following confusion matrix was employed to evaluate the performance of the approach:

- True Positive (TP): actual events are anomalies (i.e. damaged), and predicted as anomalies;
- True Negative (TN): actual events are normal (i.e. undamaged), and predicted as normal;
- False Positive (FP): actual events are normal (i.e. undamaged), but predicted as anomalies (i.e. damaged);
- False Negative (FN): actual events are anomalies (i.e. damaged), but predicted as normal (i.e. undamaged).

In addition, the accuracy of anomaly detection is presented as a percentage of the total number of data points that were correctly identified as damaged or undamaged.

As shown in Fig. 6, the performance of the approach for anomaly detection on two individual vehicle passing events (an undamaged event & an event of Case II with 1% damage severity) was investigated, demonstrating the approach's ability for near-real-time health condition assessment during each traffic loading event on a bridge. In this case, 1800 data points (15 randomly selected events) from the baseline (undamaged) case were used to train the DCAE. The maximum reconstruction error from the training process was automatically defined as the threshold for the subsequent anomaly detection. Damage identification was conducted using data from one unseen event (120 data points). The data points were identified as anomalies if the reconstruction error exceeded the pre-defined threshold, indicating the presence of damage(s). It can be observed that 3 out of 120 data points for the undamaged event (Fig. 6a) were misclassified as false positives, achieving an accuracy of 97.5%. In addition, 8 out of 120 data points for the damaged event (Fig. 6b) were not identified (i.e. false negatives), resulting in the accuracy for the damage identification of 93.33%. All these misclassifications were located at the beginning and/or the end of the event, due to the merging of the data from different vehicle passage events into a 1D array to train the DCAE without any further treatment. As a consequence, irregular patterns (i.e. abrupt jumps or downs in values) appear at those positions, challenging the definition of a 'normal' pattern (i.e. undamaged pattern).

To demonstrate the performance and robustness of the proposed approach, five subsections are discussed in the following: (1) influence of vehicle mass; (2) influence of sensor spatial resolution; (3) influence of measurement noise; (4) damage identification and localisation for a variety of single and multiple damages cases for the simply supported bridge; and (5) application to continuous bridge. However, it is not realistic to present the results for each of the vehicle passage events of all the tested cases, thus, for the subsequent cases, 15 events (simply supported bridge: 1800 data points; continuous bridge: 450 data points for left and right side spans and 900 data points for mid-span) were randomly selected from the baseline (undamaged) case to train the DCAE. The maximum reconstruction error from the training process was automatically defined as the threshold for the subsequent anomaly detection. The framework was tested against 47 different events: the remaining 16 events (simply supported bridge: 1920 data points; continuous bridge: 480 data points for left and right side spans and 960 data points for mid-span) from the baseline case (undamaged) (i.e. testing that the approach correctly identifies undamaged cases for vehicle speeds that are were not encountered during the training

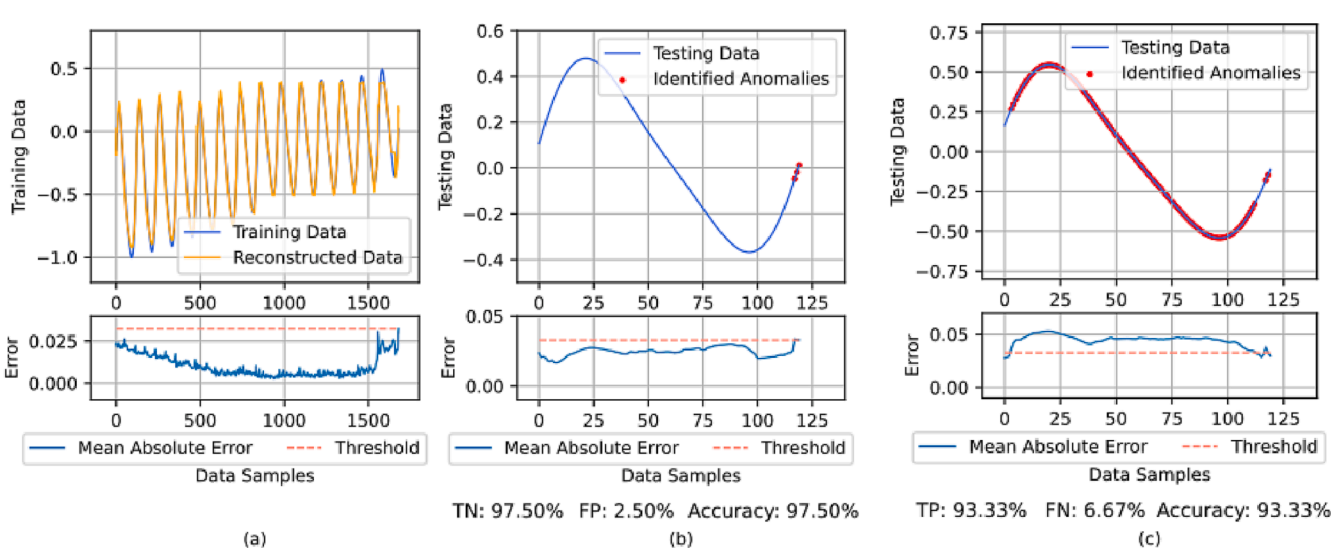

Fig. 6. Near-real-time damage identification for individual vehicle passage events: (a) model training; (b) anomaly detection for an undamaged event; and (c) anomaly detection for a damaged event. The confusion matrix and overall accuracy results are also presented.

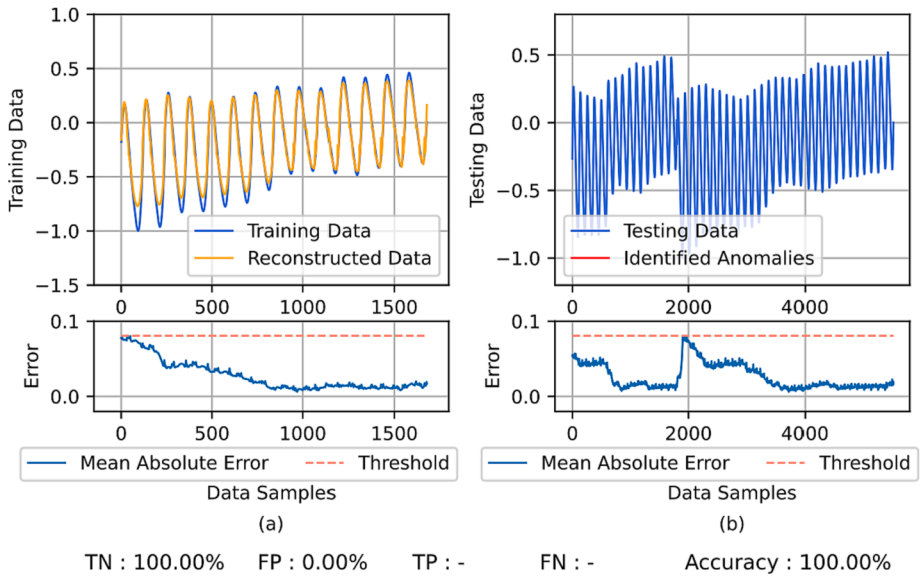

Fig. 7. Anomaly detection results for Case I (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

process), as well as 31 events (simply supported bridge: 3720 data points; continuous bridge: 930 data points for left and right side spans and 1860 data points for mid-span) from an unseen case, all with different vehicle speeds. The average accuracy was investigated among all the vehicle passage events with different moving speeds. Based on the above, the framework was tested for damage identification while also investigating the occurrence of potential false positives (i.e. identified damages for undamaged events). For each case in which damage has been identified, the damage localisation process has subsequently been conducted to determine the position of the damage(s).

4.1. Influence of vehicle mass

The finite element simulation used a linear elastic constitutive model, hence scaling the mass of the moving vehicle magnifies the magnitude of the structural responses by the same factor, and had a significant impact on the magnitude of the extracted MDs. Hence, the anomaly detection must be independent of vehicle mass.

Fig. 7a and 7b demonstrate that the performance of the proposed approach was not influenced by the mass of the moving vehicle: training used data from a 10kN moving mass and testing used data from a 20kN moving mass, which resulted in 100% of the data being correctly identified as 'normal' (undamaged). This is because the approach employed a data scaling process prior to the damage identification process, which eliminated differences in structural response magnitude.

4.2. Influence of sensor spatial resolution

The sensor spatial resolution can significantly influence the performance of the data analysis approach. In this subsection, the influence of sensor coverage on the performance of the approach is investigated by conducting damage identification on Case III (5 % damage severity) with different numbers of sensors installed on the bridge. The inversed Euclidean distance (S) of the damage-sensitive feature (i.e. MDs) was employed as a similarity measure to evaluate the differences between

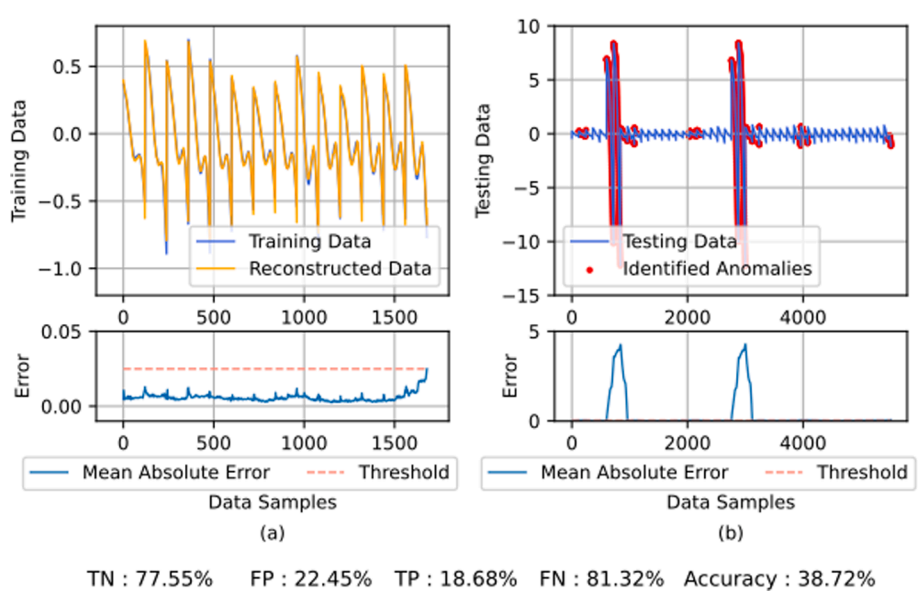

Fig. 8. Anomaly detection results for Case III (5% damage severity) with 5 equidistant sensor locations (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

