Archives of Civil and Mechanical Engineering (2024) 24:190
https://doi.org/10.1007/s43452-024-01001-9

ORIGINAL ARTICLE

Numerical benchmark for road bridge damage detection from passing vehicles responses applied to four data-driven methods

Daniel Cantero[1] · Zohaib Sarwar[1] · Abdollah Malekjafarian[2] · Robert Corbally[2] · Mehrisadat Makki Alamdari[3] · Prasad Cheema[3] · Jatin Aggarwal[4] · Hae Young Noh[4] · Jingxiao Liu[4]

Received: 14 December 2023 / Revised: 28 May 2024 / Accepted: 19 June 2024

Abstract

Drive-by bridge monitoring utilizes measured responses from passing vehicles to perform damage detection of bridge, a methodology challenged by multiple factors and operational conditions. Recently, data-driven methods have been used to improve the accuracy of drive-by monitoring. This thriving research field requires (but lacks) publicly available datasets to improve and validate its monitoring and damage detection capabilities. To foster data-driven drive-by bridge damage assessment methods, this document presents an openly available dataset consisting of numerically simulated vehicle responses crossing a range of bridge spans with various damage conditions. The dataset includes results for different monitoring scenarios, road profile conditions, vehicle models, vehicle mechanical properties and speeds. The intention is to provide a useful resource to the research community that serves as a reference set of results for testing and benchmarking new developments in the field. In addition, four recently published data-driven drive-by methods have been tested using the same dataset.

Keywords Indirect monitoring · Damage detection · Vehicle–bridge interaction · Drive-by inspection

1 Introduction

Worldwide, road infrastructure owners are responsible for the safety of an ever growing and aging stock of bridges. Even though these elements are critical for the economy, the responsible authorities lack sufficient monetary resources to perform continuous monitoring and detailed assessment of all bridges in a road network. The reality is that, at present, it is unfeasible to have and operate comprehensive monitoring systems in a timely and economical manner [1]. The direct Structural Health Monitoring (SHM) is thus generally reserved for long-span bridges.

At the same time, the developments in connected vehicles and on-board sensor systems are opening new possibilities.

✉ Daniel Cantero
daniel.cantero@ntnu.no

Zohaib Sarwar
muhammad.z.sarwar@ntnu.no

Abdollah Malekjafarian
abdollah.malekjafarian@ucd.ie

Robert Corbally
robert.corbally@ucd.ie

Mehrisadat Makki Alamdari
m.makkialamdari@unsw.edu.au

Prasad Cheema
p.cheema@unsw.edu.au

Jatin Aggarwal
jatin08@stanford.edu

Hae Young Noh
noh@stanford.edu

Jingxiao Liu
liujx@stanford.edu

[1] Department of Structural Engineering, Norwegian University of Science and Technology NTNU, Trondheim, Norway

[2] Structural Dynamics and Assessment Laboratory, School of Civil Engineering, University College Dublin, Dublin, Ireland

[3] Centre for Infrastructure Engineering and Safety, School of Civil and Environmental Engineering, University of New South Wales (UNSW), Sydney, Australia

[4] Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305, USA

Published online: 01 July 2024

It is foreseen, that within the next 2 decades, infrastructure operators will avail of large amounts of information obtained directly from travelling vehicles. It has been recognized that this data has the potential to improve the capability of infrastructure operators to perform structural safety assessments [2].

In fact, a new bridge SHM paradigm has emerged over the past 2 decades that utilizes measured responses from passing vehicles, rather than the direct registered behaviour of the bridge. This indirect monitoring of the infrastructure has been termed drive-by, vehicle scanning, mobile sensing, or vehicle assisted. It constitutes a thriving research field, reflected by the number of state-of-the-art reviews published only during the last 5 years, under the terms indirect bridge health monitoring [3], mobile sensing [4], vehicle scanning [5], vehicle-based [6], vehicle-assisted [7] and moving test vehicle [8]. This promising idea enables regular bridge monitoring to be performed either using single specialized vehicles, fleets of similar vehicles or via crowdsourcing from as many available vehicle passages. Despite the undeniable advantages that this indirect monitoring strategy might offer, these methods still face multiple challenges. The research shows that the accuracy of the methods to extract bridge properties and detect eventual damage is affected by road roughness levels, traffic properties (vehicle speed, additional traffic) and environmental conditions (ambient temperature oscillations).

A strategy to overcome the technical challenges in drive-by methods is to rely on data-driven approaches to process the signals, extract useful features and detect and quantify possible structural damages. The operational effects can potentially be assimilated by processing large numbers of vehicle responses. With data-driven approaches, the influence of vehicle speed, mechanical properties and other factors can be compensated and used to trace accurately the state of a target bridge [9].

In general, the use of data-driven methods has been identified by several road authorities as a promising way forward to assess the health and condition of highway infrastructure (including bridges) [10] and predict the remaining useful life of assets [11]. Regarding the drive-by approach, the aggregate use of multiple vehicle passages, or crowdsensing, has been mentioned as a potential substitute [12] or as a complement [13] to traditional monitoring methods. However, the progress in this field is limited by the lack of openly available datasets from real-life full-scale tests [1], needed to improve and validate the concept of drive-by bridge monitoring and damage detection.

In an effort to foster data-driven drive-by bridge damage assessment methods, this document presents an openly available dataset. The dataset consists of numerically simulated vehicle responses crossing a range of bridge spans with various damage conditions. In addition, the dataset includes results for different road profile conditions, vehicle models, vehicle mechanical properties and speeds. The intention is to provide a useful resource to the research community that serves as a reference set of results for testing and benchmarking new developments in the field. This document gives in Sect. 2 a detailed description of the dataset, together with links to the hosting repository for public download. Furthermore, four recently published data-driven drive-by methods have been applied to the mentioned dataset. The results for each of these methods are presented in separate sections. In particular, Sect. 3 performs damage detection through Artificial Neural Network (ANN) trained to predict the contact point response [14]. In Sect. 4, a deep autoencoder model (DAE) is trained for bridge damage-sensitive features, which is then used to detect bridge damage from shifts in the distribution of reconstruction error [15]. Section 5 successfully applies to the dataset a nonlinear dimensionality reduction technique based on Uniform Manifold Approximation and Projection (UMAP) together with a non-parametric clustering technique using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [16]. Section 6 explores the use of Hierarchical Multi-task Unsupervised Domain (HierMUD) [17], an adaptation framework capable of transferring damage diagnosis from one bridge to another without requiring any new labels.

Therefore, the new contribution of this work is the presentation of a publicly available dataset for benchmarking future research developments in the field of drive-by bridge monitoring. The dataset includes over half a million numerically simulated bridge and vehicle responses for a wide range of structural configurations and conditions, loaded by a variety of vehicle models with properties and speeds representative of normal traffic under operational conditions. In addition, four different existing data-driven methodologies have been applied to the same dataset. The results from these distinct methods serve as a starting point for benchmarking improvements in drive-by technology.

2 Dataset

This section describes in detail the dataset NuBe-DBBM (Numerical Benchmark for Drive-By Bridge Monitoring methods) publicly available in [18]. The dataset contains the vehicle responses for over half a million numerically simulated vehicle-bridge crossings under a variety of road, bridge, and vehicle conditions. It is designed to support studies that require large amounts of vehicle responses to indirectly evaluate the state of the traversed bridge via data-driven methods. The repository also contains additional material to further support users of the dataset.

The dataset was generated using VBI-2D [19] an open-source tool for MATLAB to simulate the vehicle–bridge interaction (VBI) of road traffic crossing bridges. This tool allows the definition of different vehicle models, road irregularities, bridge properties and structural conditions. In this tool, bridges are represented as beams in a finite element framework and the vehicles as mechanical models with multiple degrees of freedom. The coupled vehicle-bridge response is obtained by direct integration of the coupled equations of motions. For extended descriptions of the numerical model, solution procedure and user manual, the reader is referred to [19]. Figure 1 shows a schematic overview of the simulation capabilities of VBI-2D. In particular for the dataset, simply supported bridge configurations for single vehicles traversing at constant speed have been simulated, for a range of road, vehicle and bridge configurations.

The dataset consists of a collection of separate MATLAB files, each containing information and results for one event. An event is defined here as a single vehicle travelling at a constant speed over a certain road profile while traversing a bridge with no initial vibrations. The dataset has five main problem variables, namely, Bridge length (B), Damage location (DL), Damage magnitude (DM), Vehicle type (V), and Profile (P). The dataset is further divided into A and B (called here DSA and DSB) that correspond to two possible monitoring scenarios. For each possible combination of the problem variables and monitoring scenarios, there are 800 events with randomly sampled vehicle properties. Further detailed information on each problem variable, vehicle properties variability and stored information in the dataset is given next.

The dataset provides the numerical results for six different simply supported bridges. Their span lengths vary from 9 to 39 m, in 6 m increments. The bridges are represented as finite element models with 0.25 m long beam elements. The numerical values of the properties for the beam models are given in Table 3 of Appendix A. In addition, they have been modelled with an elastic modulus of 35 × 10^9 N/m and 3% Rayleigh damping. The values have been chosen to represent typical bridges of those span lengths and are taken from the values reported in [20]. Bridge damage is modelled as stiffness reduction applied to two or four elements, which results in affected bridge lengths of 0.5 m. or 1 m. The magnitude of the damage is expressed in the percentage of stiffness reduction. The dataset includes results for three levels of reduction, which are 0% (healthy bridge), 20%, and 40%. In addition, two damage locations are considered, defined in terms of the bridge span *L*, namely, quarter-span (*L*/4) and mid-span (*L*/2). It is acknowledged that the adopted damage representation is rather simple and represents significant damage. Nevertheless, such damage models have often been used in other past numerical studies to approximate cross-section loss in bridges. More refined bridge models and damage models are certainly preferred in case studies of specific bridges. However, because the aim of the numerical benchmark is to provide signals that capture the main features expected in deteriorating bridges in general, the adopted bridge modelling and damage representation is deemed appropriate.

Regarding the types of simulated vehicles, the dataset includes results for three different vehicle models. The numerical vehicle models are made of combinations of concentrated masses, rigid elements, springs, and dashpots. These models are presented schematically in Fig. 2, which also indicates the names of all mechanical properties, relevant dimensions, and degrees-of-freedom (DOF) notation. The represented vehicles can be characterized by their total number of axles, as 1-axle, 2-axle, and 5-axle models, which have been codenamed as V1, V2, and V5, respectively. The 1-axle vehicle model (V1) is generally referred to as a quarter-car model and has been frequently used in past studies to explore the performance of drive-by methods. The V2 model represents a car or 2-axle heavy vehicle, like a van or truck. The 5-axle model (V5) describes the behaviour of an articulated heavy vehicle, composed of a tractor and a trailer joined by an articulation.

Every event included in the dataset simulates the response of one of these vehicle models. For each simulation, the

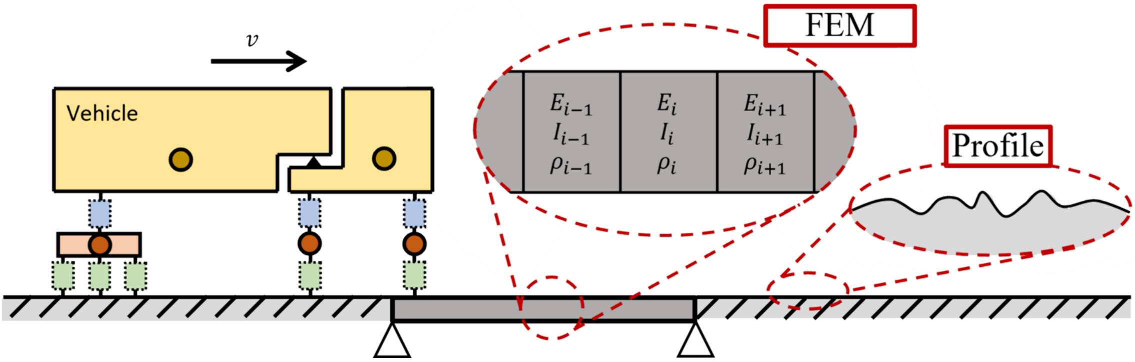

v
Vehicle
FEM
E_{i-1} I_{i-1} ρ_{i-1}
E_i I_i ρ_i
E_{i+1} I_{i+1} ρ_{i+1}
Profile

Fig. 1 Schematic description of VBI-2D model

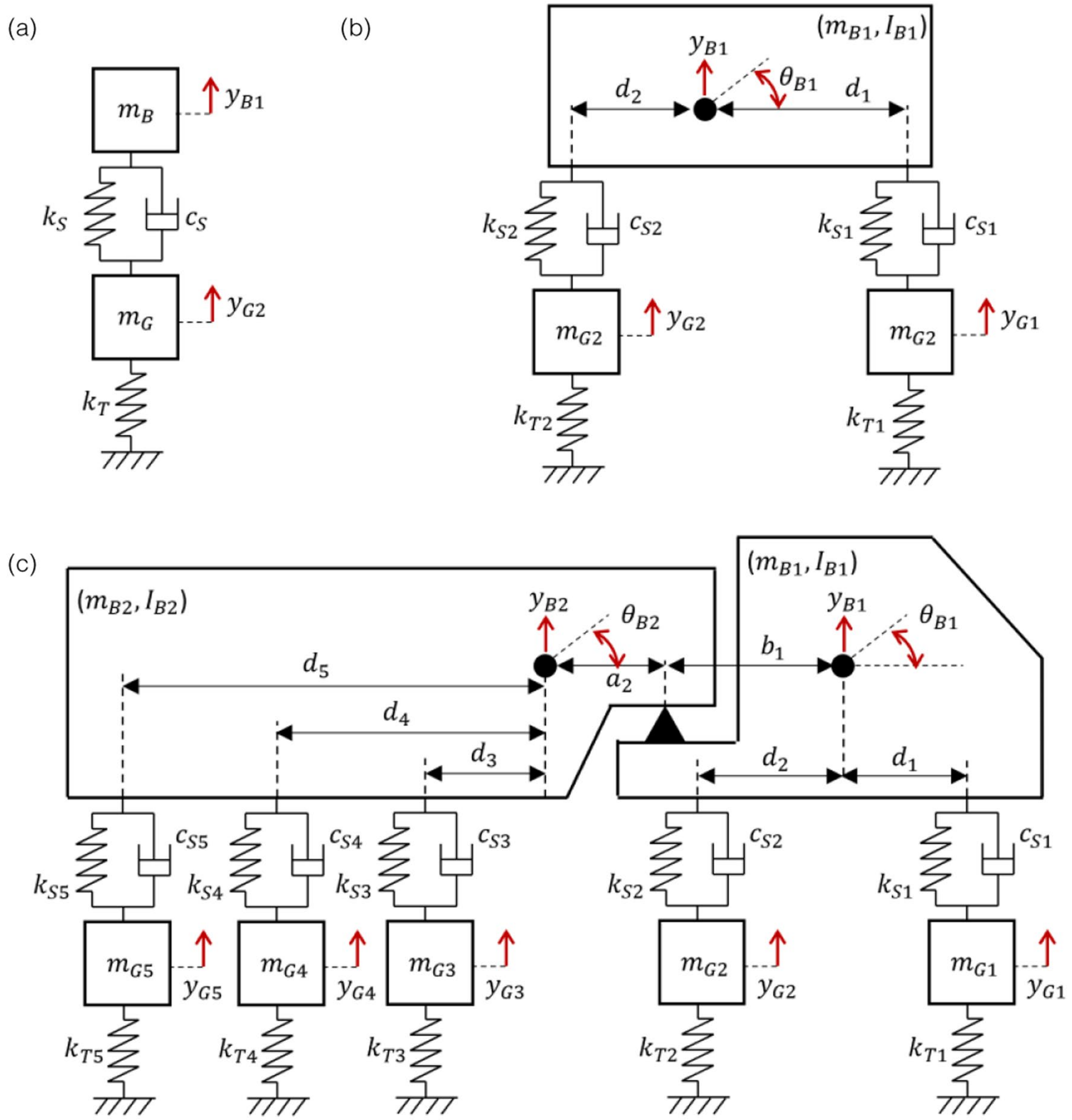


Fig. 2 Vehicle models; **a** V1: 1-axle vehicle model; **b** V2: 2-axle vehicle model; **c** V5: 5-axle articulated vehicle model