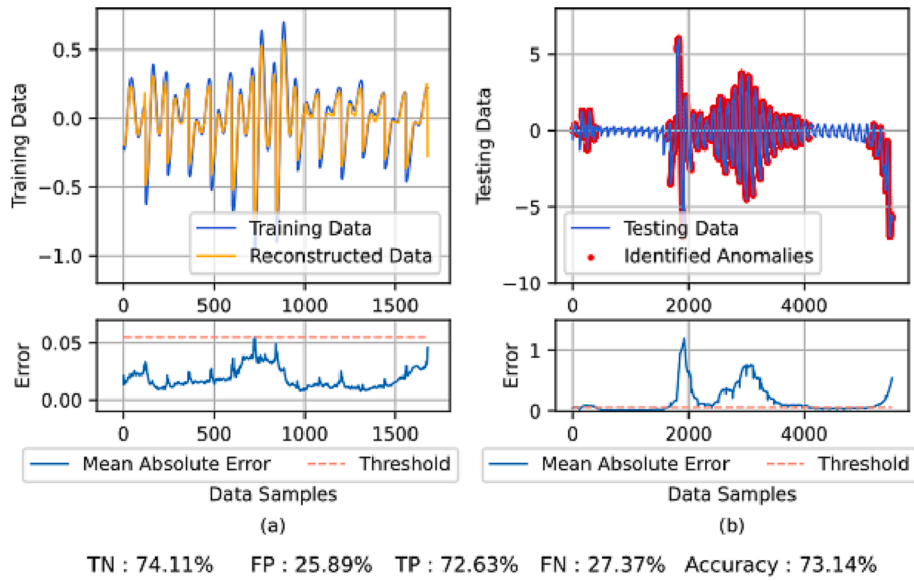

Fig. 9. Anomaly detection results for Case III (5% damage severity) with 7 equidistant sensor locations (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

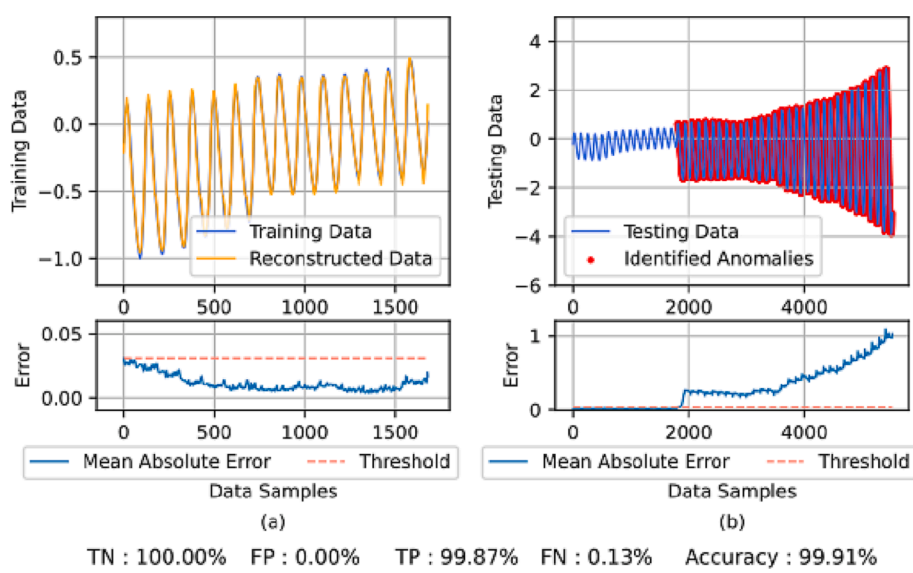

Fig. 10. Anomaly detection results for Case III (5% damage severity) with 9 equidistant sensor locations (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

features from undamaged ($\mathbf{u}_i$) and damaged ($\mathbf{v}_i$) cases with different amounts of sensors on the bridge. The similarity measure is expressed as:

$$S = \frac{1}{1 + \sqrt{\sum_{i=1}^{n} (\mathbf{u}_i - \mathbf{v}_i)^2}} \tag{13}$$

Figs. 8 and 9 show that the anomaly detection was less accurate for cases of 5 and 7 equidistant sensors, which resulted in accuracies of 38.72% and 73.14%, respectively. False alarms (i.e. false positives) among the tested events appeared and a large number of data points from the damaged case were not identified. Figs. 10 to 12 show that all cases with 9, 11 and 13 equidistant sensor locations consistently achieved over 99.8% accuracy.

Fig. 13 shows that the MD similarities between the damaged and undamaged cases decreased with increasing sensor coverage, which demonstrates that the extracted features were more sensitive to structural damage when a larger number of sensors (9 or more) were installed. This highlights that the anomaly detection (damage identification) algorithm performs well with 9 or more sensors for the cases studied.

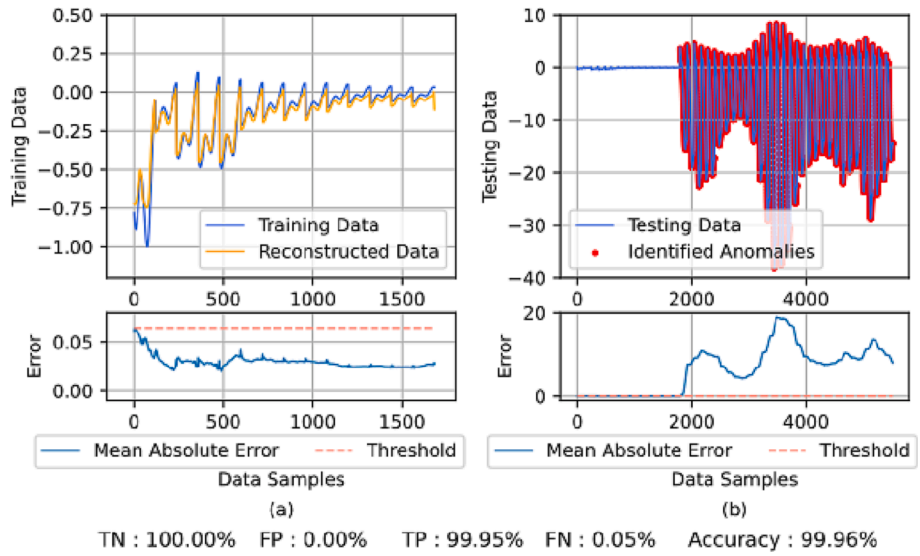

Fig. 11. Anomaly detection results for Case III (5% damage severity) with 11 equidistant sensor locations (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

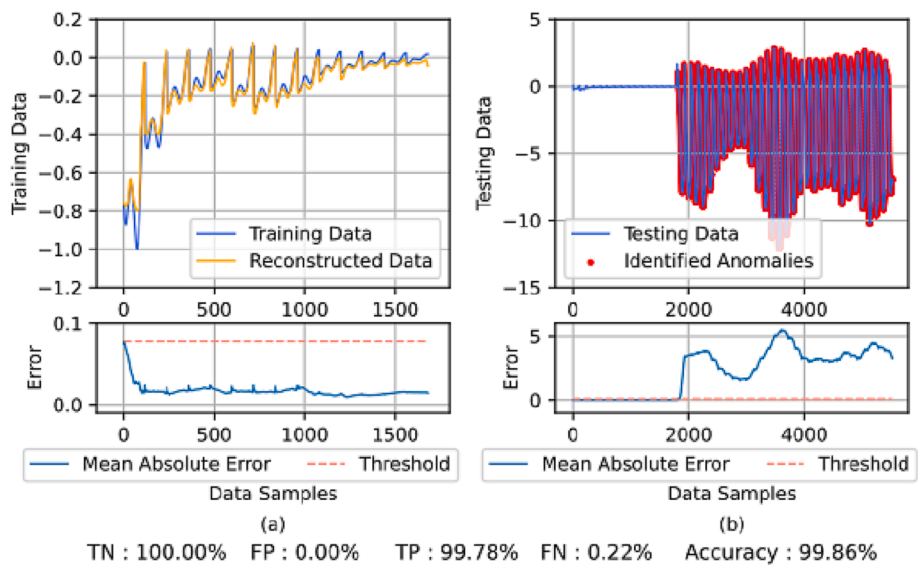

Fig. 12. Anomaly detection results for Case III (5% damage severity) with 13 equidistant sensor locations (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

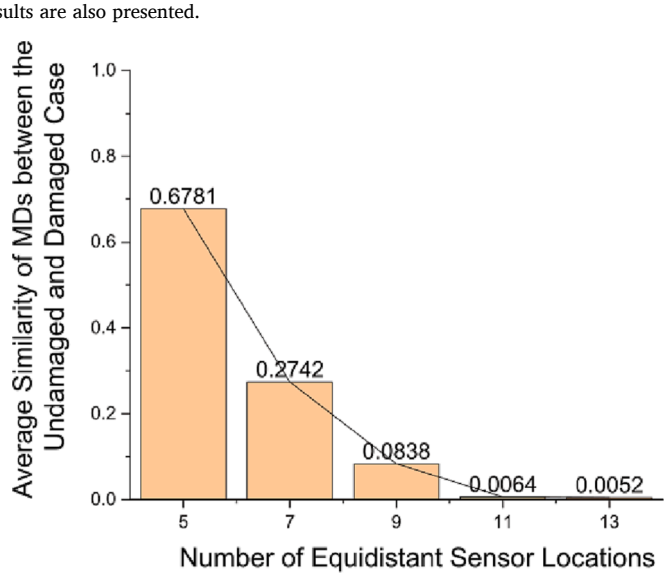

Fig. 13. Average similarity of MDs between the baseline case and Case III (5% damage severity) with different sensor coverage.

4.3. Influence of measurement noise

In practice, measurement noise is unavoidable in the field and can significantly contaminate the data, leading to discrepancies between measurements and the real structural behaviour. It is critical to mitigate the impact of noise to enable accurate damage identification. In this study, noisy data were simulated by adding a white Gaussian noise (a normal distribution with zero means) to the clean data, resulting in a signal–noise ratio (SNR) of 60 dB.

The initial data acquisition sampling rate was set to 20 kHz and white Gaussian noise was added to achieve an SNR of 60 dB. The first denoising process downsampled the noisy data to 50 Hz. This was implemented by averaging every 400 data points among the data array. A low-pass filter was then employed with a cut-off frequency of 15 Hz for accelerations and 10 Hz for deflections and bending moments, to improve the noise reduction effect. The performance of the proposed denoising approach was evaluated using data from Case III with a damage severity of 1%.

Fig. 14 shows that the denoising method can significantly reduce the influence of noise on the extracted damage-sensitive features. As shown in Fig. 15, following denoising, the accuracy of damage identification was over 98% (only 88 out of 5640 data points were incorrectly classified), even with just a 1% of local reduction in the second moment of area at the specific location on the bridge.

4.4. Damage identification & localisation

4.4.1. Damage identification

Figs. 16 to 18 show the damage identification results for Case II, and all the damage identification results are summarised in Tables 4 and 5. Table 4 shows that the approach can accurately identify the health condition of the bridge when a single location was damaged. The accuracy of anomaly detection was consistently greater than 93% even for very small damage severity (1% reduction in the second moment of area at a specific location) on the bridge. Accuracies over 98.5% were achieved when the damage severity was 5%, and accuracies over 99.5% when the damage severity was 30%. Table 5 shows the approach enabled accurate damage identification on multi-damage scenarios,

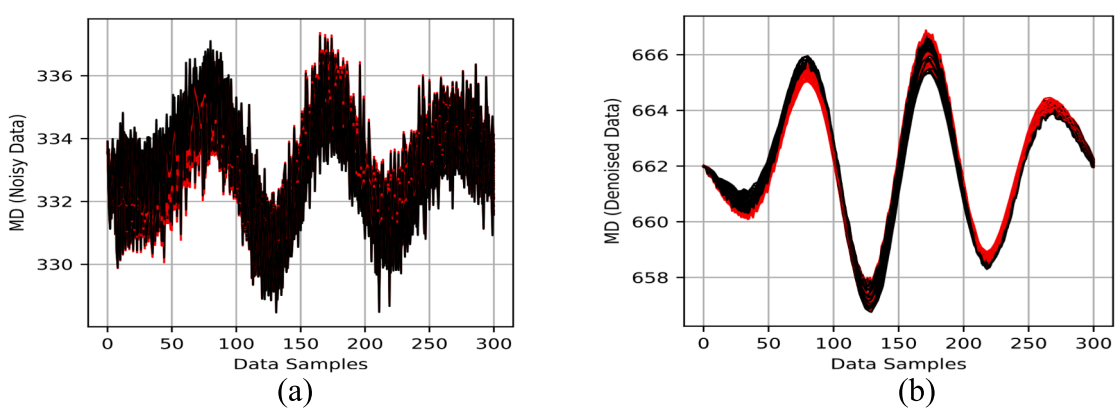

Fig. 14. Extracted MDs after multi-level data fusion (a) noisy data; (b) denoised data (undamaged case: in red colour; damaged case: in black colour). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

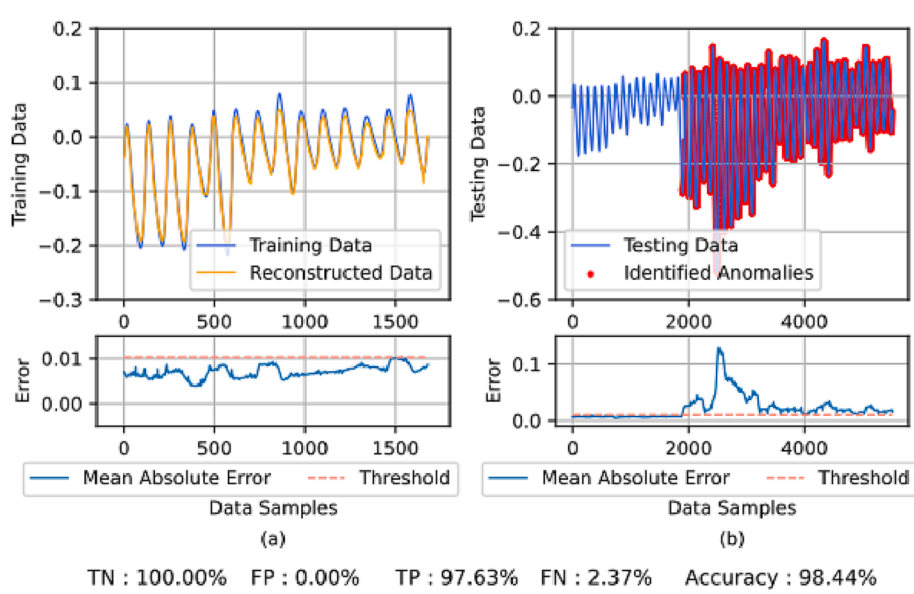

Fig. 15. Damage identification after denoising the data (Case III with 1% damage severity) (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

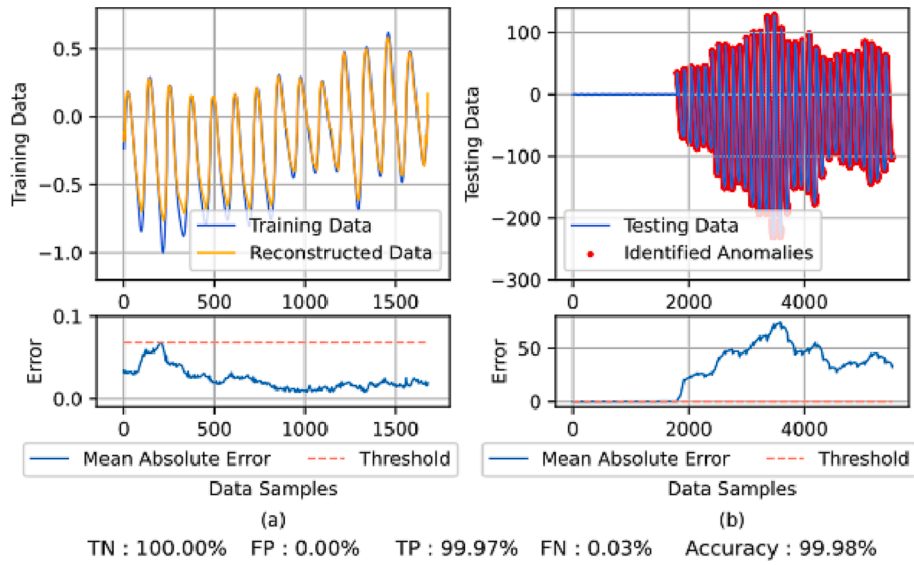

Fig. 16. Damage identification for Case II with 30% damage severity (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

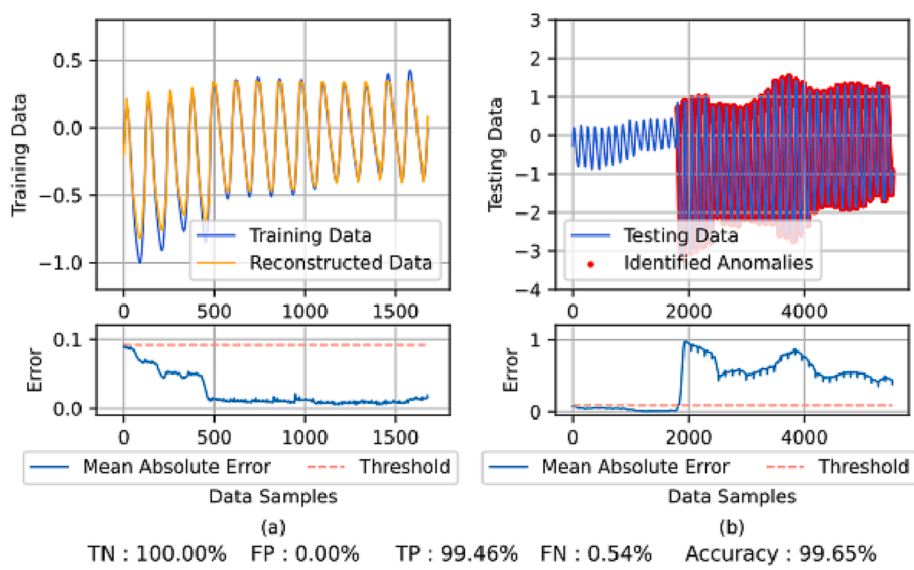

Fig. 17. Damage identification for Case II with 5% damage severity (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

resulting in accuracies greater than 97.5%. The anomaly detection accuracy for all tested cases decreased as the damage severity decreased. This is expected because smaller damage severity induces only minor changes to the data pattern, which increases the challenge of damage identification. By comparing Tables 4 and 5, it can be observed that the accuracy of multi-damage scenarios was higher than that of each single damage scenario. This is because, with additional damage on the bridge, the damage-induced data pattern was more significant (i.e. less similarity to the undamaged case), which improved the performance of damage identification.