actual mechanical and geometrical properties are randomly sampled according to the probabilistic description specified in Appendix A. Thus, the model parameters are defined in Table 4 for V1, in Table 5 for V2, and Table 6 for V5. Note that the variability of vehicle model parameters has been designed to represent fleets of similar vehicles but are not identical. This means that the geometry of all vehicles (for a certain vehicle model type) is the same, while their mechanical properties are varied in each new simulation. This is because in reality no two vehicles are exactly the same, since they have different payloads, suspension properties and tyre pressures. Furthermore, the travelling speed of each simulated event is randomly sampled following a uniform probabilistic distribution. The possible speed values are determined between the minimum and maximum limits defined for each monitoring scenario (Table 1).

The effect of road condition is also included in the dataset. First, some simulations include a perfectly smooth

Table 1 Monitoring scenarios specifications

Dataset code name	DSA	DSB
Number of damaged elements	4	2
Sampling rate of signals (Hz)	1024 (2^{10})	256 (2^8)
Speed variation (km/h)	70–80	30–80

road surface denoted P00. In addition, the dataset includes simulations with two randomly sampled class A road profiles generated according to the standardized procedure in ISO 8608 [21]. In addition, a moving average filter of window size 0.25 m is applied to the created profiles to emulate the actual wheel footprint as suggested in [22]. The generated profiles are 600 m long and are defined with the reference system in the middle. The same profiles are used for all bridge spans, locating the left beam support at the origin of the profile's reference system. All vehicles are simulated including a 100 m long approach distance.

Finally, the dataset considers two possible monitoring scenarios. Scenario A (DSA) represents a situation where the target bridge has significant damages, and the responsible authority has decided to perform more controlled and detailed drive-by inspections, while maintaining normal operational conditions. In this case, the extent of the bridge damage is four elements (1 m long) and the fleet of vehicles gathers signals at high sampling rates while travelling at controlled (very similar) speeds. On the other hand, scenario B (DSB) corresponds to a normal operational condition, where the goal of the authorities is to identify possible incipient bridge deteriorations. In this second case, the damage is only 0.5 m long (two elements), while the vehicles are free to travel at any allowed speed (large variability) and the gathered signals are sampled at a lower rate. Table 1 presents an overview of the specifications for both monitoring scenarios. The two monitoring scenarios represent two levels of difficulty for drive-by methods. DSA has larger bridge damages and less variability in vehicle speeds, aspects that should facilitate the damage detection performances of data-driven methods. On the other hand, DSB represents a more challenging situation.

Table 2 presents an overview of all dimensions of the dataset together with their possible values.

As mentioned before, the dataset is made of individual files containing the numerical results and additional relevant information for single vehicle crossing events. Each file has a unique name that codifies the particularities of the simulated event according to the dataset dimensions (see Table 2). The notation adopted for naming each file is:

$$\mathbf{DS}a + \_ + \mathbf{B}bb + \mathbf{DL}cc + \mathbf{DM}dd + \mathbf{V}e + \mathbf{P}fg + \mathbf{E}hhhh + \ .mat$$

for

DSa = Dataset name, where *a* indicates either A or B monitoring scenario

Bbb = Bridge type, where *bb* indicates span length

DLcc = Damage location, where *cc* indicates the location as a percentage of span length

DMdd = Damage magnitude, where *dd* indicates the percentage of stiffness reduction

Ve = Vehicle type, where *e* indicates the number of axles of the vehicle model

Pfg = Profile, where *f* indicates the class of profile (A), and *g* is the profile number

Ehhhh = Event number, where *hhhh* indicates the event number

As an example, consider the event file *DSA_B15DL50DM20V5PA2E0327.mat*. This corresponds to the event number 327 for a 5-axle truck travelling over the 2nd Class A profile and traversing a 15 m simply supported bridge with a damage of 20% stiffness reduction at the mid-span for the monitoring scenario A.

The total number of events available in the dataset is 518 400 in 52Gb of files openly available to download [18], conveniently divided into compressed subfolders for each bridge span and monitoring scenario. Each event file stores the simulated acceleration responses from all the DOFs of the corresponding vehicle model. These signals include both, off-bridge and on-bridge responses, this is, the vehicle responses while approaching and traversing the bridge. The file also includes the exact time when the vehicle enters the bridge. In addition, the particular realization of vehicle mechanical properties for the simulated event is included together with the corresponding natural frequencies of the vehicle model. The reader is referred to the documents *ToRead.pdf* available in the repository [18] for further information about the stored content and practical guidelines to read the event files.

In addition, the repository contains complementary files that provide information about the vehicle models and their DOF notation (*Vehicles_DOF.zip*), the road profiles used in the dataset generation (*Profiles.zip*), and bridge model parameters (*Bridges.zip*). It is also worth noting, that the

Table 2 Summary of dataset's dimensions

Property	Notation	Possible values
Monitoring scenario	DS + letter	DSA–DSB
Bridge	B + span in m	B09–B15–B21–B27–B33–B39
Damage location	DL + location in % of span	DL25–DL50
Damage magnitude	DM + % of stiffness reduction	DM00–DM20–DM40
Vehicle	V + number of axles	V1–V2–V5
Road profile	P + class + Number	P00–PA1–PA2
Event number	E + number	E0001–E0002–…–E0800

vehicle responses included in the dataset are clean signals. Therefore, these signals should be corrupted with noise to emulate real measurements. To facilitate this, the file *Noise.zip* provides two separate but equivalent implementations (for MATLAB and Python), to add noise to the signals.

In the particular dataset presented here, the authors attempted to find a balanced set of cases among the many possibilities. It was necessary to define a representative number of vehicle models, bridge configurations, damage types, traffic scenarios and road irregularities. While the dataset should cover as many configurations as possible. The total size of the dataset could not be excessive. The provided benchmark is still rather large (52Gb) but has been conveniently divided into smaller files that can easily be handled by today's standard personal computers. For reference, all the simulations included in the dataset were performed with a personal computer (i7-9700 CPU 3 GHz 8 cores) for 80 h of 4 Matlab instances running in parallel.

3 Artificial neural network

3.1 Proposed method

This method uses a data-driven approach which incorporates an Artificial Neural Network (ANN), as depicted in Fig. 3. The algorithm, proposed in [14], is divided into two phases (i) training of the ANN and (ii) condition monitoring. During the training phase, the contact-point (CP) response between the vehicle tire and the bridge surface is used to train the ANN so that it can predict how the frequency spectrum of the CP-response should look for any given vehicle speed. This allows the influence of operational and environmental effects to be learned, so that damage-related changes in the frequency spectrum can more easily be identified. Once trained, the condition monitoring phase of the process uses the trained ANN to predict the Fast Fourier Transform (FFT) of the CP response, for each subsequent vehicle passage. The predicted FFT is compared to that calculated directly from the measured signals for each vehicle passage and an individual damage indicator is calculated for each vehicle passage. As data collection continues over time, an overall Bridge Damage Indicator (BDI) is calculated, by applying a smoothing algorithm to the individual damage indicators, to provide an overall measure of the progression of damage.

3.1.1 Artificial neural network

The ANN used herein is programmed using MATLAB's Deep Learning Toolbox and consists of an input layer with 2 input neurons, 2 hidden layers, each containing 30 neurons, and an output layer with a single output which is the predicted FFT response. Figure 4 shows the ANN architecture.

The two inputs (x_1 and x_2) are those depicted in Fig. 4 (vehicle speed and the x-values of the frequency spectrum), which are fed into the neurons in the subsequent hidden layers using a series of pre-defined weights. An output prediction of the CP-response magnitude for each frequency is made at the end of the sequence. During the training process, the optimal values of these weights are calculated through a learning process such that the difference between the output predictions and the input training data are minimized. A Levenberg–Marquardt backpropagation (LMBP) algorithm is used to train the ANN, with the hidden layers of the ANN containing hyperbolic tangent activation functions and the output layer containing a linear activation function.

3.1.2 Calculating the contact-point response

The response at the point of contact between the vehicle tire and the bridge surface can be inferred from the in-vehicle vibrations, by using a model which describes the relationship between the vehicle vibrations and those at the point of contact. The expression shown in Eq. (1) is used in this

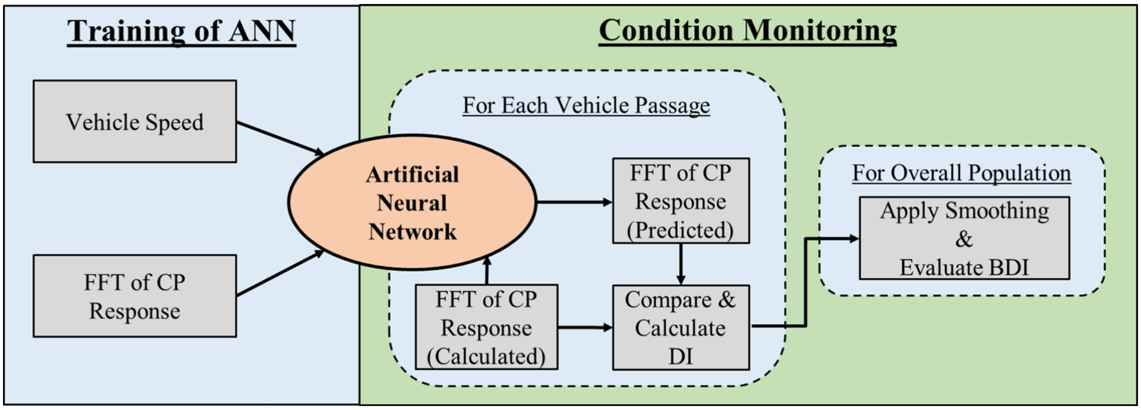


Fig. 3 Data-driven bridge condition monitoring algorithm

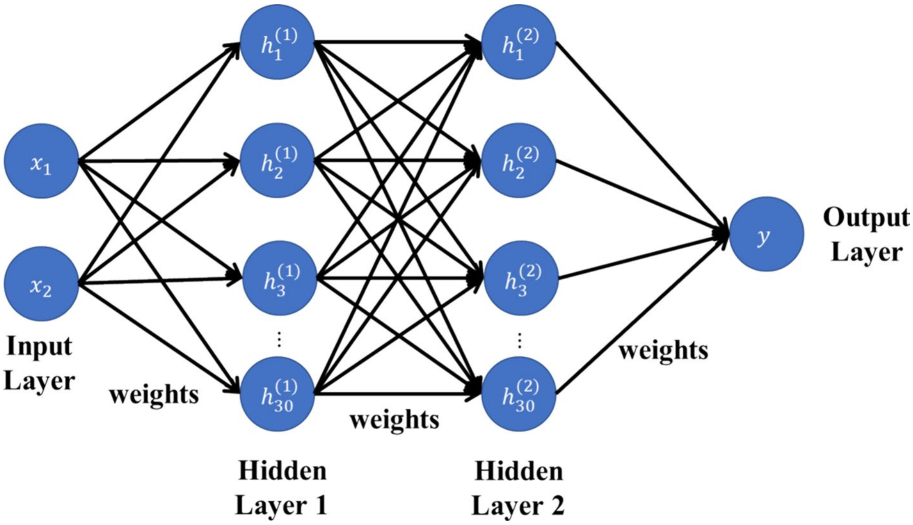


Fig. 4 Proposed ANN architecture

study to infer the CP-response. It has been shown that the CP-response is governed by the bridge frequency and not influenced by the vehicle frequencies, unlike the responses measured directly on the vehicle. More details on the derivation of this formula can be found in [23]. It is noted that it is assumed that the vehicle properties are known with some degree of accuracy in order to allow Eq. (1) to be evaluated. Although the vehicle properties will not necessarily be known with exact accuracy, it has been shown that the FFT of the CP-response is not particularly sensitive to errors in the assumed vehicle properties [14]. Equation (1) allows the CP-response, $\ddot{u}_{cp}$, to be inferred, by idealizing the vehicle as a quarter-car:

$$\ddot{u}_{\mathrm{cp}} = \frac{M_{\mathrm{W}}}{k_{\mathrm{T}}}\frac{\mathrm{d}^2\ddot{y}_{\mathrm{W}}}{\mathrm{d}t^2} + \frac{c_{\mathrm{V}}}{k_{\mathrm{T}}}\left(\frac{\mathrm{d}\ddot{y}_W}{\mathrm{d}t} - \frac{\mathrm{d}\ddot{y}_{\mathrm{V}}}{\mathrm{d}t}\right) + \frac{k_{\mathrm{V}}}{k_{\mathrm{T}}}\left(\ddot{y}_{\mathrm{W}} - \ddot{y}_{\mathrm{V}}\right) + \ddot{y}_{\mathrm{W}} \tag{1}$$

where M_{W} represents the mass of the combined wheel/axle. The stiffness of the suspension is represented by k_{V} and the suspension damping is represented by c_{V}. The tire stiffness at the point of interaction between the vehicle and the bridge is represented by k_{T}. In Eq. (1), $\ddot{y}_{\mathrm{V}}$, $\ddot{y}_{\mathrm{W}}$ represent the measured accelerations on the vehicle body and the wheel/axle, respectively. The $\frac{\mathrm{d}^n\ddot{y}_{\mathrm{V}}}{\mathrm{d}t^n}$ notation is used to represent the nth time derivative of the measured acceleration signals from the vehicle.