It is notable that no false alarms (i.e. false positives) occurred among the tested cases, except for Case II with a damage severity of 1%, which also demonstrated that the proposed approach was not influenced by different velocities of the vehicle. False negatives (i.e. undetected damage) were less than 2% for most of the tested cases and this was generally due to the specific data arrangement prior to the damage identification process, that is, the data were merged in a 1D array. A few data points at the beginning and/or the end of some events were incorrectly identified as false negatives. However, these points could be filtered out from the reconstructed 1D array to achieve higher accuracy.

As shown in Fig. 18 (i.e. Case II with 1% damage severity), 15 out of 1920 data points were classified as false positives, and 361 out of 3720 data points were classified as false negatives. This indicated that the DCAE did not correctly define the boundary for the 'normal' (undamaged) region during the training process. This is because a moving average process was employed prior to damage identification, in which the MD was smoothed. Since the case had only a 1% reduction of the second moment of area at the specific location, the smoothing process reduced the influence of the damage on the data pattern. In addition, the DCAE incorporated an early-stopping strategy that caused the training of the DCAE to be insufficient for this case. This led to 6.67% of the data being misclassified. To overcome this issue, retraining of the DCAE could be conducted to improve its performance, as the retraining uses the existing model parameters as the starting point instead of random initialisation of the parameters which avoids a biased network.

The proposed damage identification approach provided a reliable

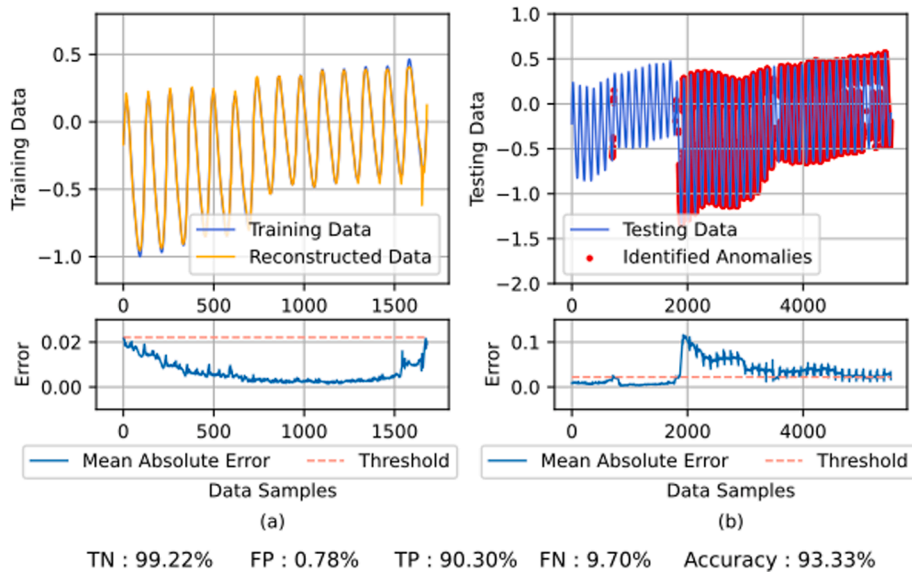

Fig. 18. Damage identification for Case II with 1% damage severity (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

Table 4
Damage identification results for single-damage scenarios.

Case	Damage Severity	True Negative	False Positive	True Positive	False Negative	Overall Accuracy
II	30%	100.00%	0.00%	99.97%	0.03%	99.98%
	5%	100.00%	0.00%	99.46%	0.54%	99.65%
	1%	99.22%	0.78%	90.30%	9.70%	93.33%
III	30%	100.00%	0.00%	99.97%	0.03%	99.98%
	5%	100.00%	0.00%	99.92%	0.08%	99.95%
	1%	100.00%	0.00%	98.06%	1.94%	98.72%
IV	30%	100.00%	0.00%	99.97%	0.03%	99.98%
	5%	100.00%	0.00%	99.95%	0.05%	99.96%
	1%	100.00%	0.00%	98.23%	1.77%	98.83%
V	30%	100.00%	0.00%	99.41%	0.59%	99.61%
	5%	100.00%	0.00%	97.93%	2.07%	98.63%
	1%	100.00%	0.00%	92.50%	7.50%	95.05%

Table 5
Damage identification results for multiple-damage scenarios.

Case	Damage Severity	True Negative	False Positive	True Positive	False Negative	Overall Accuracy
VI	30%	100.00%	0.00%	99.97%	0.03%	99.98%
	5%	100.00%	0.00%	99.95%	0.05%	99.96%
	1%	100.00%	0.00%	98.28%	1.72%	98.87%
VII	30% & 1%	100.00%	0.00%	99.97%	0.03%	99.98%
	1% & 5%	100.00%	0.00%	96.69%	3.31%	97.82%

framework to assess the health condition of a simply supported bridge. It is notable that the approach assumes no damage is present on the bridge at the moment of training the DCAE, so that the acquired data are solely from the undamaged situation. In other words, the suggested method diagnoses the extent of relative damage developed on the bridge after the training process has been completed. In practice, a structural inspection of the bridge is a necessary first step before the monitoring system is configured, which will enable the initial bridge health condition to be established.

4.4.2. *Damage localisation*

Figs. 19 to 23 show that the proposed damage localisation approach can accurately locate the damage position(s) on the bridge. For single-damage cases (Figs. 19 to 21), the predicted damage region overlapped with the actual damage region for all the tested cases (i.e. varying damage locations and severities). In the cases with two damages, it can be observed that the approach successfully localised both damage locations if these damages had equal severity, as shown in Fig. 22. However, performance was reduced for multiple damages with different

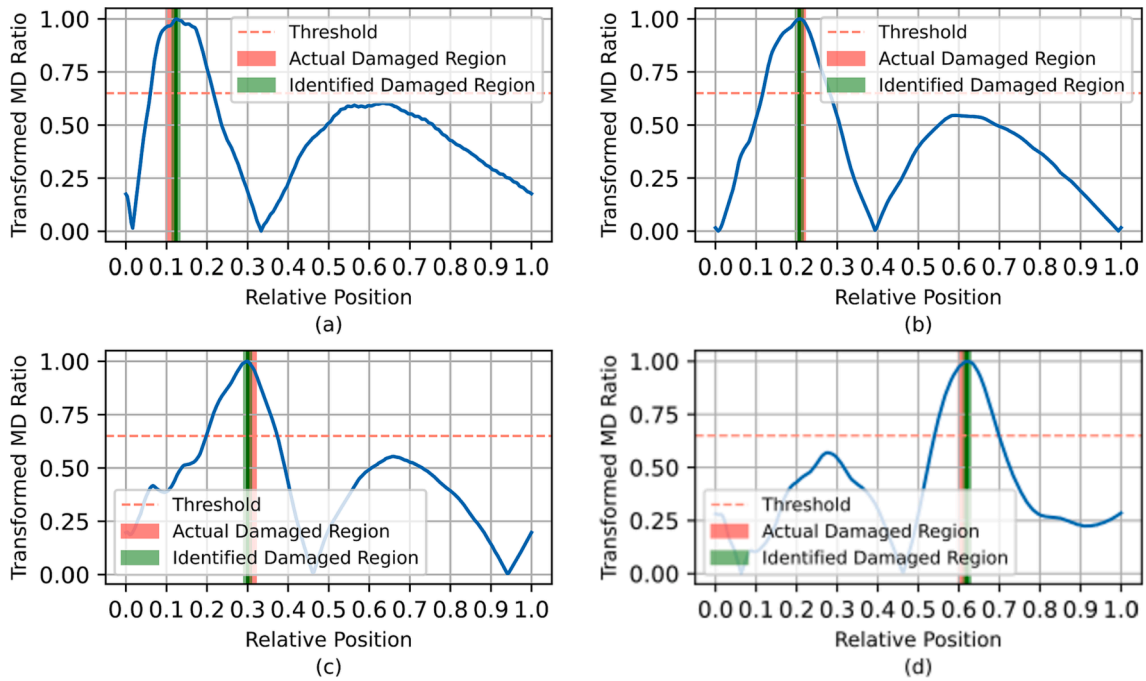

Fig. 19. Damage localisation for single-damage scenarios with 30% damage severity: (a) Case II; (b) Case III; (c) Case IV; and (d) Case V.

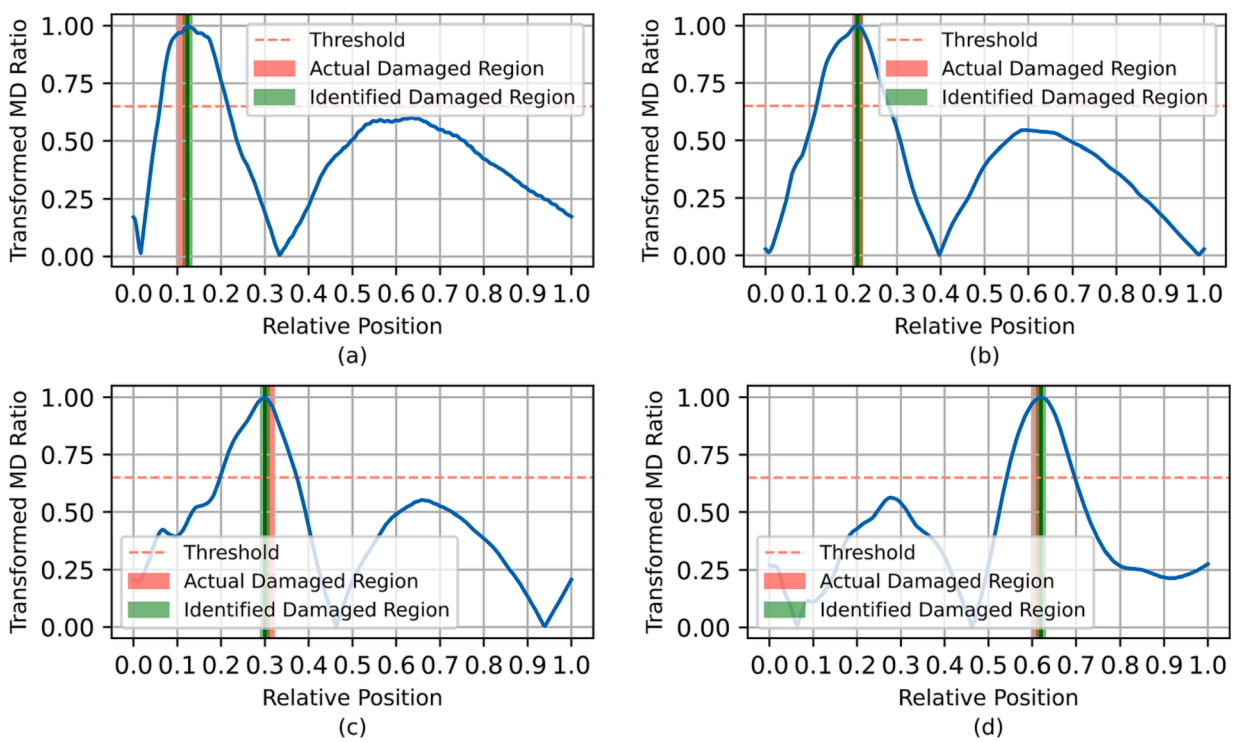

Fig. 20. Damage localisation for single-damage scenarios with 5% damage severity: (a) Case II; (b) Case III; (c) Case IV; and (d) Case V.

damage severities. If the difference in damage severity between the multiple damages was small (e.g. 1% and 5%, as shown in Fig. 23b), both damage locations could be identified. However, if the difference in damage severity was large (e.g. 30% and 5%, as shown in Fig. 23a), only the largest damage severity could be located. In practice, once the largest damage is repaired, the small damage can be detected by conducting damage identification using the previously trained DCAE, and the damage location can be provided subsequently.

The proposed damage localisation method requires the data from both undamaged and damaged cases to be from events where the moving vehicle was travelling at the same or similar speed, in order to provide accurate localisation to damage(s). In a real-world situation, once the damage identification process raises an alarm to the control system, a future passing event with a vehicle travelling within the required speed range could be used for damage localisation, or a vehicle could be employed to pass over the bridge at a designated speed, so that the system could provide a reliable decision on the damage location.

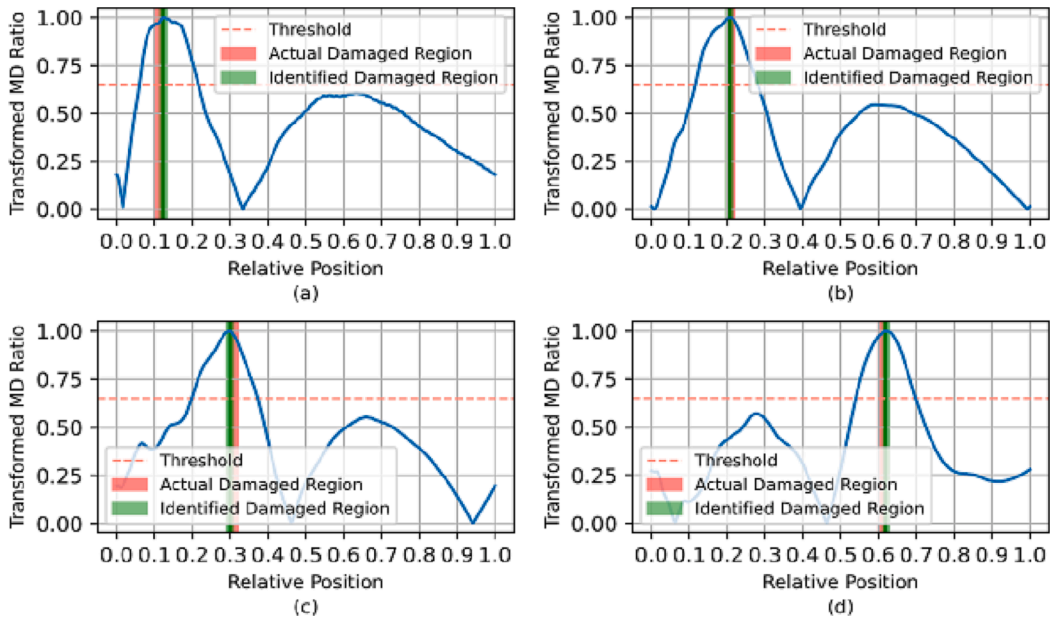

Fig. 21. Damage localisation for single-damage scenarios with 1% damage severity: (a) Case II; (b) Case III; (c) Case IV; and (d) Case V.

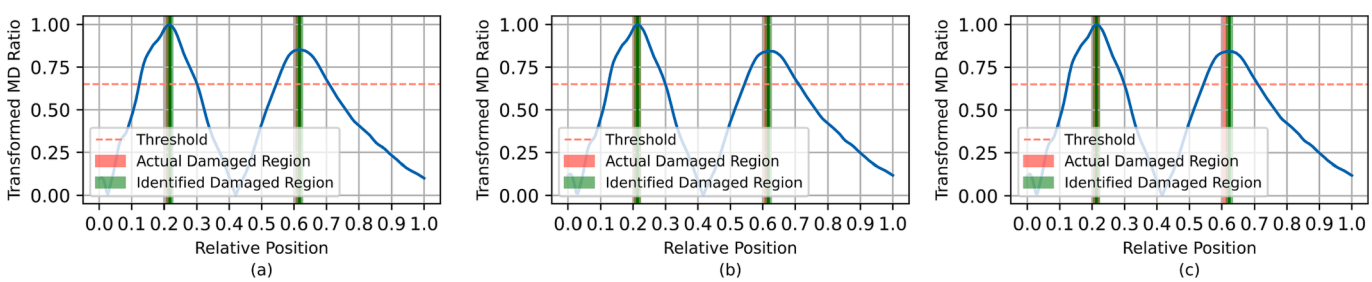

Fig. 22. Damage localisation for multiple-damage scenarios with equalled damage severity: (a) 30% damage severity; (b) 5% damage severity; and (c) 1% damage severity.

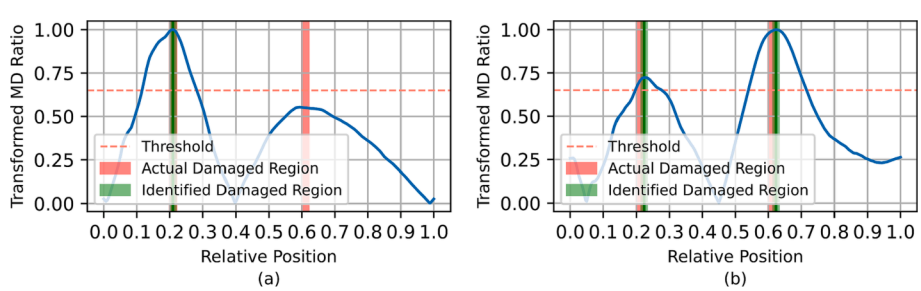

Fig. 23. Damage localisation for multiple-damage scenarios with unequalled damage severity: (a) 30% & 1% damage severity; and (b) 1% & 5% damage severity.

4.5. Application to continuous bridge

Fig. 24 shows the numerical models of a continuous bridge with a total length of 30 m and with a moving vehicle, simulated using SAP2000. It was divided into 180 equal-sized finite elements. Accelerations, deflections and bending moments at the nodes corresponding to nine equidistant sensor locations per span (Fig. 24) were measured at a sampling rate of 200 Hz.

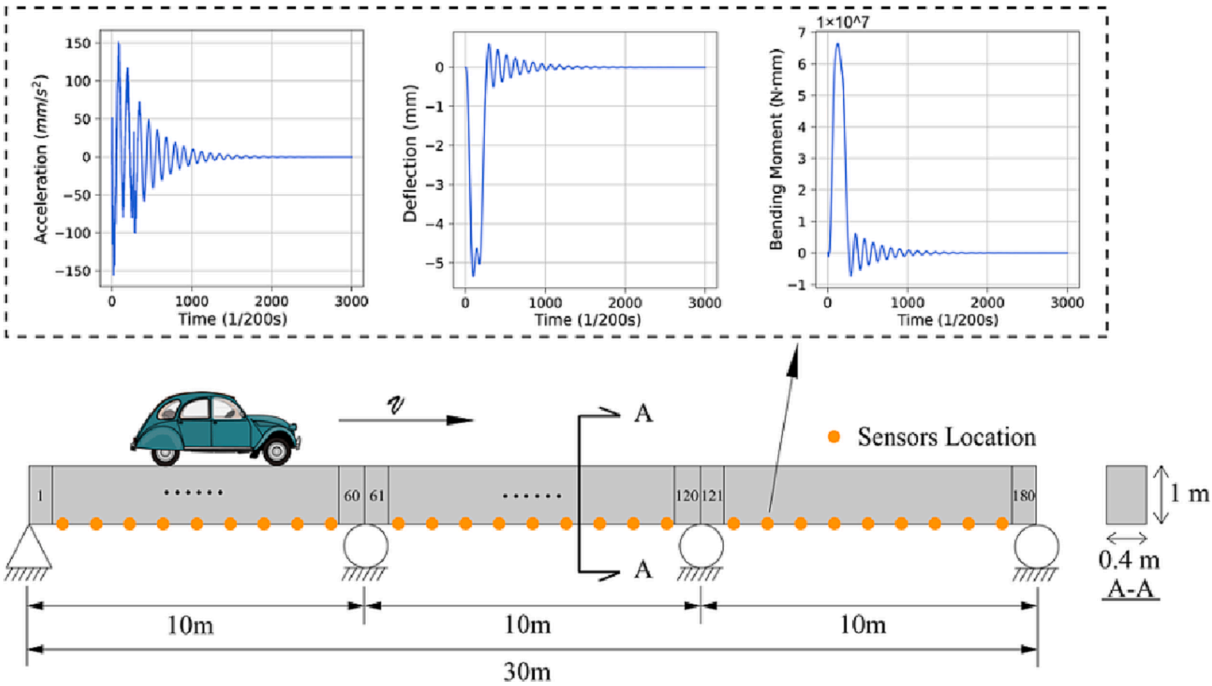

Fig. 24. Surrogate model of the continuous bridge under a moving vehicle with a constant speed.