3.1.3 Bridge damage indicator

During the condition monitoring phase of the process, an individual damage indicator (DI) is calculated for every single vehicle passage over the bridge. The DI is defined based on the prediction error of the ANN and uses a root mean square approach to calculate the prediction error, for the FFT of the CP-response. The prediction error (pe) for any given vehicle passage, j, is defined as:

$$\mathrm{pe}_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{pred}_i - y_i\right)^2} \tag{2}$$

where pred_i and y_i are the predicted and actual values of the response at sample i, with the squared difference between the two, being summed for all samples n. It is expected that prediction error will vary to some degree across the training data, and as such, a DI, which normalizes the prediction errors with respect to this variation is defined as follows:

$$\mathrm{DI}_j = \frac{\mathrm{pe}_j - \mu_{\mathrm{training}}}{\sigma_{\mathrm{training}}} \tag{3}$$

where μ_{training} and $\sigma_{\mathrm{training}}$ are the mean and standard deviations of the training errors, respectively. Normalising the prediction errors in this way reduces the variation in errors for case when the bridge is undamaged and means that the DI will be close to zero while the bridge is undamaged. While these individual DI s can provide insight into changes in bridge behavior, they can be sensitive to vehicle velocity and other operational factors. As such, there can be noticeable variation in the DI values calculated for individual vehicle passages. This can cause difficulty when trying to detect subtle changes in bridge condition. In order to address this, an overall bridge damage indicator (BDI) is proposed to provide a more robust measure of bridge condition. The BDI is calculated by applying a smoothing technique to the individual DI s and evaluating a smoothed moving average value of the DI s, as described below.

- Choose number of vehicle passages, k, which are to be used for moving average.

- Calculate moving average (μ_{DI}) and moving standard deviation (σ_{DI}) as per Eq. (4).
- Identify DI values which deviate significantly from the mean and adjust them so that they are equal to the mean of the k neighbouring DI values using the following correction:
  - For any value of DI where:

    $$\mathrm{DI} > \left(\mu_{\mathrm{DI}_j} + 0.25\sigma_{\mathrm{DI}_j}\right)\text{or } \mathrm{DI} < \left(\mu_{\mathrm{DI}_j} - 0.25\sigma_{\mathrm{DI}_j}\right)$$

  - Set the value of DI $= \mu_{\mathrm{DI}_j}$
  - Re-evaluate the moving average of the adjusted DI s to obtain final BDI values.

Using this smoothing approach allows the variation in the DI to be removed and the general trend of the DI s over time to be better visualised. Equation (4) provides the formulae for the moving average ($\mu_{\mathrm{DI}j}$) and moving standard deviation ($\sigma_{\mathrm{DI}j}$) values associated with vehicle passage j, considering a moving window of k neighbouring DI values.

$$\mu_{\mathrm{DI}j} = \frac{1}{B-A+1}\sum_{i=A}^{B}\mathrm{DI}_i \sigma_{\mathrm{DI}j} = \sqrt{\frac{\sum_{i=A}^{B}\left(\mathrm{DI}_i - \mu_{\mathrm{DI}j}\right)^2}{B-A}} \quad (4)$$

The parameters A and B are defined in Eq. (5), where N is the total number of recorded vehicle passages and the floor $\lfloor x \rfloor$ and ceiling $\lceil x \rceil$ notation is used to denote values which are rounded down or up to the nearest integer, respectively.

$$A = \begin{cases} 1 & \text{if } j < k - \left[\frac{k}{2}\right] + 1 \\ j - \left[\frac{k}{2}\right] & \text{if } j \geq k - \left[\frac{k}{2}\right] + 1 \end{cases}$$
$$B = \begin{cases} j + \left[\frac{k}{2}\right] - 1 & \text{if } j \leq N - \left[\frac{k}{2}\right] \\ N & \text{if } j > N - \left[\frac{k}{2}\right] \end{cases} \quad (5)$$

3.2 Demonstration of concept

In order to demonstrate the concept and test the ability to identify bridge damage under normal operational conditions, the ANN was trained using the signals from 600 passages of the vehicle V1 across bridge B09, with a smooth pavement surface (P00), for monitoring scenario B (DSB) as defined in Table 1. The signals were polluted with 5% Gaussian noise, to simulate measurement noise which would be experienced in practice. The frequency spectrum of the CP-response, in the range of 8–10 Hz was used as the input to the ANN as this range contained the peak associated with the first natural frequency of the bridge. Once trained, the DIs were calculated for 300 vehicle passages. The first 100 passages, were over the undamaged bridge, followed by 100 passages over the bridge for damage case DM20 (20% loss of stiffness at mid-span), and another 100 vehicle passages for damage case DM40 (40% damage at mid-span). Figure 5a shows the DIs for each vehicle passage, using different symbols to depict the DIs associated with each of the three bridge conditions. The overall smoothed BDI considering $k = 50$ vehicles in the moving average, is also shown by the solid

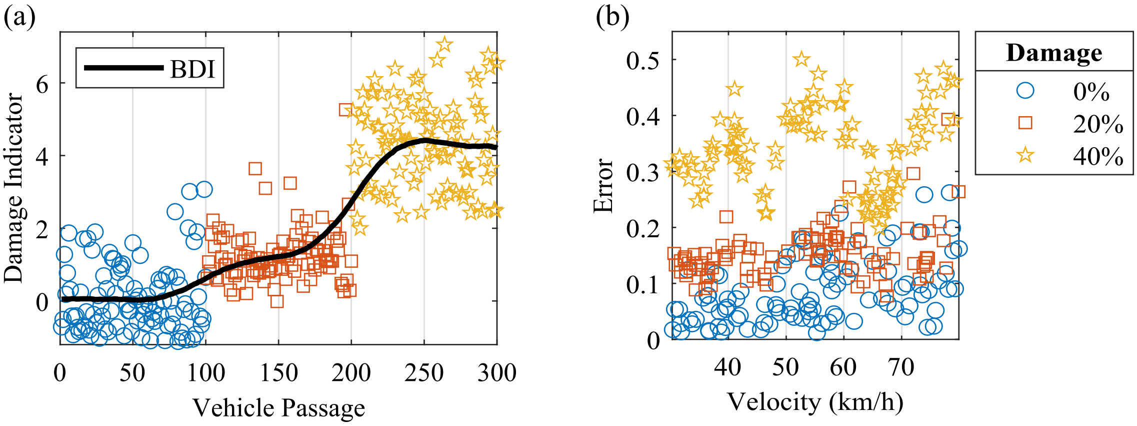


Fig. 5 Demonstrating the approach for vehicle V1 crossing bridge B09 for mid-span damage (DSB) **a** individual DIs and overall BDI and **b** prediction errors vs. vehicle speed

black line. Figure 5(b) shows the ANN prediction errors, as per Eq. (2), vs. vehicle velocity, where it can be seen that the damage cases can be separated quite well at all vehicle speeds considered, indicating that the ANN can remove the influence which vehicle speed has on the measured frequencies.

3.3 Results

In order to test the ability of the method to detect damage for the more realistic scenario where the pavement roughness is included, the ANN was trained using 800 passages of vehicle V1 across the bridge B09 in its healthy condition. Again, the frequency spectrum in the range 8–10 Hz was used as the target output for the ANN and the signals were polluted with 5% Gaussian noise. The damage detection capabilities of the algorithm were then tested by calculating the BDIs for 2400 subsequent vehicle passages (800 across the healthy bridge, and 800 with 20% or 40% damage), this time using $k = 75$ vehicles in the moving average. Figure 6 shows the results, for both mid-span and ¼ span damage on the bridge.

Figure 6a shows the results for the case where there is no pavement (P00) and Fig. 6b shows the results when the class A pavement, PA2, is considered. It is clear that for the idealised case of no pavement, the algorithm can clearly identify the presence of damage at both locations. For the case where the pavement is included, in Fig. 6b, the sensitivity to damage is less distinct. Mid-span damage of 20% and 40% can still be readily identified, however, for the case of damage at the ¼ span, the 20% damage case is not clearly visible.

Previous research has shown that the influence of vehicle velocity on drive-by damage detection capabilities can be significant, so the algorithm was applied again, this time separating the vehicle passages into low- (30–55km/h) and high- (55–80km/h) speed categories. Figure 7 shows the results, where it is observed that for mid-span damage (Fig. 7a), the low-speed passages do show increased sensitivity to the 40% damage case, but for the 20% damage case

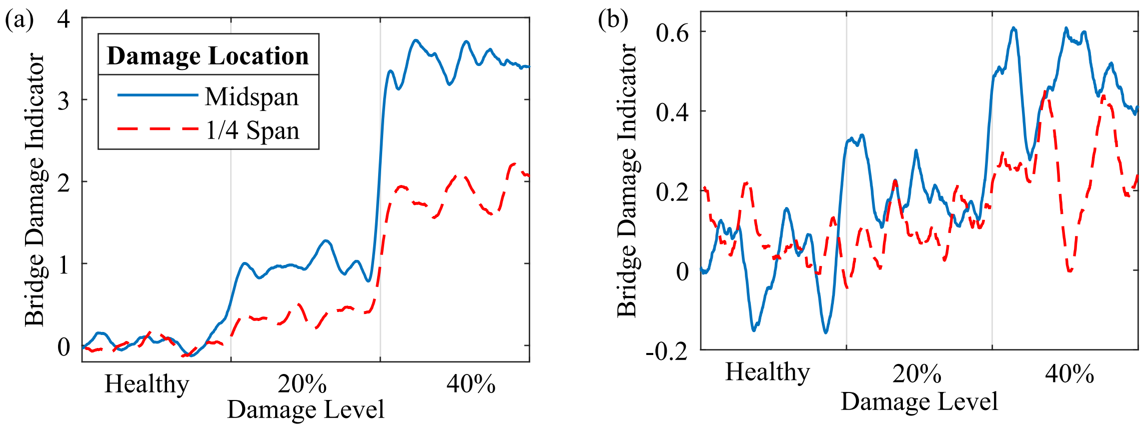


Fig. 6 BDI for vehicle V1 crossing bridge B09 (DSB) with pavement **a** P00 and **b** PA2

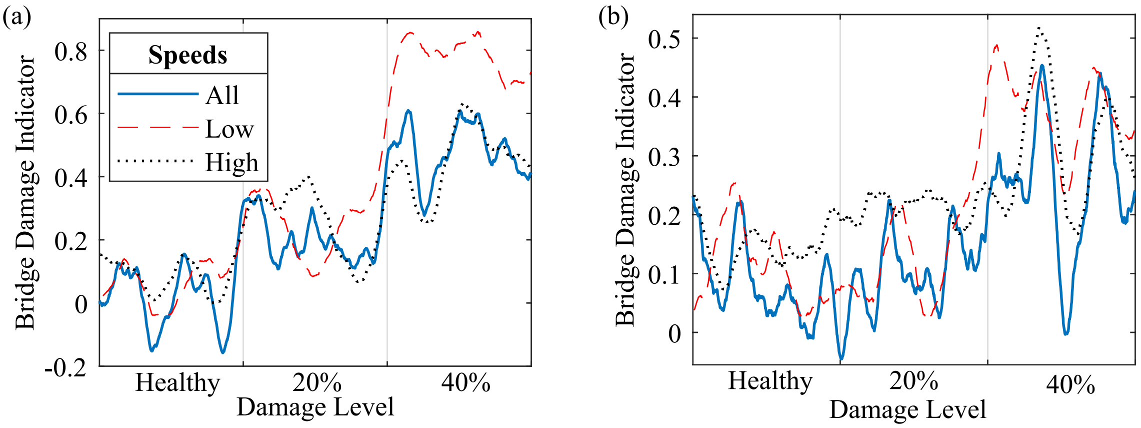


Fig. 7 BDIs for different vehicle speeds: vehicle V1 crossing bridge B09 (DSB) with pavement PA2 for damage at **a** mid-span, **b** ¼ span

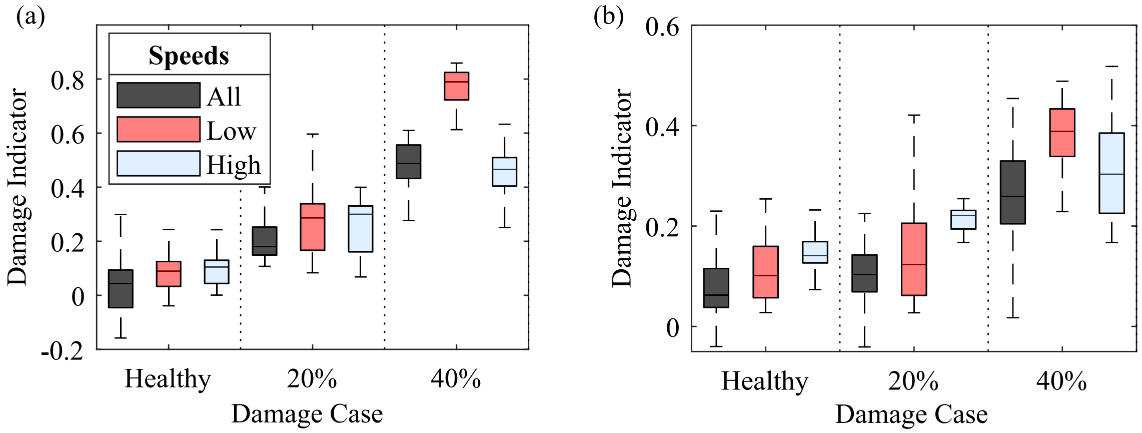


Fig. 8 Box plots of BDIs for different vehicle speeds for damage at **a** mid-span, **b** ¼ span (DSB)

the results are similar. The high-speed vehicle passages are very similar to the case when all vehicle speeds were considered. In the case of ¼ span damage (Fig. 7b), the results show that the 40% damage case is slightly more distinct when using either low- or high-speeds, however the 20% damage case is still difficult to identify.