4.5.1. Modification to the framework

Changes to boundary conditions alter the structural response, hence minor algorithm modifications (data augmentation steps) were required for damage identification and localisation of the continuous bridge. Firstly, the damage identification and localisation approach for the continuous bridge works on a span by span basis. This is because although damage in one span could affect the response in all spans, only minor influence occurred compared to that in the damaged span. Conducting damage identification and localisation span by span directly neglects information from the other spans, which improved the accuracy of damage identification.

One change was made in estimating the number of principal components for PCA, 98% of the total variance in the original data should be preserved when analysing data from the mid-span, while 99.99% of the total variance was required to analyse data from the side spans. The main modification to the framework was in the data augmentation/preparation step prior to the training process of the DCAE, in which the aligned MDs with respect to the examined span are used. Double moving averages were employed before the data transformation process, and the data transformation used to enlarge the difference between the 'normal' data (i.e. undamaged) and anomalies (i.e. damaged) was achieved by detrending the scaled MDs, and then multiplying each scaled MD by its own standard deviation and squared mean. For a long series of data, double moving averages eliminate the short-run fluctuations in the data [47]. Fig. 25 shows the modification of the damage identification framework for the continuous bridge compared to the framework for the simply supported bridge.

The damage localisation approach is conducted after the presence of damage(s) is identified in the specific span. Data from the damaged span identified is used to localise the damage(s), following the first three steps from the damage identification approach. It is notable that the preserved percentage of the total variance in PCA is modified (i.e. 90% of the total variance in the original data is required). The calculation of the MD ratio is implemented by dividing the MD from an undamaged event by the MD from the damaged event with the same or similar speed of the moving vehicle. The lowest valley indicates the damage location. The region formed by the valley(s) (i.e. identified damage location(s)) and 0.084 m on either side of the valley(s) is defined as the identified damage region, as the damage(s) is simulated by reducing the second moment of area of a specific finite element(s) (each with a length of 0.167 m). It is notable that the damage localisation approach has minor differences from that used in the simply supported bridge, in which the latter employed a data transformation process. This is because the valley in MD ratio did not consistently indicate the damage location in the simply supported bridge, while the data transformation process ensured accurate damage localisation in an automatic framework.

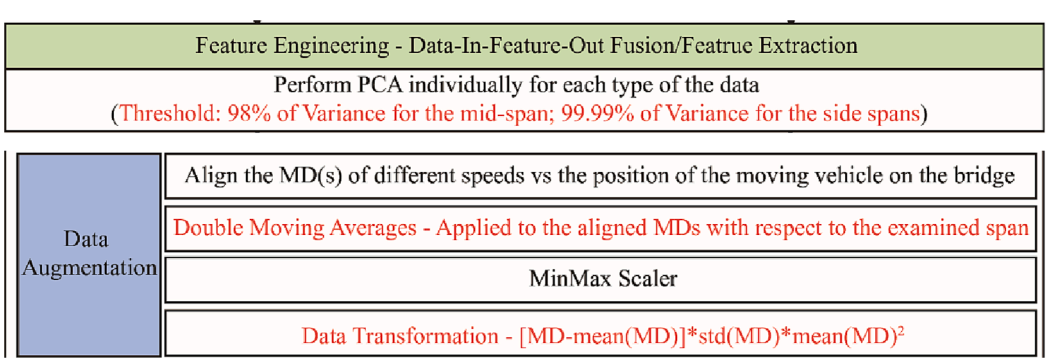

Fig. 25. Modification of the damage identification framework for the continuous bridge.

4.5.2. Damage identification and localisation

Figs. 26 to 28 show examples of the damage identification results of the continuous bridge (Case I – Damage Location: 0.033~0.039) with more results summarised in Table 6. Figs. 29 to 30 show examples of the damage localisation results. Refer to the thesis[48] for full details. As shown in Table 6, the approach accurately identified the health condition of the continuous bridge for single and multiple damaged scenarios, as the accuracy of anomaly detection was greater than 94.9% for the tested scenarios, only except the case when evaluating the health condition of the right side span, though the damage was present at the mid-span (i.e. Case II – Damage Location: 0.566~0.572), where an accuracy of 82.62% was achieved. The accuracy of damage identification can be improved if using data obtained from the vehicle passage events from the opposite direction. In this case, the right span would become the first span to conduct damage identification, and an accuracy of greater than 95.9% could be achieved.

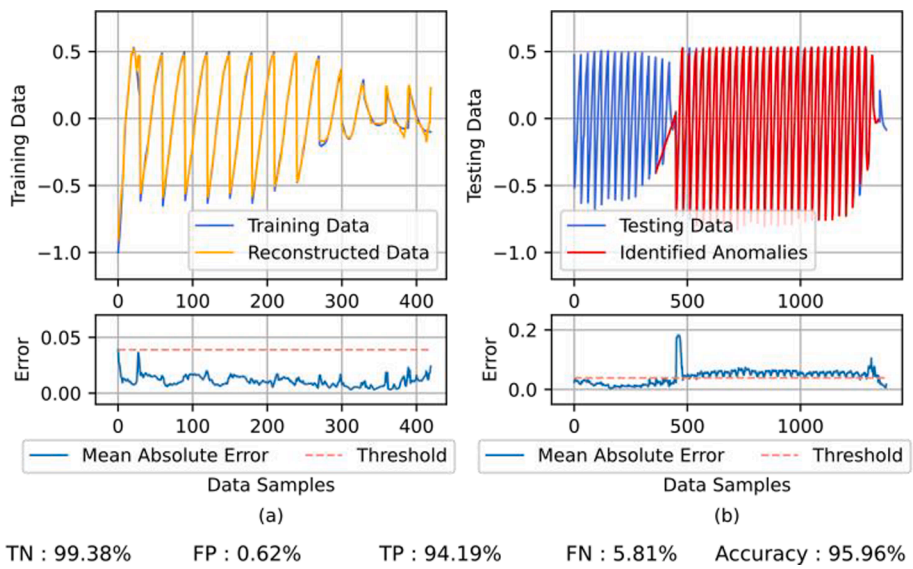

Fig. 26. Damage identification for Case I (damaged span: left side span, damage location: 1~1.17 m from the left-hand side support) with 15% damage severity using data obtained from the left side span: (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

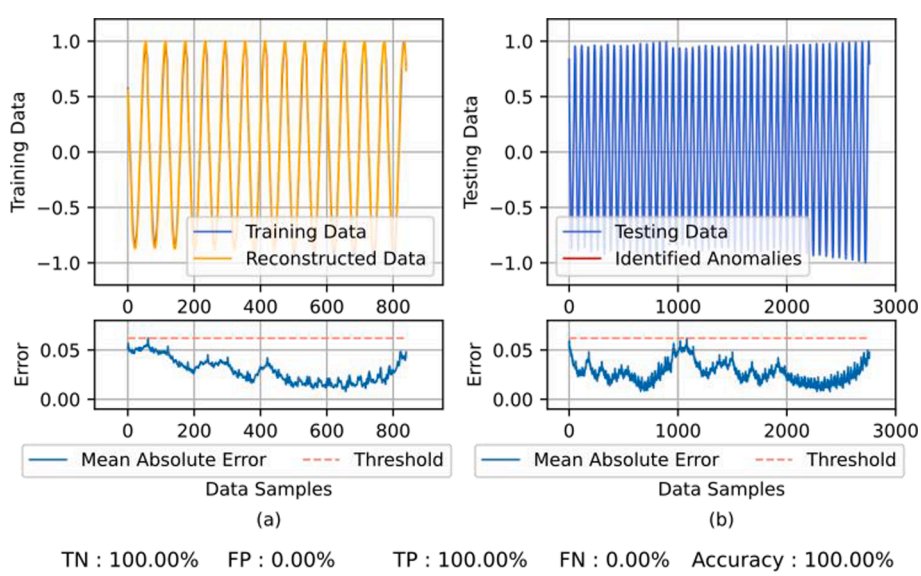

Fig. 27. Damage identification for Case I (damaged span: left side span, damage location: 1~1.17 m from the left-hand side support) with 15% damage severity using data obtained from the mid-span: (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

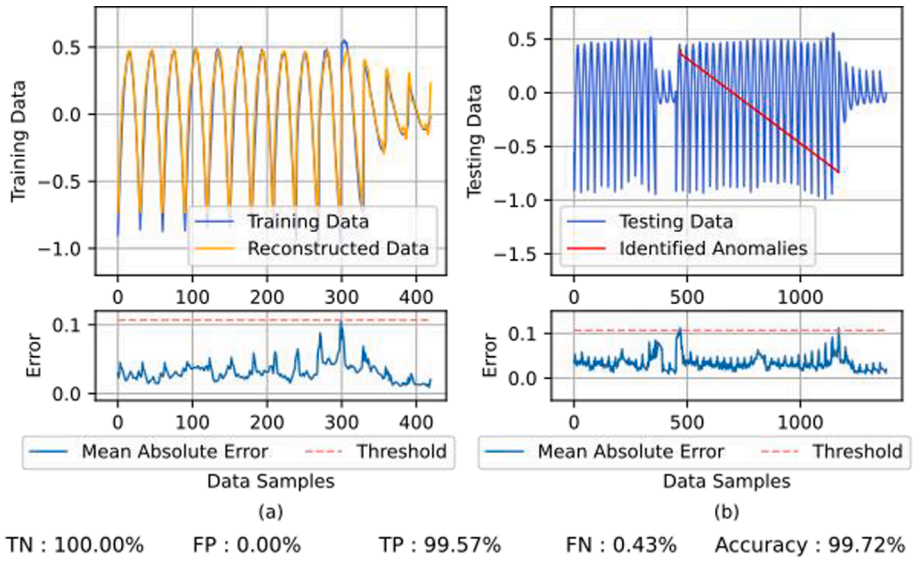

Fig. 28. Damage identification for Case I (damaged span: left side span, damage location: 1~1.17 m from the left-hand side support) with 15% damage severity using data obtained from the right side span: (a) model training; and (b) anomaly detection. The confusion matrix and overall accuracy results are also presented.

Table 6
Damage identification results for the continuous bridge.

Case	Damage Severity	Span	True Negative	False Positive	True Positive	False Negative	Overall Accuracy
I	15%	Left	99.38%	0.62%	94.19%	5.81%	95.96%
(Damage Location: 0.033 ~ 0.039)		Mid	100%	0%	100%	0%	100%
		Right	100%	0%	99.57%	0.43%	99.72%
II	15%	Left	100%	0%	99.89%	0.11%	99.93%
(Damage Location: 0.566 ~ 0.572)		Mid	99.90%	0.10%	93.98%	6.02%	95.99%
		Right	98.33%	1.67%	74.52%	25.48%	82.62%
III	15%	Left	100%	0%	97.42%	2.58%	98.30%
(Damage Location: 0.787~0.793)		Mid	100%	0%	100%	0%	100%
		Right	90.00%	10.00%	99.89%	0.11%	96.52%
IV	15% & 15%	Left	100%	0%	97.96%	2.04%	98.65%
(Damage Location: 0.1~0.106 & 0.233~0.239)		Mid	100%	0%	100%	0%	100%
		Right	96.25%	3.75%	95.91%	4.09%	96.03%
	25% & 15%	Left	100%	0%	99.89%	0.11%	99.93%
		Mid	100%	0%	99.89%	0.11%	99.93%
		Right	92.29%	7.71%	96.34%	3.66%	94.96%
V	15% & 15%	Left	95.62%	4.38%	98.71%	1.29%	97.66%
(Damage Location: 0.1~0.106 & 0.787~0.793)		Mid	100%	0%	100%	0%	100%
		Right	100%	0%	99.25%	0.75%	99.50%
	20% & 25%	Left	100%	0%	99.03%	0.97%	99.36%
		Mid	100%	0%	98.44%	1.56%	98.97%
		Right	96.67%	3.33%	99.68%	0.32%	98.65%

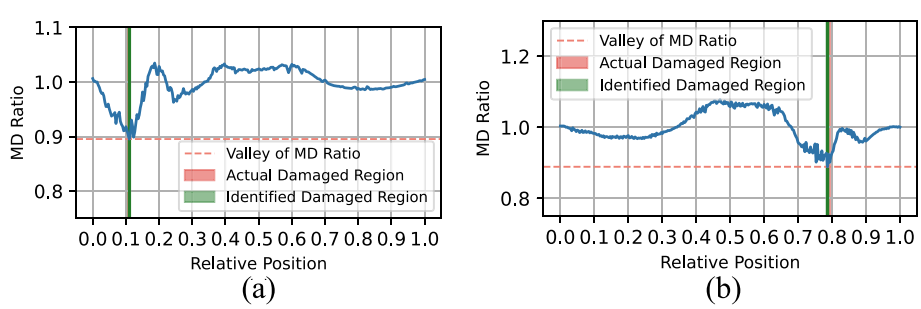

Fig. 29. Damage localisation results for Case V with equalled 15% damage severity (a) using data obtained from the left side span; and (b) using data obtained from the right side span.

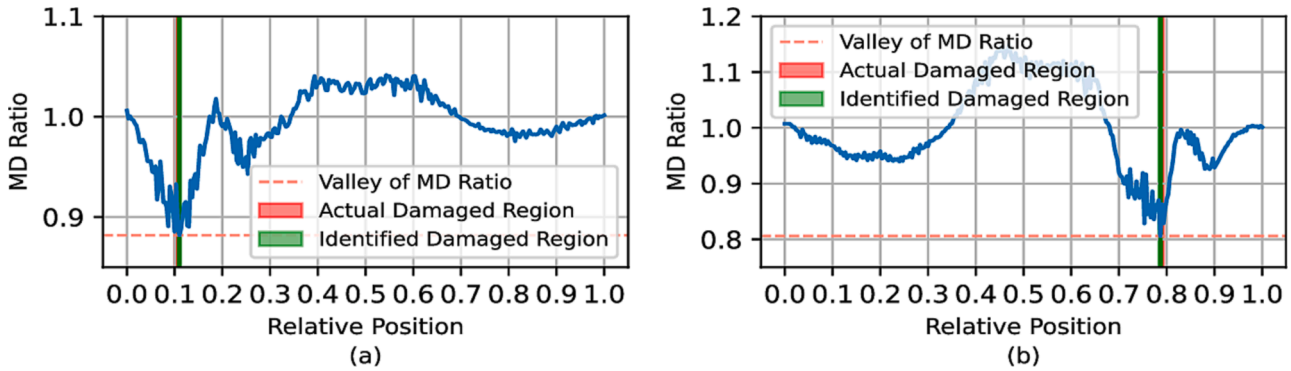

Fig. 30. Damage localisation results for Case V with unequalled damage severity (20% & 25% damage severity respectively): (a) using data obtained from the left side span; (b) using data obtained from the right side span.

5. Conclusions

In this paper, a new approach that integrates multi-level data fusion and machine learning techniques has been developed for damage identification and localisation. Performance of the approach has been demonstrated for a simply supported bridge and a continuous bridge. Accelerations, deflections and bending moments at multiple sensor positions were obtained from surrogate models of the bridges subjected to a single moving vehicle. The extracted damage-sensitive features, which coupled PCA and Mahalanobis distance, were significantly modified by the presence of damage(s) on the bridge. The proposed deep convolutional autoencoder was able to define a representative normal (i.e. undamaged) region and its boundary, which enabled accurate anomaly (damage) identification.

The results described in this paper show that the proposed approach is highly successful in detecting and localising damage(s). For the simply supported bridge, the damage identification accuracies were consistently greater than 93%, even for extremely small damage severity (e.g. 1% reduction in the second moment of area at a specific location), and multiple damages. The method of damage localisation was able to determine the location(s) of single and multiple damages on the bridge, however, if the multiple damages had significantly different severities, only the more sever damage could be located. For the continuous bridge, the accuracy of damage identficcation was greater than 94.9% for the majority of the tested cases.

The approach's ability for near-real-time health condition assessment during each traffic loading event on the bridge was demonstrated using single, independent vehicle passing events. In addition, the developed approach has the ability to handle variations in traffic loading, e.g. mass and speed of vehicles. For real-world applications, the approach should be combined with the recommended sensor coverage (i.e. a minimum of 9 sensor positions per span) and the proposed denoising method, in order to provide accurate and reliable damage identification and localisation. Future work will investigate the performance of the approach for damage identification and localisation on other types of structures using surrogate models, and monitoring data obtained from real structures. In addition, an investigation will be carried out to reduce the sensor coverage while maintaining the damage identification and localisation performance of the approach.

Funding information:

The research presented in this article was conducted with the support of a PhD studentship from Loughborough University, UK. Alister Smith acknowledges the support of a Philip Leverhulme Prize (PLP-2019-017).

CRediT authorship contribution statement

Hao Wang: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft. **Giorgio Barone:** Conceptualization, Methodology, Project administration, Resources, Supervision, Writing – review & editing. **Alister Smith:** Conceptualization, Methodology, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hao Wang reports financial support was provided by Loughborough University. Alister Smith reports financial support was provided by The Leverhulme Trust.

Data availability

Data will be made available on request.

References

[1] Zhang Z, Sun C. A numerical study on multi-site damage identification: a data-driven method via constrained independent component analysis. Struct Control Heal Monit 2020;27(10):1–22. https://doi.org/10.1002/stc.2583.
[2] Chen Z, Zhou X, Wang X, Dong L, Qian Y. Deployment of a smart structural health monitoring system for long-span arch bridges: a review and a case study. Sensors (Switzerland) 2017;17(9). https://doi.org/10.3390/s17092151.
[3] Song G, Wang C, Wang B. Structural health monitoring (SHM) of civil structures. Appl Sci 2017;7(8):3–5. https://doi.org/10.3390/app7080789.
[4] Sony S, Laventure S, Sadhu A. A literature review of next-generation smart sensing technology in structural health monitoring. Struct Control Heal Monit 2019;26(3). https://doi.org/10.1002/stc.2321.
[5] Altunışık AC, Okur FY, Kahya V. Modal parameter identification and vibration based damage detection of a multiple cracked cantilever beam. Eng Fail Anal December 2016;2017(79):154–70. https://doi.org/10.1016/j.engfailanal.2017.04.026.
[6] Pirboudaghi S, Tarinejad R, Alami MT. Damage detection based on system identification of concrete dams using an extended finite element–wavelet transform coupled procedure. JVC/J Vib Control 2018;24(18):4226–46. https://doi.org/10.1177/1077546317722428.
[7] Giagopoulos D, Arailopoulos A, Dertimanis V, Papadimitriou C, Chatzi E, Grompanopoulos K. Structural health monitoring and fatigue damage estimation using vibration measurements and finite element model updating. Struct Heal Monit 2019;18(4):1189–206. https://doi.org/10.1177/1475921718790188.
[8] Kurian B, Liyanapathirana R. Machine learning techniques for structural health monitoring. In: Proceedings of the 13th International Conference on Damage Assessment of Structures; 2019. p. 3–24.
[9] Gulgec NS, Shahidi SG, Matarazzo TJ, Pakzad SN. Current challenges with BIGDATA analytics in structural health monitoring. In: Structural Health Monitoring & Damage Detection. Springer US, vol. 7; 2017. p. 79–84. 10.1007/978-3-319-54109-9.
[10] Cremona C, Santos J. Structural health monitoring as a big-data problem. Struct Eng Int 2018;28(3):243–54. https://doi.org/10.1080/10168664.2018.1461536.
[11] Sarmadi H, Entezami A, Saeedi Razavi B, Yuen KV. Ensemble learning-based structural health monitoring by Mahalanobis distance metrics. Struct Control Heal Monit 2021;28(2):1–24. https://doi.org/10.1002/stc.2663.
[12] Zhou YL, Maia NMM, Abdel WM. Damage detection using transmissibility compressed by principal component analysis enhanced with distance measure. JVC/J Vib Control 2018;24(10):2001–19. https://doi.org/10.1177/1077546316674544.