Finally, Fig. 8 presents box plots of the BDI values to allow a more distinct comparison. In reality, the damage cases would need to be known to allow these box plots to be created, however it is useful here to examine the sensitivity of the BDI to damage. Each box plot shows the interquartile range (between the 25th and 75th percentile) of the BDI values using a shaded box. The horizontal line within the box represents the median value of the BDIs and the whiskers extend to the most extreme values in the data, which are not considered to be outliers. The differences in the BDIs can now clearly be separated, and even for the case of ¼ span damage, it is seen that there is a distinct increase in the BDI values, particularly when using high-speed vehicle passages.

4 Deep autoencoder model

4.1 Proposed method

The proposed method's overall framework consists of three main steps. Firstly, acceleration responses are collected from multiple sources and the signals are pre-processed. In the next step, these vehicle responses are used to train a deep learning model that defines the baseline condition. The trained deep learning model is used to reconstruct the response of subsequent events and compared to the actual measurements. In the last step, prediction errors are assessed via Kullback–Leibler (KL) divergence to estimate the distance between two distributions as a damage indicator. An overview of the proposed framework is presented in the Fig. 9.

4.1.1 Deep autoencoder model

An autoencoder is a deep learning model that operates on an unsupervised basis to perform dimensionality reduction and feature extraction. The proposed framework obtains a compressed feature representation of multiple vehicles'

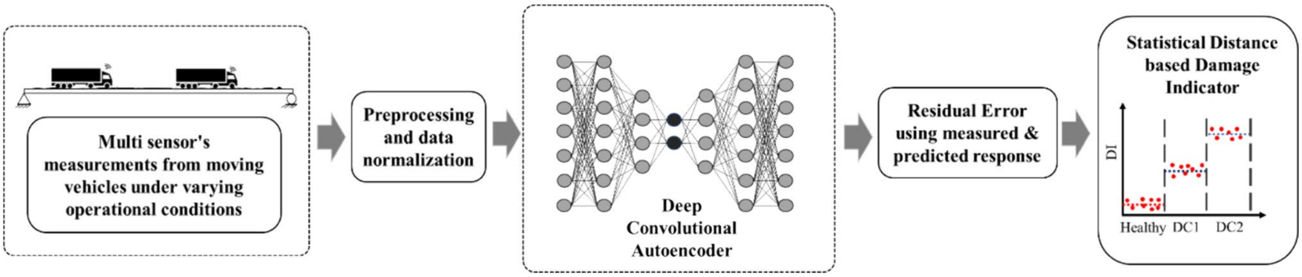

Multi sensor's measurements from moving vehicles under varying operational conditions
Preprocessing and data normalization
Deep Convolutional Autoencoder
Residual Error using measured & predicted response
Statistical Distance based Damage Indicator
DI
Healthy DC1 DC2

Fig. 9 Overview of the proposed method

acceleration responses, which can then be employed for robust damage detection as discussed in [15]. The architecture of an autoencoder is divided into two primary components: (1) an encoder that compresses or transforms the input data into a low-dimensional space that describes the input data, and (2) a decoder that employs the low-dimensional feature representation obtained from the encoder to reconstruct the original input data.

The Deep Autoencoder architecture (DAE) was used in this study to obtain a condensed hidden representation of the training dataset. Compared to previous work in [15], this proposed DAE considers measurements from three different locations of the 5-axle vehicle (DOFs 1, 4, and 5) instead of just one. The proposed model accurately reconstructs the input data and is particularly sensitive to damage information. The encoder's hidden layers are split into two levels. The first level consists of multiple convolutional blocks that extract local features and reduce parameters through pooling layers. The Leaky-ReLU activation function is used at this level to add non-linearity. To retain temporal dependencies of features and obtain a robust latent representation, the first-level feature map is then fed into a second-level Long Short-Term Memory (LSTM) layer. This layer learns the temporal dependencies of similar features for extracting a smooth latent space. The last LSTM layer is flattened and mapped to the bottleneck layer for a fixed latent space representation. The decoder consists of deconvolutional layers followed by up-sampling and Leaky-ReLU activation function in each convolutional block. The proposed architecture is optimized for loss function through an end-to-end method for weight and bias parameters, rather than the stepwise training of hidden layers and stacking of pre-trained layers. Figure 10 illustrates the architecture of the DAE. For more details about the model's hyperparameters (activation function, filters, and kernel size), refer to [15].

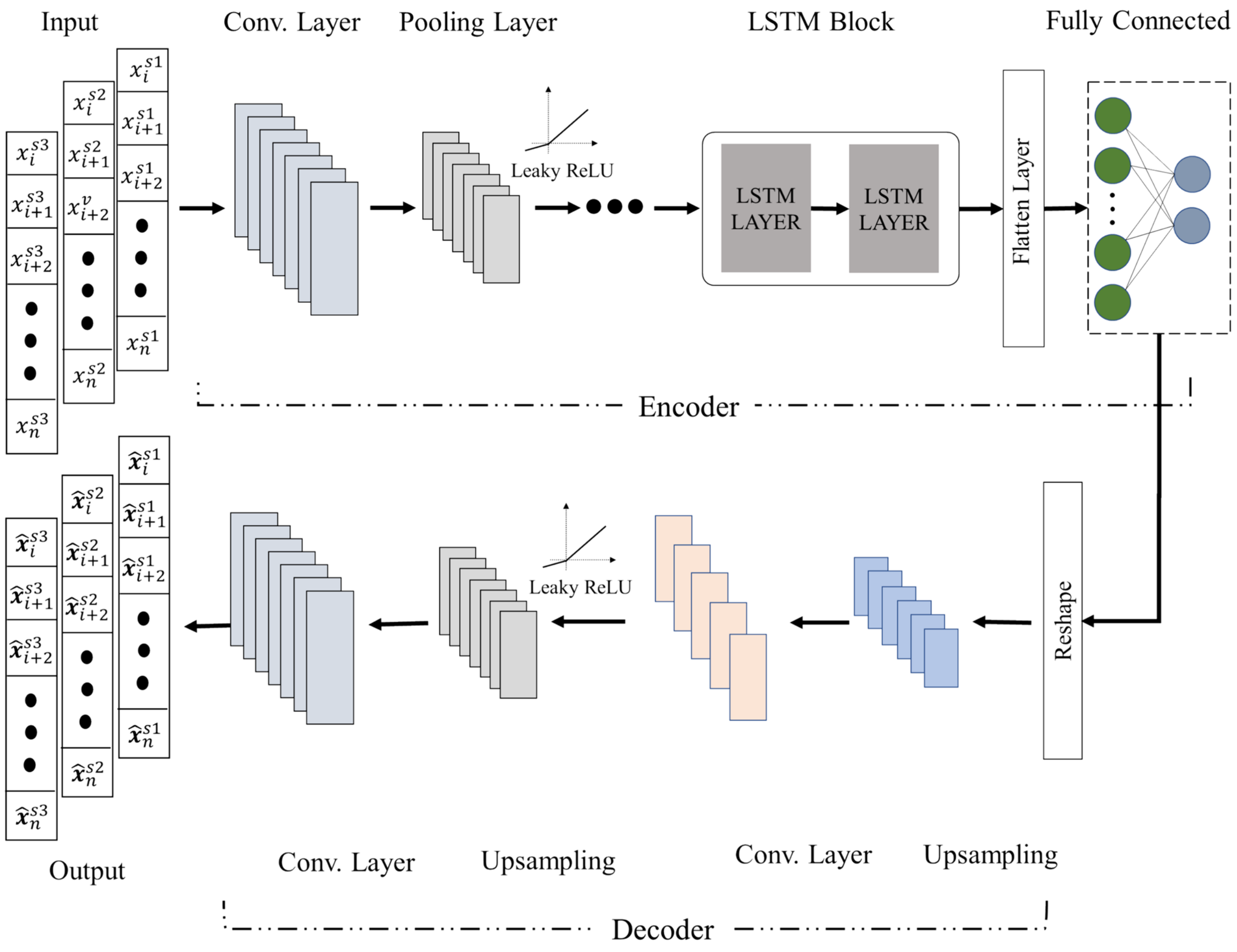


Fig. 10 Architecture of the proposed deep autoencoder model (DAE)

4.1.2 Damage index

To detect damage and quantify its severity, the mean absolute error (MAE) is used to evaluate the reconstruction loss. The MAE in Eq. (6) computes the difference between the measured acceleration and the reconstructed response, estimated by the trained DAE model, for each sensor and vehicle passage.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{x}_k\left(t_i\right) - x_k\left(t_i\right)\right| \tag{6}$$

where $\hat{x}_k\left(t_i\right)$ and $x_k\left(t_i\right)$ are the reconstructed and measured responses, respectively, at sample i for a total of n samples for each sensor k.

In the case of a fleet of vehicles, when each vehicle has different speed and properties, the MAE error is found to vary significantly. However, when multiple events are grouped together in batches, the MAE values follow a statistical distribution. This statistical distribution for each batch of vehicles can be used to differentiate between a healthy and a damaged state. The Kullback–Leibler (KL) divergence is used in this study to calculate the variation between two distributions. The study assumes that the MAE for multiple sensors follows a multidimensional Gaussian distribution for each batch of vehicles, where the KL divergence can be determined by Eq. (7).

$$D_{\text{KL}}\left(p_0 \| q_1\right) = \frac{1}{2}\left[\ln\frac{\left|\Sigma_1\right|}{\left|\Sigma_0\right|} + \text{trace}\left(\Sigma_1^{-1}\Sigma_0\right) + \left(\mu_1 - \mu_0\right)^T\Sigma_1^{-1}\left(\mu_1 - \mu_0\right) - k\right] \tag{7}$$

where $\mu_0$and μ_1 are the mean matrices and Σ_0 and Σ_1 are the covariance matrices for the baseline and unknown condition, respectively, and k refers to the number of sensors. From Eq. (7), it is evident that the KL divergence is exponentially related to the distance between distributions. For damage detection, the equation is mapped to a linear relationship, as shown in Eq. (8), where e is Euler number.

$$\text{DI} = \ln\left[D_{KL}\left(p_0 | q_1\right) + e\right] - 1 \tag{8}$$

4.1.3 Dataset and signal processing

The numerical evaluation of the proposed method is performed using the dataset results for bridges B09 and B15, traversed by vehicle type V5 and road profiles P00 and PA2, for the monitoring scenario DSB. The method processes the vertical accelerations from the tractor's main body and axles (see Fig. 2c). Because each event has different speed, these signals have different durations. In order to obtain uniform signal lengths, the signals are resampled into the spatial domain by means of the vehicle's speed. Moreover, additional processing is applied to the signals to remove the dynamic effect of road profile before converting them into spatial domain.

The road profile has significant impact on the vehicle response and the measured accelerations. The road-induced vibrations mask the component of the bridge response in those signals. In previous studies, different methods were used to eliminate the effect of road profile from vehicle acceleration signals, but they had limitations [24]. Therefore, there is a need for a reliable and efficient method that can automatically extract the bridge dynamic response from sensors in passing vehicles. To address this challenge, the authors used maximal overlap discrete wavelet packet transforms (MODWPT) [25] to remove the road profile component from the vehicle's vertical acceleration signals. MODWPT uses a wavelet filter to divide the signal into wavelet components of narrow band frequencies. For a given signal, MODWPT generates $2n$ equivalent wavelet components, each with a passband range of $\text{Fs}/(2n + 1)$ for a sampling frequency Fs and level number n. MODWPT also partitions the energy at each wavelet component and adds up the energy across all wavelet components, which equals the total energy of the input signal.

When a vehicle passes over a bridge, the wavelet component that contains the bridge's response can be identified by computing the residual response of the wavelet energies from the first and second axle. During a single vehicle passage, both the first and second axle measure the bridge's response and the excitation from the road profile. If the component containing the bridge's frequency content can be identified in the signal, then those wavelet components can be selected to isolate the bridge's response in the measured signals. To isolate the bridge's response from a vehicle traveling at a speed v, the MODWPT with $n = 8$ levels is applied to the first and second axle response (DOFs 4 and 5). The component containing the bridge's response in both axles always has a similar energy magnitude. When the residual of the energies between the two responses is computed, the components with energy near to zero are summed together for reconstructing the acceleration signal of each DOF. Figure 11a displays the power spectral density response before applying the MODWPT, where the first and second modes of the bridge are not clearly visible. However, after applying MODWPT, as shown in Fig. 11b, the peak of the first two bridge modes is clearly distinguishable. Therefore, it is evident that applying MODWPT can effectively isolate the bridge's response in the vehicle's signals.

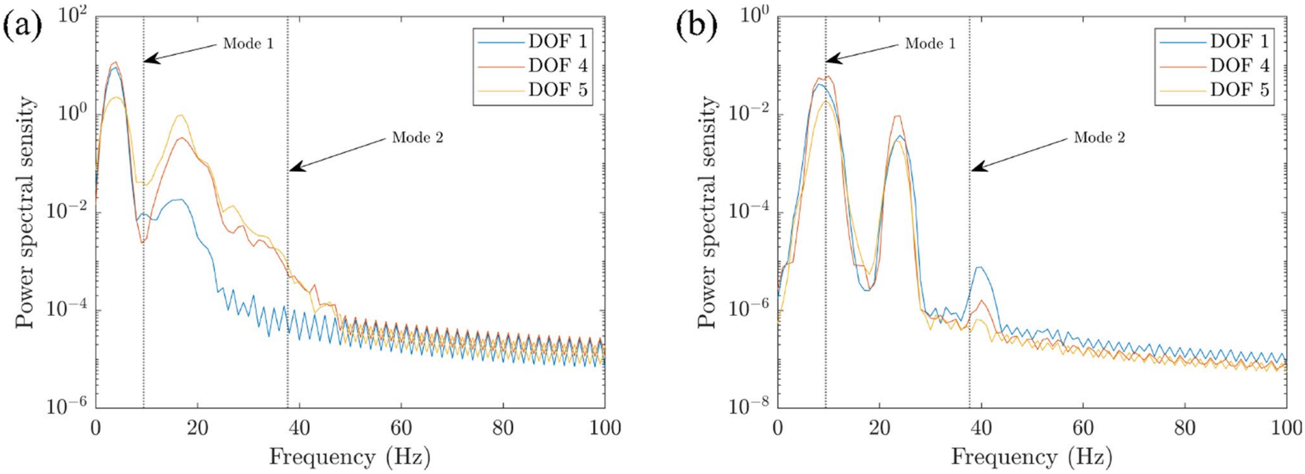


Fig. 11 Pre-processing example: **a** Power spectral density of DOFs before applying MODWPT; **b** Power spectral density of DOFs after applying MODWPT