[13] Shang Z, Sun L, Xia Y, Zhang W. Vibration-based damage detection for bridges by deep convolutional denoising autoencoder. Struct Heal Monit Published online 2020. https://doi.org/10.1177/1475921720942836.

[14] Sousa Tomé E, Pimentel M, Figueiras J. Online early damage detection and localisation using multivariate data analysis: application to a cable-stayed bridge. Struct Control Heal Monit 2019;26(11):1–20. https://doi.org/10.1002/stc.2434.

[15] Lee K, Jeong S, Sim SH, Shin DH. A novelty detection approach for tendons of prestressed concrete bridges based on a convolutional autoencoder and acceleration data. Sensors (Switzerland) 2019;19(7). https://doi.org/10.3390/s19071633.

[16] Wang Z, Cha YJ. Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage. Struct Heal Monit 2021; 20(1):406–25. https://doi.org/10.1177/1475921720934051.

[17] Ozdagli AI, Koutsoukos X. Machine learning based novelty detection using modal analysis. Comput Civ Infrastruct Eng 2019;34(12):1119–40. https://doi.org/10.1111/mice.12511.

[18] Avci O, Abdeljaber O, Kiranyaz S, Hussein M, Gabbouj M, Inman DJ. A review of vibration-based damage detection in civil structures: from traditional methods to Machine Learning and Deep Learning applications. Mech Syst Signal Process 2021; 147:107077. https://doi.org/10.1016/j.ymssp.2020.107077.

[19] Lee K, Jeong S, Sim SH, Shin DH. Field experiment on a PSC-I bridge for convolutional autoencoder-based damage detection. Struct Heal Monit Published online 2020. https://doi.org/10.1177/1475921720926267.

[20] Eleftheroglou N, Zarouchas D, Loutas T, Alderliesten R, Benedictus R. Structural health monitoring data fusion for in-situ life prognosis of composite structures. Reliab Eng Syst Saf June 2017;2018(178):40–54. https://doi.org/10.1016/j.ress.2018.04.031.

[21] Ding Y, Yao X, Wang S, Zhao X. Structural damage assessment using improved Dempster-Shafer data fusion algorithm. Earthq Eng Eng Vib 2019;18(2):395–408. https://doi.org/10.1007/s11803-019-0511-z.

[22] Silva MF, Santos A, Santos R, Figueiredo E, Costa JCWA. Damage-sensitive feature extraction with stacked autoencoders for unsupervised damage detection. Struct Control Heal Monit 2021;28(5):1–19. https://doi.org/10.1002/stc.2714.

[23] Havaran A, Mahmoudi M, Ebrahimpour R. Extraction of the structural mode shapes utilizing image processing method and data fusion. Mech Syst Signal Process 2021; 151:107380. https://doi.org/10.1016/j.ymssp.2020.107380.

[24] Guo T, Xu Z. Data fusion of multi-scale representations for structural damage detection. Mech Syst Signal Process 2018;98:1020–33. https://doi.org/10.1016/j.ymssp.2017.05.045.

[25] Aria A, Lopez Droguett E, Azarm S, Modarres M. Estimating damage size and remaining useful life in degraded structures using deep learning-based multi-source data fusion. Struct Heal Monit 2020;19(5):1542–59. https://doi.org/10.1177/1475921719890616.

[26] Mariniello G, Pastore T, Menna C, Festa P, Asprone D. Structural damage detection and localization using decision tree ensemble and vibration data. Comput Civ Infrastruct Eng. Published online 2020: 1–21. 10.1111/mice.12633.

[27] Esfandiari A, Nabiyan MS, Rofooei FR. Structural damage detection using principal component analysis of frequency response function data. Struct Control Heal Monit 2020;27(7):1–21. https://doi.org/10.1002/stc.2550.

[28] Guo T, Wu L, Wang C, Xu Z. Damage detection in a novel deep-learning framework: a robust method for feature extraction. Struct Heal Monit 2020;19(2):424–42. https://doi.org/10.1177/1475921719846051.

[29] Solimine J, Niezrecki C, Inalpolat M. An experimental investigation into passive acoustic damage detection for structural health monitoring of wind turbine blades. Struct Heal Monit 2020;19(6):1711–25. https://doi.org/10.1177/1475921719895588.

[30] Azim MR, Gül M. Data-driven damage identification technique for steel truss railroad bridges utilizing principal component analysis of strain response. Struct Infrastruct Eng 2020:1–17. https://doi.org/10.1080/15732479.2020.1785512.

[31] Ma Z, Yun CB, Wan HP, Shen Y, Yu F, Luo Y. Probabilistic principal component analysis-based anomaly detection for structures with missing data. Struct Control Heal Monit December 2020;2021:1–25. https://doi.org/10.1002/stc.2698.

[32] Sen D, Erazo K, Zhang W, Nagarajaiah S, Sun L. On the effectiveness of principal component analysis for decoupling structural damage and environmental effects in bridge structures. J Sound Vib 2019;457:280–98. https://doi.org/10.1016/j.jsv.2019.06.003.

[33] Kumar K, Biswas PK, Dhang N. Time series-based SHM using PCA with application to ASCE benchmark structure. J Civ Struct Heal Monit 2020;10(5):899–911. https://doi.org/10.1007/s13349-020-00423-2.

[34] Singh SK, Sikdar S, Malinowski PH. An optimized data fusion strategy for structural damage assessment using electromechanical impedance. Smart Mater Struct 2021; 30(3). https://doi.org/10.1088/1361-665X/abdc07.

[35] Li W, Huang Y. A combined method of cross-correlation and PCA-based outlier algorithm for detecting structural damages on a jacket oil platform under random wave excitations. Appl Ocean Res 2020;102(July):102301. https://doi.org/10.1016/j.apor.2020.102301.

[36] Meixedo A, Santos J, Ribeiro D, Calçada R, Todd M. Damage detection in railway bridges using traffic-induced dynamic responses. Eng Struct. 2021;238(February). 10.1016/j.engstruct.2021.112189.

[37] Casas JR, Moughty JJ. Bridge damage detection based on vibration data: Past and new developments. Front Built Environ 2017;3(February):1–12. https://doi.org/10.3389/fbuil.2017.00004.

[38] Sampaio RPC, Maia NMM, Almeida RAB, Urgueira APV. A simple damage detection indicator using operational deflection shapes. Mech Syst Signal Process 2016;72–73:629–41. https://doi.org/10.1016/j.ymssp.2015.10.023.

[39] Yunkai Z, Jianjun G, Guohua L, Yuchen F. Damage Identification of Simply-Supported Beam Bridge Based on Deflection Influence Line. *IOP Conf Ser Earth*. Environ Sci 2021;643(1). https://doi.org/10.1088/1755-1315/643/1/012049.

[40] Jolliffe IT, Cadima J, Cadima J. Principal component analysis: a review and recent developments Subject Areas. PhilTransRSocA 2016;374(20150202):1–16.

[41] Thennadil SN, Dewar M, Herdsman C, Nordon A, Becker E. Automated weighted outlier detection technique for multivariate data. Control Eng Pract September 2017;2018(70):40–9. https://doi.org/10.1016/j.conengprac.2017.09.018.

[42] Pathirage CSN, Li J, Li L, Hao H, Liu W, Ni P. Structural damage identification based on autoencoder neural networks and deep learning. Eng Struct 2018;172 (April):13–28. https://doi.org/10.1016/j.engstruct.2018.05.109.

[43] Chow JK, Su Z, Wu J, Tan PS, Mao X, Wang YH. Anomaly detection of defects on concrete structures with the convolutional autoencoder. Adv Eng Informatics. 2020;45(February). 10.1016/j.aei.2020.101105.

[44] Cha YJ, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Comput Civ Infrastruct Eng 2017;32(5): 361–78. https://doi.org/10.1111/mice.12263.

[45] Li S, Zhao X, Zhou G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. Comput Civ Infrastruct Eng 2019;34 (7):616–34. https://doi.org/10.1111/mice.12433.

[46] Dozat T. Incorporating Nesterov momentum into Adam. ICLR Work 2016;1: 2013–6.

[47] Yaffee RA, McGee M. An introduction to time series analysis and forecasting: with applications of SAS®and SPSS®. Elsevier Science; 2000. https://books.google.co.uk/books?id=LSojZBiBZBgC.

[48] Wang H. Automated identification of infrastructure damage using real-time monitoring and machine learning. Published online 2022.