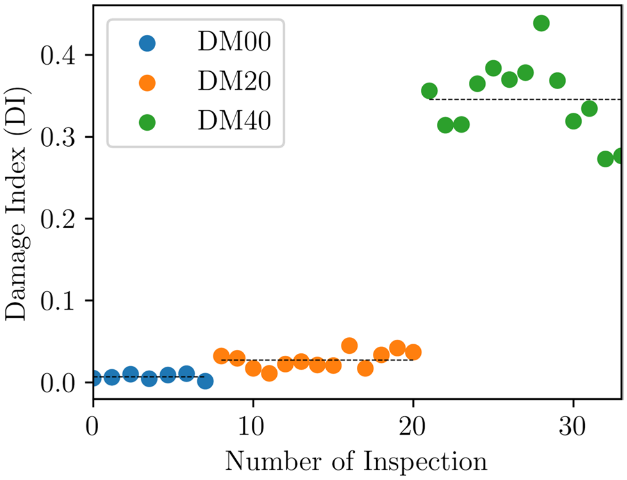


Fig. 12 Evolution of daily damage index (60 events/day) during progressive bridge condition change for B09. Dotted line indicates average value

4.2 Results

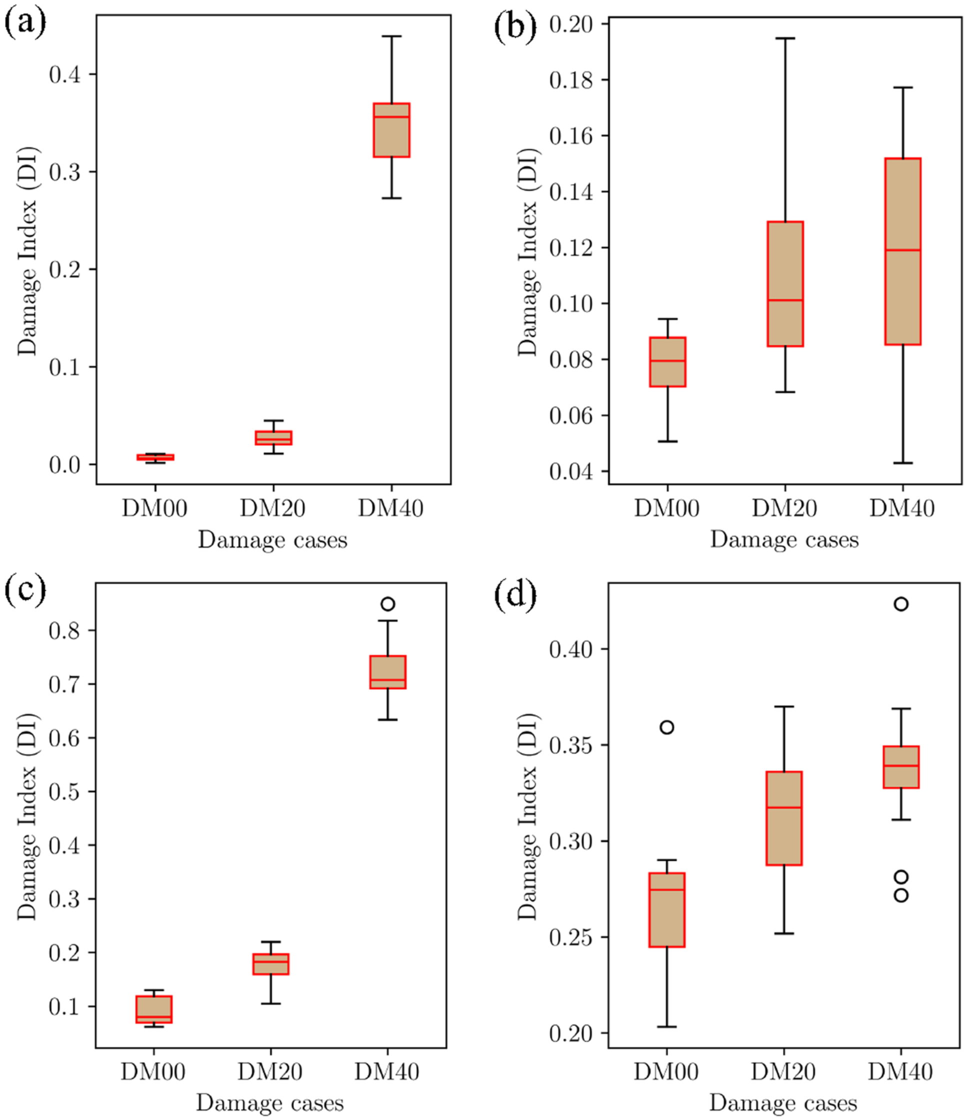
This section demonstrates a method for detecting damage using vehicle responses from a fleet of similar vehicles. The method relies on a proposed DAE model, which is trained using healthy responses from a dataset (as described in [15]). The model is trained with 800 vehicle events and validated with 200 events, using similar architecture and hyperparameters for each case.

To illustrate how the model can be used for damage detection and quantification, three different damage scenarios (DM00, DM20, and DM40) are studied, with damage located at the mid-span (DL50) of bridges B09 and B15, for vehicle type V5 on road profiles PA00 and PA2. For every inspection, 60 vehicle crossing events are assumed to be available, and the severity of damage is increased progressively. Figure 12 shows that the damage index (DI) values are distinctly different for different bridge conditions, with the magnitude of DI increasing with damage severity. However, the magnitude of DI displays some variation for a given bridge state, due to operational conditions and different vehicle properties. Nevertheless, for any given bridge condition, the average value of DI remains constant.

Furthermore, to compare the performance of the proposed damage assessment for two different bridges (B09 and B15) with road profiles P00 and PA2, a box plot representation is used (Fig. 13). Each box in the plot shows the middle 50% of the data, with the bottom of the box representing the 25th percentile (Q1) and the top representing the 75th percentile (Q3). The line in the middle of the box represents the median value (Q2). The range of the data that is not considered an outlier is shown by the black lines, which extend to the minimum and maximum values of the data that fall within Q1-1.5 × IQR (Inter Quartile Range) and Q3 + 1.5 × IQR, respectively. Any data points that fall outside this range are considered outliers. By using box plot representations to compare the damage assessment performance of different damage cases, it is possible to identify any significant differences in the distribution of multiple inspections.

Figure 13 shows a comparison of the damage index (DI) for two bridges with different road profiles, namely P00 and PA2. The results indicate that when there is no road profile (P00), the proposed method can clearly separate each damage case with increasing severity, as shown

Fig. 13 Damage index performance comparison for different cases: **a** B09 with no profile; **b** B09 with profile PA2; **c** B15 with no profile; **d** B15 with profile PA2



in Fig. 13a, c. However, when the road profile is present (PA2), the performance of the method is relatively poor, as shown in Fig. 13b, d, compared to the P00 case. The degradation in performance in the presence of road profiles is mainly due to the pre-processing of acceleration signals, which removes higher modes of the bridge frequency that are generally more damage sensitive. Nevertheless, the results still demonstrate that the DI increases with damage severity for both cases. To improve the robustness of damage detection in the presence of road profiles, it is important to increase the fleet size of vehicles and the number of inspections, which can help to clearly separate different damage cases. Overall, the proposed method shows potential for use in a wide range of vehicles and bridge configurations for long-term road bridge monitoring.

5 Uniform manifold approximation and projection (UMAP) combined with hierarchical density-based spatial clustering of applications with noise (HDBSCAN)

5.1 Proposed method

A practical approach to the problem of drive-by bridge inspection should be formulated based on an unsupervised learning framework. In particular, this section will demonstrate a methodology based on the work of Cheema et al. [16], which took a topologically driven approach to the problem. This relies centrally upon the notion of "ε-balls", as they appear in two areas of machine learning: uniform manifold approximation and projection (UMAP) [26], and hierarchical density-based spatial clustering of applications with noise (HDBSCAN) [27].

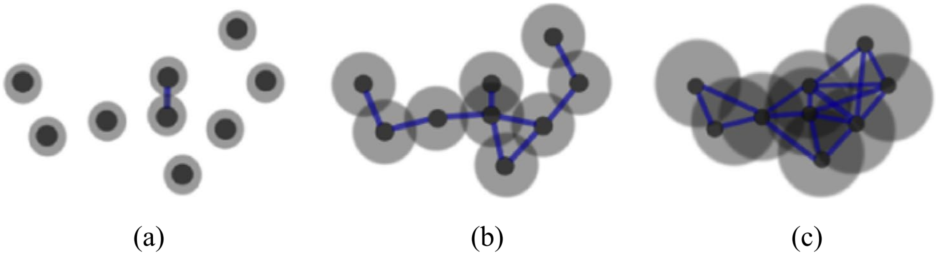


Fig. 14 The growth in the connectedness of a simplicial complex as derived by one's choice of ε. The image is based on [16]. **a** Simplicial complex with low ε level. **b** Simplicial complex with moderate ε level. **c** Simplicial complex with large ε level

This is because ε-balls allow one to better understand how data points are connected, in low and high-dimensional space. A brief overview of each method is provided in the following subsections. The code to reproduce the graphs found in this section is available at: https://github.com/pche123/VBI_UMAP_HDBSCAN (See also in Table "7" in Appendix B).

5.1.1 Uniform manifold approximation and projection (UMAP)

An ε-ball, defined with respect to the i-th input data point, $\boldsymbol{X}_i \in \mathcal{X}$, is the set of all data points, $\boldsymbol{X}_j \in \mathcal{X}$, that are within a ε radius from $\boldsymbol{X}_i$. This is clarified through Eq. (9).

$$\mathcal{B}_i(\varepsilon) = \left\{ \boldsymbol{X}_j \in \mathcal{X} \middle| d\left(X_i, X_j\right) \le \varepsilon \right\} \tag{9}$$

Particular to UMAP, these ε-balls are in fact designed locally over each data point, in the sense that each data point, X_i has its own local radius, ε_i, so that really one is working with $\mathcal{B}_i\left(\varepsilon_i\right)$, instead of $\mathcal{B}_i(\varepsilon)$. Further, in UMAP, the choice of distance function, d, is flexible (for example a standard choice is the Euclidean distance). In order to simplify the choice of which radius to pick locally, ε_i, for each $\boldsymbol{X}_i$, UMAP works with the hyperparameter, k, which is the number of nearest neighbors around each point to consider [26]. This allows ε_i to change locally on a point-to-point basis (dense regions of points will have a smaller ε_i, as opposed to sparse regions of points). The purpose of doing this is that it helps to construct a notion of connectivity between the data points via a structure known as a simplicial complex. A diagrammatic example of this structure is shown in Fig. 14, albeit it is simplified because a constant ε is defined over all the data points. A more precise treatment on this topic is available at [26].

Once a simplicial complex is constructed in the high dimensional data space, the UMAP algorithm will attempt to project it to a low dimensional space, whilst maintaining connectedness. It does this by minimizing a term known as fuzzy set cross-entropy, which requires the mathematics of fuzzy set theory. This is necessary due to the locally varying nature of the radii of the ε-balls. Further technical discussion on this topic is available at [16].

In terms of algorithmically working with UMAP, there are several hyperparameters to consider as follows. n_neighbors: which defines how many neighbours to consider as a valid connection (this is what changes the radius locally on each data point); min_dist: the level of separation to enforce between the points in the lower dimensional space; and finally, n_components: which is the size of the embedding dimension. Additional hyperparameters used specifically for the analysis in this paper are that of op_mix_ratio, and metric_UMAP. The former is a term that controls trade-off between the intersection and union operations in the UMAP algorithmic process [16]. More details about this parameter can be found in [26]. The latter usually defaults to "Euclidean metric", but in practice, when working in higher dimensions the "Manhattan metric" is generally preferable due to its correspondence to the L_1 norm.

5.1.2 Hierarchical density-based spatial clustering of applications with noise (HDBSCAN)

HDBSCAN is a non-parametric clustering algorithm meaning that the total number of clusters is dynamically learned based on the input data. It builds upon the previously successful DBSCAN algorithm by combining the properties of the density-based approach of DBSCAN, with the inclusion of the properties inherent in hierarchical clustering approaches [27]. Similar to UMAP, it works by constructing ε-balls. However, unlike UMAP, there is a constant radius applied over the input data. The size of this radius is determined by specifying a parameter known as min_cluster_size; which is the minimal amount of data points to consider as being part of a cluster [27].

The HDBSCAN algorithm was found to be an effective clustering algorithm in an earlier study on drive-by bridge inspection [16]. This is because HDBSCAN addresses two key modelling assumptions often required for conventional clustering approaches. Namely, (i) How many clusters there are in total, and (ii) What the shapes of the clusters are Point (i) is naturally addressed since HDBSCAN only requires the specification of a min_cluster_size hyperparameter. Considering point (ii), conventional use of algorithms such as the k-means under a Euclidean distance implicitly assumes a

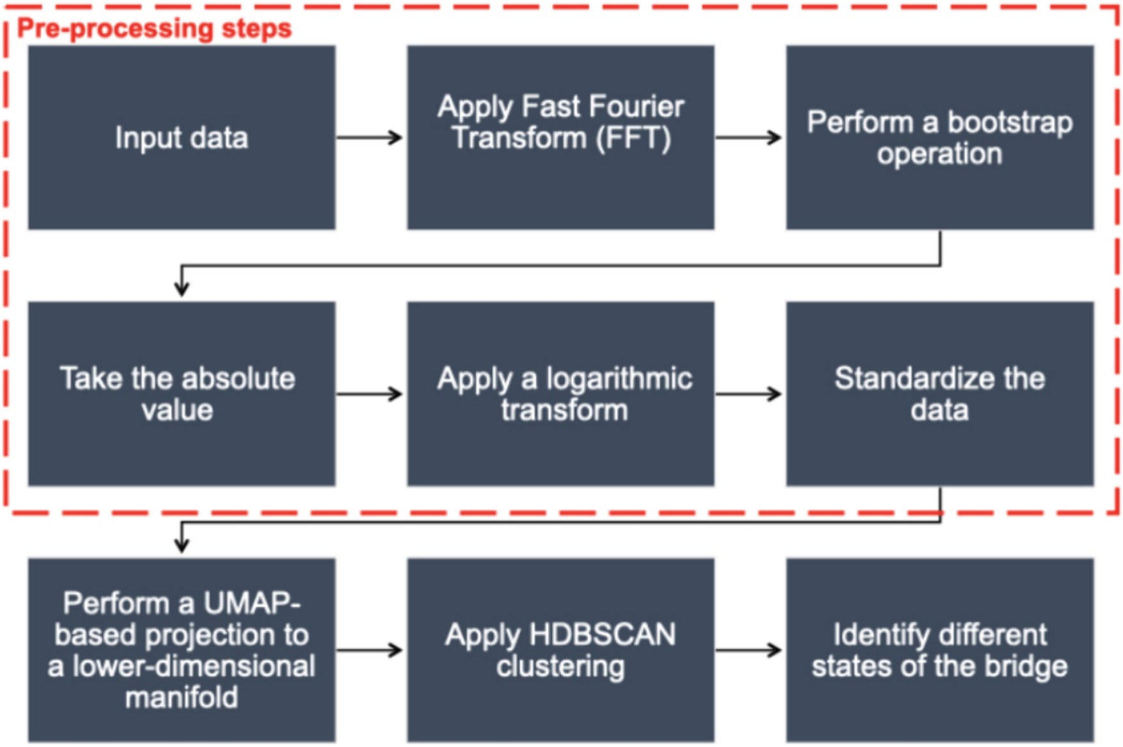

Pre-processing steps
Input data
Apply Fast Fourier Transform (FFT)
Perform a bootstrap operation
Take the absolute value
Apply a logarithmic transform
Standardize the data
Perform a UMAP-based projection to a lower-dimensional manifold
Apply HDBSCAN clustering
Identify different states of the bridge

Fig. 15 Illustration of the data pipeline flowchart

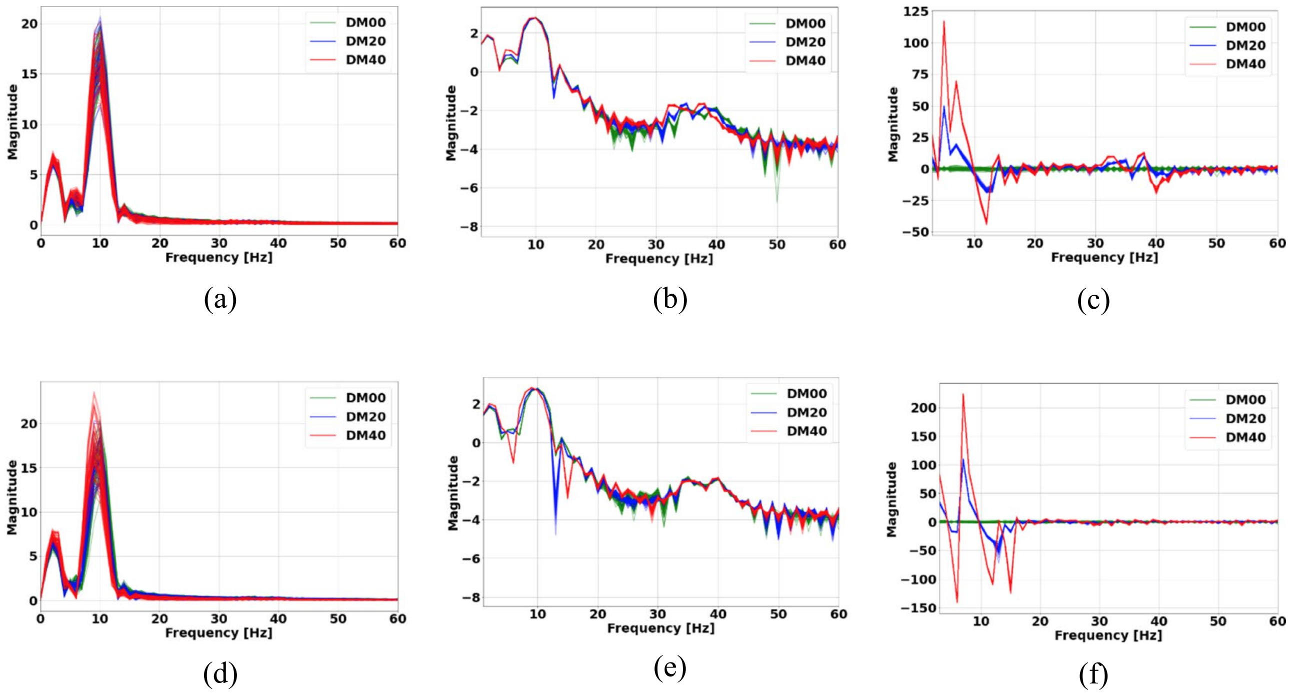
spherical shape for the clusters, which in a practical setting is not always expected to occur, however, HDBSCAN does not make such an assumption. More details on this can be found in [16].

5.2 Methodology

In order to apply the mathematical theories described above, the data is passed through an engineering pipeline as shown in the flowchart depicted in Fig. 15. The steps closely parallel the original work of Cheema et al. [16] in which first data pre-processing steps are carried out followed by UMAP projection to transform data from high dimension to low dimension, at which point an HDBSCAN clustering operation is performed.

An additional step is taken in this study to apply a bootstrapping-like process during the data pre-processing. Bootstrapping is a technique that has been applied in statistical literature since the 1980s in order to try and infer standard errors of an estimand in relation to some population distribution given that only a subset sample from the population is observed [28]. It achieves this by sampling with replacement many times, and then aggregating the samples to generate a statistic (such as the mean, or median). A similar bootstrapping principle is used here, albeit with a slightly different end goal in mind. Due to the large degree of random variations between each observable data point (varying car properties, with varying velocities, compounded with random road profiles), it was found that such a pre-processing procedure was necessary over the input data so that UMAP projections could consistently map new input points to the same clusters in which these points belong to. Further, it has been shown that even in non-iid (independent and identically distributed) settings, bootstrapping offers a reasonable, and conservative estimate for confidence intervals and statistics [28]. This is beneficial for the case of drive-by bridge inspection which often deals with non-iid settings.

As illustrated in Fig. 15, before projecting the data using UMAP, a standardization procedure is performed (subtract the mean, and divide by the variance). Typically, such procedures are often performed when it is expected that the underlying data set follows a normal distribution in a limiting (population) sense. Here, however, it is not expected that the input data stream will necessarily be Gaussian in nature. Rather, the purpose of the standardization procedure here (relative to the healthy data set), is that it provides a useful means by which data in different groups can be further separated (in particular aided by the division in variance between each group). Due to this, however, it should be noted that in principle, any appropriate data scaling method could be used in practice, and it will be dependent upon the reader's problem context, and their modelling assumptions.



(a) (b) (c)

(d) (e) (f)

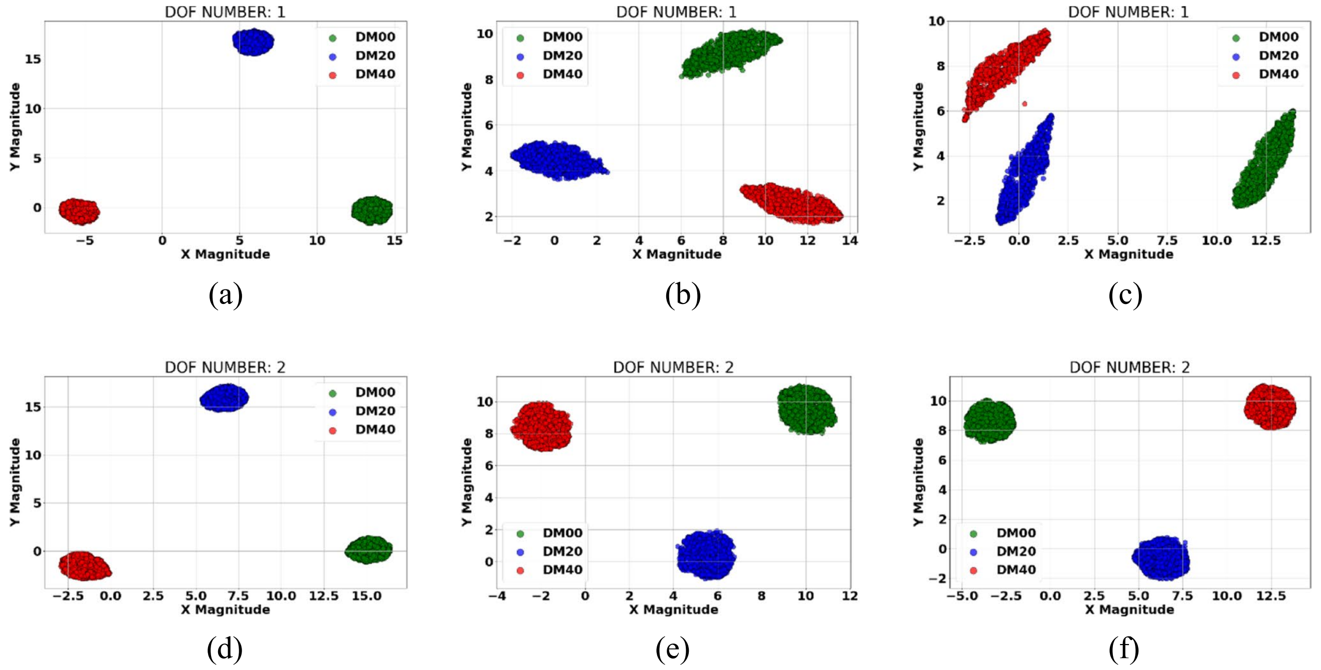
Fig. 16 Progressive application of the pre-processing feature engineering steps in order to demonstrate maximal separation between 50 randomly selected events for cases DM00, DM20, and DM40, respectively. The images were based on: B09, V1, P00, for the second Degree of Freedom (DOF 2), and pre-processing was shown at the two damage locations, DL25, and DL50, respectively. **a** Application of FFT for DL25. **b** Bootstrap mean aggregation and logarithm transform for DL25. **c** Standardization with respect to the healthy data for DL25. **d** Application of FFT for DL50. **e** Bootstrap mean aggregation and logarithm transform for DL50. **f** Standardization with respect to the healthy data for DL50

In order to demonstrate the effectiveness of the proposed data processing pipeline, shown in Fig. 15, the dataset from monitoring scenario DSA, bridge B09, vehicle V1, and road profile P00 was investigated. In particular, three bridge states including, DM00, DM20, and DM40, were studied. Two different cases of damage at quarter-span, DL25, and damage at mid-span, DL50 were analysed. The impact of the proposed pre-processing procedure is made clear in Fig. 16. Notice that in Fig. 16 the pre-processing step is presented for one Degree of Freedom (DOF) (in particular, DOF 2 of vehicle model V1 in this case). This is because, in the current data pipeline set-up, the outlined procedure is intended to work on a per-DOF basis, meaning that pre-processing, projection, and clustering is applied independently for each DOF. Also, one may notice that in Fig. 16, there is no 0 Hz axis value for subfigures (b), (c), (e), and (f). This is because a slight offset is taken away from 0 Hz, in order to eliminate computational issues which, arise from taking the logarithm of small epsilon values near 0 during the pre-processing.

Furthermore, Fig. 16 makes it clear that the proposed data pre-processing pipeline works effectively independent of damage location, in that it is effective for both: DL25, and DL50, for the given vehicle, bridge, and road profile set-up. Finally, as per standard data analysis, there is a requirement for training, validating, and testing in the overall pipeline. For the demonstration of this algorithm for the provided data set, a grid search was performed offline. Moreover, the best overall hyper-parameters were found such that they work well across a large variety of demonstrable settings involving bridge B09, and vehicle V1. However, as the problem increases in complexity (larger bridge spans, coupled with increasing vehicle DOFs), it becomes necessary to learn the optimal hyper-parameters for each specific problem setting.

5.3 Results

This section will focus on analysing bridge B09, vehicle V1, and road profiles P00, PA1, and PA2 under the proposed data pipeline. Further, for every setting, the same hyper-parameters for the algorithm have been used. For UMAP, these include, n_components, n_neighbors, min_dist, op_mix_ratio, and metric_UMAP which are, respectively, set to 2, 30, 0, 0.9, and 'Manhattan'. For HDBSCAN, the hyper-parameters include min_cluster_size which is set to 60, and metric_HDBSCAN which is set to 'Euclidean'. For the bootstrap procedure, the only hyper-parameter is sample_times which is set to 1000. The exact interpretation of these hyper-parameters can be found throughout the reference [16].



(a) (b) (c)

(d) (e) (f)

Fig. 17 Clustering results for bridge, B09, damage location, DL25, vehicle, V1, for road profiles PA00, PA1 and PA2. **a** Road profile P00, DOF1. **b** Road profile PA1, DOF1. **c** Road profile PA2, DOF1. **d** Road profile P00, DOF2. **e** Road profile PA1, DOF2. **f** Road profile PA2, DOF2

The results of this case study for damage at quarter-span, DL25 and damage at mid-span, DL50 are, respectively, made clear in Figs. 17 and 18. These figures independently show the results for the three road profiles considered (P00, PA1 and PA2), as well as for the two DOFs of vehicle model V1 investigated (DOF1, and DOF2). Further, three different damage scenarios are studied: DM00, DM20, and DM40.

From Figs. 17 and 18, it can be noticed that in all cases investigated, different bridge states are successfully separated from one another. Note that these successful results have been obtained in a fully unsupervised setting and without specifying any cluster number. From these results, it can also be observed that while a PA2 road roughness is considered, the obtained clusters for DOF1 become elongated which is more dominant for the case of DL25 compared to DL50 (see Fig. 17b, c). Thus, it is important that a clustering algorithm such as HDBSCAN is used here, since methods such as k-means clustering, and Gaussian mixture models, which tend to implicitly assume a spherical shape for the clusters, would probably fail [16]. Finally, if one wants to extend the framework to other bridges, and vehicle models, the previous, highly general hyperparameter settings may no longer be appropriate and a more bespoke training, testing, and validation procedure should be undertaken.

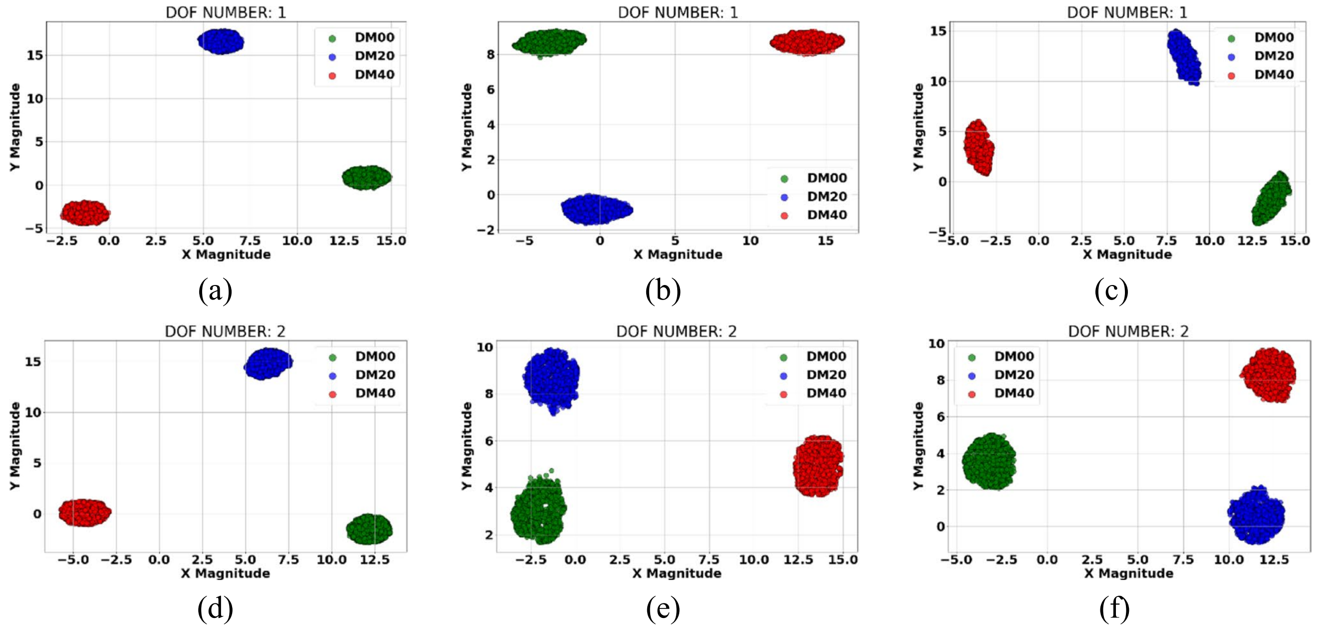
Overall, these promising results demonstrate the effectiveness and capability of the proposed unsupervised framework by the combined use of UMAP, HDBSCAN and the proposed data pre-processing pipeline which has a high potential to advance the field of drive-by bridge inspection in practice.

6 Hierarchical multi-task unsupervised domain

6.1 Introduction

Drive-by bridge health monitoring (BHM) is a scalable approach as it allows a vehicle to monitor multiple bridges as it drives by them, using the vehicle vibration responses. However, the effectiveness of this method can be hindered by the diverse properties of bridges, which can result in distribution shifts in the extracted damage-sensitive features from the vehicle vibrations even for the same damage state. Consequently, models that are trained for one bridge may not work effectively on another bridge due to these shifts in data distribution [15]. Training a model for the new bridge is time-consuming and costly as it requires labelled data for the new bridge.

To this end, Hierarchical Multi-task Unsupervised Domain Adaptation (HierMUD), diagnoses damage of multiple bridges while eliminating the need to collect data labels from every bridge [17]. A bridge with available labelled data is referred to as a source bridge (or the source domain) while the bridge without labelled data is referred to as a target



(a) (b) (c)

(d) (e) (f)

Fig. 18 Clustering results for bridge, B09, damage location, DL50, vehicle, V1, for road profiles PA00, PA1 and PA2. **a** Road profile P00, DOF1. **b** Road profile PA1, DOF1. **c** Road profile PA2, DOF1. **d** Road profile P00, DOF2. **e** Road profile PA1, DOF2. **f** Road profile PA2, DOF2

bridge (or the target domain). The present method diagnoses the damage on the target bridge by extracting features that are invariant across multiple bridges (bridge-invariant) while remaining sensitive to various damage states (damage-sensitive). Additionally, bridge damage diagnosis generally consists of multiple tasks, such as detection, localization, and severity quantification. These tasks have distinct shifted distributions between the source and target bridges and different learning difficulties. To overcome this challenge, multiple tasks are categorized into easy-to-learn and hard-to-learn tasks based on their prediction performance for the source bridge data [29]. Then, this method formulates a feature hierarchy to allocate more learning resources to improve the performance of the hard-to-learn tasks.

6.2 HierMUD for multiple bridge diagnosis

In this subsection, we present the HierMUD method that predicts damage information of a target bridge by adapting the source bridge damage diagnosis model to the target bridge. This method uses labelled data from the source bridge and only unlabelled data from the target bridge. The HierMUD model has three components: hierarchical feature extractors (orange blocks), task predictors (blue blocks), and domain classifiers (red blocks) as shown in Fig. 19. The feature extractors extract damage-sensitive features from the input signal. The task predictors predict the task labels from the learned features. The domain classifiers distinguish whether the extracted feature is from the source bridge or target bridge. Domain classifiers and feature extractors are trained in an adversarial way to extract damage-sensitive and bridge-invariant features [30]. The model is optimized in such a way that domain classifier cannot distinguish whether the features come from the source or the target bridge, while preserving the information of various damage states.

Further, to learn multiple tasks with distinct learning difficulties, the hierarchical feature extractors allocate more learning resources to hard-to-learn tasks. As mentioned above, we first categorize the multiple tasks into easy-to-learn tasks and hard-to-learn tasks. In our problem, the damage detection and localization tasks are considered easy-to-learn tasks; and the damage quantification task is considered a hard-to-learn task, based on their performance using source bridge data. Then, the feature extractors learn two-level features: task-shared and task-specific features. For easy-to-learn tasks, task-shared features are extracted from the input signals and then, for hard-to-learn tasks, task-specific features are further extracted from the task-shared features. By extracting the deeper task-specific feature representation, more learning resources are allocated to learn hard-to-learn tasks, which improves overall performance. Once the HierMUD model is learned during the training phase, unlabelled data from the target bridge is input into the learned model for predicting their damage state.

6.3 Evaluation of HierMUD

In this section, we describe vehicle-bridge interaction data used for our evaluation, preprocessing of the data, setup for the HierMUD model, and its performance.

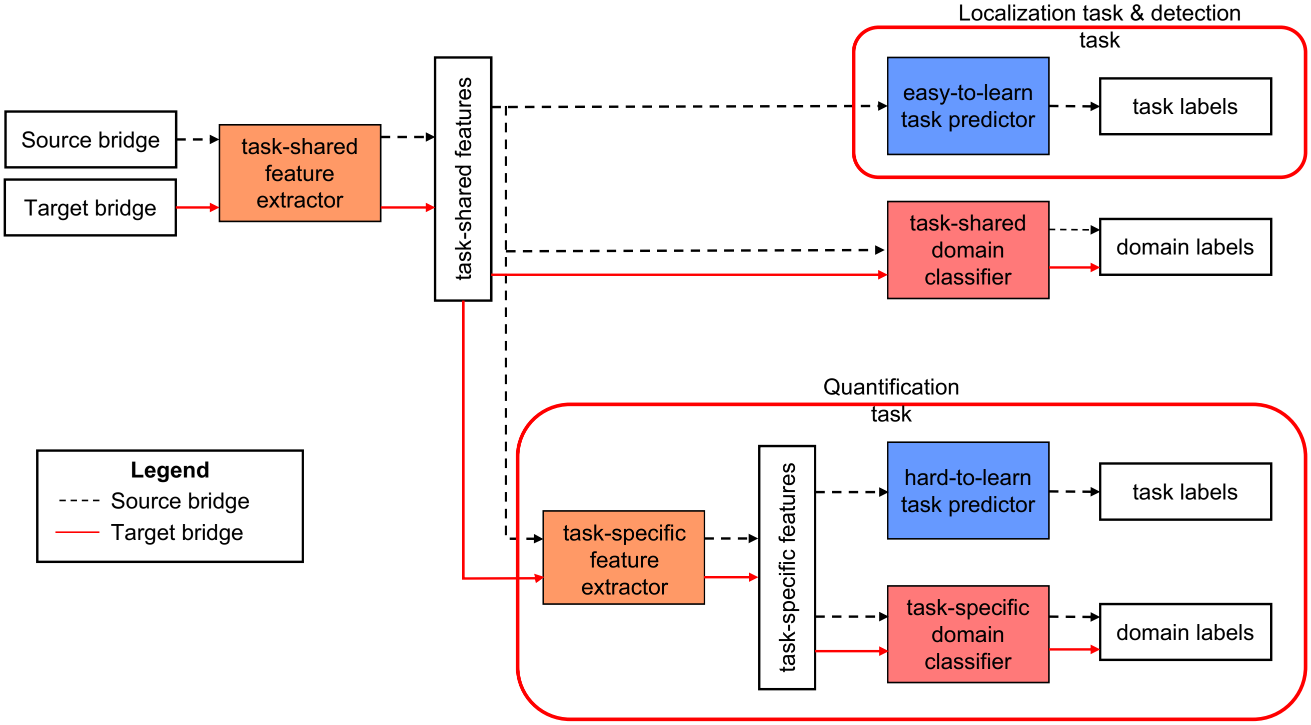

Localization task & detection task
Source bridge
Target bridge
task-shared feature extractor
task-shared features
easy-to-learn task predictor
task labels
task-shared domain classifier
domain labels
Quantification task
Legend
Source bridge
Target bridge
task-specific feature extractor
task-specific features
hard-to-learn task predictor
task labels
task-specific domain classifier
domain labels

Fig. 19 Architecture of our Hierarchical Multi-task Unsupervised Domain Adversarial Learning (HierMUD) Algorithm. Black and red lines depict the data flows for source and target bridges, respectively

6.3.1 Simulated vehicle–bridge interaction data description

The evaluation of our learning model utilizes a subset of the DSB monitoring scenario available at the repository [18]. This sub-dataset includes vehicle vibration data for two bridge lengths (33 m and 39 m), one vehicle type (V1), and one road profile (P00). To assess the model's performance, we consider two damaged locations (25% and 50% of bridge length) for each bridge length, and for each damage location, we consider three damage severities (0%, 20%, and 40% damage). Our dataset consists of 2 (bridge length) × 1 (vehicle type) × 1 (road profile) × 2 (damage location) × 3 (damage severity) = 12 (inspection scenarios). To evaluate the robustness of the model, there are 400 simulated events for each inspection scenario with varying vehicle speed and dynamic vehicle properties. In total, our dataset has 12 (inspection scenarios) × 400 (events) = 4800 (data samples). Both the body acceleration and axle acceleration signals of the vehicle are used to detect the damage.

6.3.2 Vehicle vibration data preprocessing

To have the consistent length for the model input and reduce the data variability due to vehicle properties, the raw signal is pre-processed, as shown in Fig. 20, before it is fed into the learning model. In the first step of our data pre-processing, the signal is chopped from the datapoint where the vehicle reaches the start of the bridge to the datapoint where the vehicle reaches the end of the bridge. This is because in the simulated dataset, the vehicle is made to run over the approach before it reaches the bridge, and vehicle vibration data contains bridge damage information when it is over the bridge.

Secondly, due to varying vehicle speeds, time-domain vehicle vibration data have different lengths. Thus, we

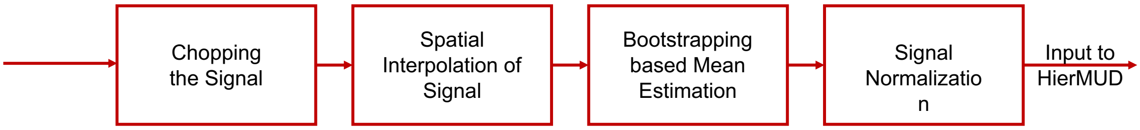

Chopping the Signal
Spatial Interpolation of Signal
Bootstrapping based Mean Estimation
Signal Normalizatio n
Input to HierMUD

Fig. 20 Schematic diagram of the data pre-processing

interpolate each vehicle vibration data at N equally spaced locations along the bridge using spline interpolation [31]. In this way, we ensure the consistent length of the input to the HierMUD model.

Further, we introduce a bootstrapping-based mean estimation to reduce the data variability due to varying vehicle properties, such as different suspension systems, vehicle weights, moving speeds, and so on. For each inspection scenario, 400 events are randomly divided into 250 training events and 150 testing events subsets. Then, we generate 600 new training samples and 150 new testing samples by averaging 100 randomly sampled events with replacements from the training and testing events subset, respectively. Based on the central limit theorem, averaging the bootstrapped samples reduces vehicle vibration data variation due to varying vehicle properties by the square root of the number of random samples, which is 100 in this case [32]. Note that the number of random samples is determined empirically to best reduce the vehicle property variations, and the number of events in training and testing event subsets should be greater than the number of random samples.

Finally, the mean estimated data are normalized based on zero mean and unit standard deviation to help the data-driven model learn faster and leads to faster convergence. Additionally, to avoid the overfitting or biased training problem in the learning algorithm, the data are augmented by introducing a white noise. We chose 30 dB as signal-to-noise as it introduces moderate level of noise while keeping the information in the signal informative [33]. The pre-processed data is then fed into the HierMUD model.

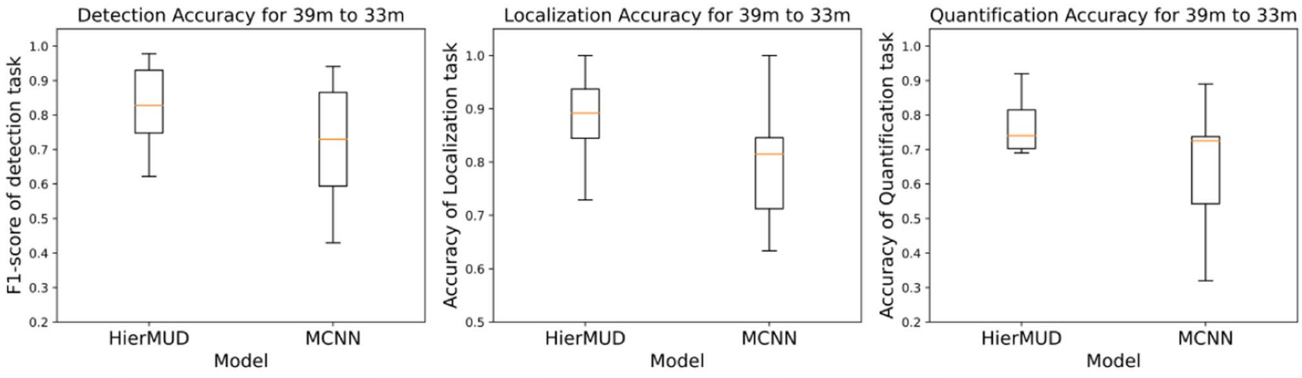
6.3.3 Setup for HierMUD

This subsection describes the architecture of our model and hyperparameters. We have two easy-to-learn tasks (damage detection, damage localization) and one hard-to-learn task (damage quantification). Our model is optimized using stochastic gradient descent (SGD) with weight decay and L2 regularization is added to loss to avoid the overfitting problem on training samples [34]. We randomly generated ten datasets and ran the experiment ten times. Hyper-parameters are selected empirically such that the learning rate is set as 0.0025 and hyperparameters for localization loss, quantification loss, domain classifier loss for localization task, and domain classifier loss for quantification task are selected as 1, 1, 0.1, and 0.5, respectively. A batch size of 100 was chosen for learning. Further, the model was made to run for 300 epochs.

6.3.4 Performance evaluation of HierMUD

The predicted damage diagnosis results from our model are compared against a baseline method. We compare the performance of HierMUD for damage detection, localization, and quantification with a baseline method, MCNN, which is a multi-task convolutional neural network model that uses the source bridge data as the training dataset and the target bridge as the testing dataset without employing domain adaptation. The architecture of MCNN remains the same as HierMUD except for the domain classifiers layer not being included. This comparison shows the effectiveness of domain adaptation for diagnosing damage across multiple bridges.

Figure 21 shows the prediction performance of HierMUD against the baseline method (MCNN) for damage detection, damage quantification, and localization tasks. The results show that our model outperforms the baseline method by



(a) Detection (b) Localization (c) Quantification

Fig. 21 Comparison between HierMUD and the baseline method (MCNN) for **a** damage detection, **b** localization, and **c** quantification for the 33 m target bridge when considering the 39 m as source bridge. **a** Detection. **b** Localization. **c** Quantification

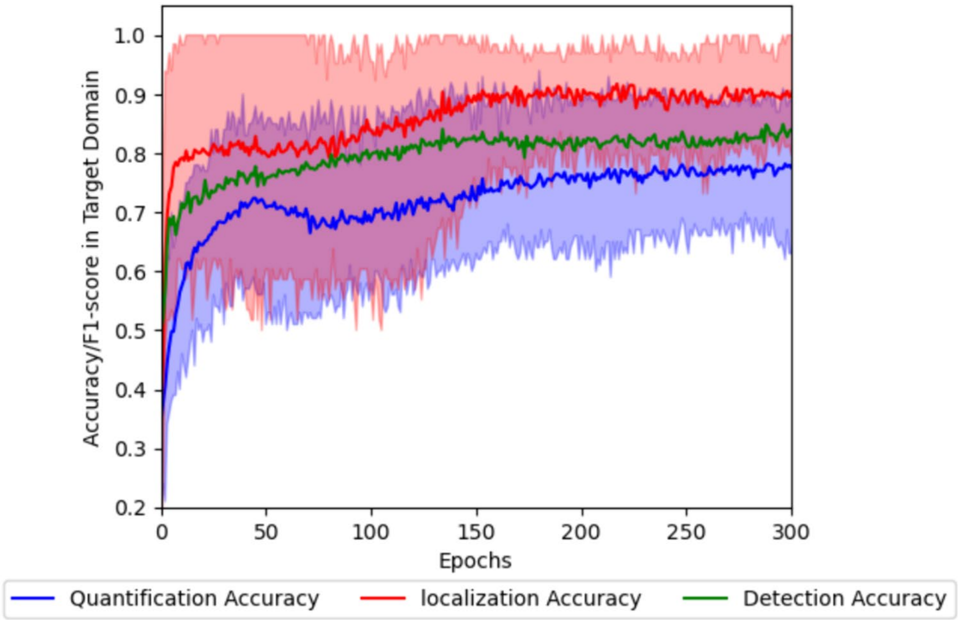


Fig. 22 Damage detection, localization, and quantification accuracy for the 33 m target bridge when considering the 39 m as source bridge

15% in terms of F1-score for damage detection, 10% in terms of accuracy for damage localization, and 20% for damage quantification. Furthermore, our method is more reliable as it has smaller performance variations (10% less variation in localization and detection accuracy and 50% less variation in quantification accuracy) across the ten experiments compared to the baseline method.

Figure 22 presents the F1-score of damage detection (green curve), the accuracy of damage localization (red curve), and damage quantification (blue curve) in the target bridge with an envelope for the ten experiments when predicting the damage on 33 m bridge (target bridge) and considering the 39 m bridge as source bridge. This figure shows that there is a reduction in accuracy at around 50 epochs before it starts increasing again. This reduction is due to adversarial learning, where the model tries to find the optimal trade-off between domain invariance and damage sensitivity. Moreover, compared to the quantification task,

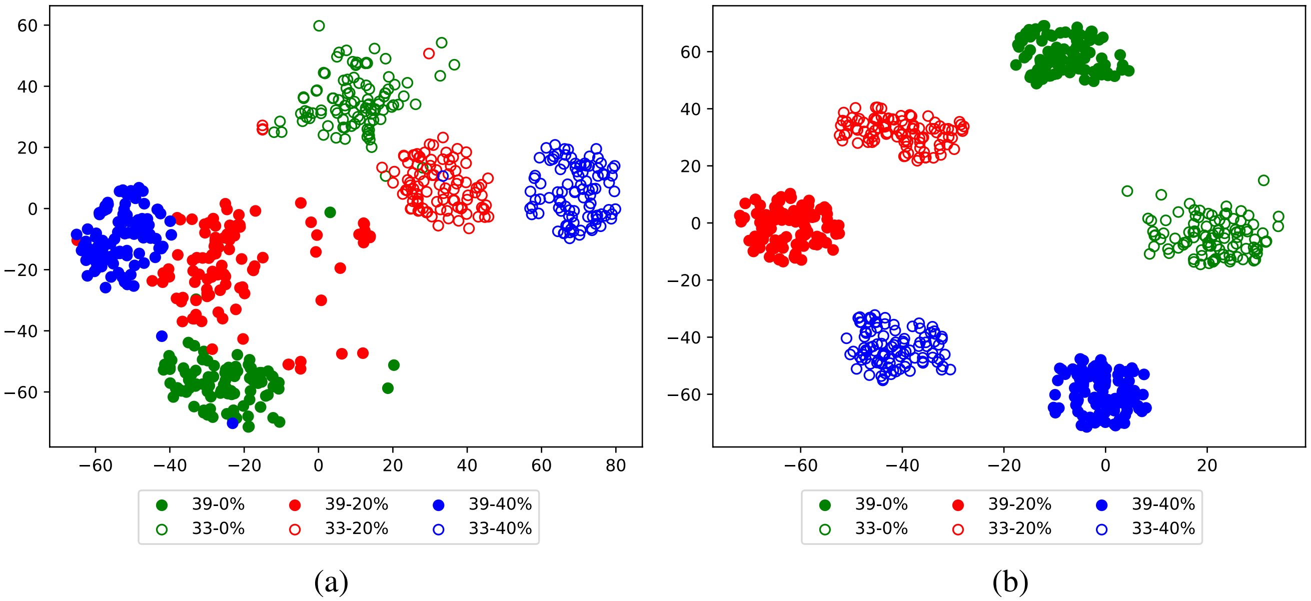


Fig. 23 t-SNE embedding plot of various damage states for damage at mid-span for 33 m (target bridge) and 39 m (source bridge) using **a** pre-processed vehicle vibration data and **b** task-specific features extracted from the HierMUD model

the final accuracy for the localization task is higher and its envelope becomes smaller around 150–200 epochs, which implies that the localization task is easy-to-learn.

To show the effectiveness of domain adaptation on matching data distributions of the extracted features, we visualize the task-specific features of both before and after the model training in the low-dimension embedding using t-SNE [35]. Figure 23 shows the t-SNE embedding plots of the pre-processed vehicle vibration data (Fig. 23a) and the task-specific features (Fig. 23b) extracted from the HierMUD model. Green markers represent no damage on the bridge, red markers represent 20% damage at mid-span, and blue markers represent 40% damage mid-span. Filled markers represent the 39 m bridge (source bridge) features while unfilled markers represent the 33 m bridge (target bridge) features. Figure 23a clearly shows the data distribution shifts between the target bridge and the source bridge. Additionally, the clusters of the same damage state for the target bridge and the source bridge are spread out far apart (i.e., farther than those of different damage states) making them more difficult to classify. However, Fig. 23b shows that the clusters of the same damage state for the source bridge and the target bridge are much closer while the clusters for different damage states are much farther away and remain separable. Therefore, the features extracted by our model are bridge-invariant as well as damage-sensitive.

In summary, our method performs well for monitoring multiple bridges using a wide range of vehicle dynamic properties and speed without using the data labels for the new bridge. Our method achieves up to 98% accuracy in the detection task (mean of 83%), up to 99% accuracy in the localization task (mean of 88%), and up to 92% accuracy in the quantification task (mean of 76%). Our method provides a scalable approach for efficient and low-cost bridge health monitoring.

7 Discussion

The aspirational goal of drive-by monitoring is to perform structural health monitoring of all bridges at the network level. The authors believe that to reach this ambition, the future solution will consist of a combination of procedures and methods. This is mainly because of the big heterogeneity of structural types, configurations and materials that compose existing road bridge inventories. But this is also because of the huge variability in traffic, road, and operational conditions. Therefore, no single solution will be applicable to all bridges in a road network. With this in mind, future developments in drive-by technology need to improve and propose new strategies to solve different aspects of the problem. Future developments should focus on signal processing of vehicle responses, understanding and removing the influence of external environmental and operational influences, noise reduction, profile influence compensation strategies, identification of bridge damage features for different damage types, data-driven model architectures, among others.

The future studies in drive-by technology should pave the way for its adoption by road authorities. They need to get clear and validated procedures, indicating what information is required from the vehicles and how to process it to optimize structural condition assessments. Drive-by technology is deemed to give supporting tools to the decision-makers. However, to reach this point, additional developments in this technology are necessary.

As mentioned in the introduction, to further develop drive-by technology it is crucial to have extensive real vehicle measurements, which is currently not the case. However, it is difficult to find the economic support to gather this data, because of the extent and practical complications of such a monitoring campaign. In those few cases where relevant data has been collected, it is either scarce and bridge-specific, and generally not available to the rest of the research community. The potential funders of such extensive campaigns need to be convinced of the potential of drive-by methods. Therefore, the dataset presented here serves as a first step towards that direction. Even though, the dataset is based on numerical simulations, it covers a variety of structural lengths and damage conditions for several different monitoring scenarios, road conditions and vehicle properties. Furthermore, it provides a uniform arena to benchmark and compare performances of existing strategies and future developments in the field.

8 Conclusions

This document has presented, first, a publicly available dataset intended as a benchmark for drive-by monitoring methods, which subsequently has been used to evaluate the performance of four recently published data-driven damage detection approaches.

The dataset consists of numerically generated vehicle responses during bridge crossings. A wide variety of configurations and conditions have been modelled and systematically stored in files readily available at [18]. The variability of the dataset includes different bridge spans, damage locations, damage magnitudes, road profile irregularities, monitoring scenarios and vehicle properties. In total, the dataset includes over half a million files with vehicle responses and additional information for individual crossing events.

The remainder of the document introduced four different data-driven damage detection methods, which were then applied to subsets of the dataset. First, a bridge damage indicator based on the results from an artificial neural network model was tested on results for the 9 m span bridge traversed by quarter-cars under normal uncontrolled operating conditions with lower severity damage levels. The results showed a distinct increase in the index values with damage, for both damage locations. The proposed data-driven method, based on the outputs from a trained ANN and a simple damage indicator, allows the influence of vehicle speed or other environmental/operational factors to be learned. This enables damage-related changes to be isolated and clearly identified. However, it is noted that the increase in the damage indicator does not give any direct insight into, or quantification of, the nature of the damage such as the type, location, or magnitude. In addition, the algorithm relies on changes in bridge frequency due to damage, which may not be the most sensitive parameter. Further development of this approach could consider different damage-sensitive input features to enhance damage-detection capabilities.

The second method defined a damage index based on the prediction error from a deep autoencoder model. The idea was tested using the responses from 5-axle trucks on 9 m and 15 m long bridges. The results demonstrated that the damage severity is captured in variations of the indicator, albeit a reduction in performance when including irregularities in the road. This method demonstrates considerable promise as a practical tool for the early detection of damage, owing to its capacity to offset the inherent fluctuations that occur under operational conditions with minimal preprocessing of the signals. However, the main challenge with this method is the need for extensive datasets of synchronized signals for each bridge being monitored. This challenge requires further investigation, incorporating other methods discussed in this study.

The third study presented an unsupervised learning framework based on a topologically driven methodology, which included dimensionality reduction and clustering. It was tested on the shortest bridge (9 m) using quarter-car vehicle responses and showed a clear separation for all bridge states, for all road conditions in the dataset. This method is a topological approach to data analysis. It has the benefits of suggesting that data which is similar in nature should in principle be grouped closely to one another. However, it does lack the ability to suggest how far, in a geometric sense, the individual data points are far away from one another. That would need to be the purpose of a further, deeper analysis once the topological data analysis is performed.

Finally, the last method achieved damage diagnosis on multiple bridges via domain adaptation, taking the 39 m span as reference (source) bridge. The study proceeded to transfer the diagnosis procedure to the 33 m target bridge, using quarter-car vehicle responses. The results showed good accuracy in detecting, locating and quantifying bridge damage, under the studied conditions. This method successfully diagnosed multiple bridges through vehicle vibrations without requiring labelled data from all the bridges. However, this method assumes that vehicle properties such as type, weight, size, and suspension system are consistent throughout the data collection. Future investigations should extend the method to crowd sensing scenarios, where vibration data can be collected from multiple vehicles passing by.

Overall, the methods applied to the dataset in this document show that there is clear potential in data-driven methods. Promising damage detection performances have been reported here for different bridge spans, road conditions and vehicle models. Nevertheless, these results also highlight that further improvements are necessary. This document challenges the research community to use this dataset to test and improve drive-by monitoring methods.

Appendix A

Model properties in dataset generation

See Tables 3, 4, 5, 6.

Table 3 Beam models properties

Name	Span (m)	Mass per unit length (kg/m)	Second moment of area (m^4)	f_1 (Hz)	f_2 (Hz)
B09	9	16,875	0.1139	9.4256	37.7024
B15	15	28,125	0.5273	5.6553	22.6211
B21	21	16,530	0.8722	4.8405	19.3619
B27	27	19,372	1.7055	3.7824	15.1295
B33	33	20,952	2.9327	3.1926	12.7705
B39	39	23,352	4.5132	2.6860	10.7440

Table 4 V1 model properties

Parameter	Probability distribution	Min	Max	Mean	Standard deviation	Unit
Body mass (m_B)	Normal	5000	12,000	8000	500	kg
Axle mass (m_G)	Normal	700	1500	1100	100	kg
Suspension stiffness (k_S)	Normal	1.5 E6	2.5 E6	2 E6	0.25 E6	N/m
Suspension damping (c_S)	Normal	30 E3	50 E3	40 E3	5 E3	Ns/m
Tire stiffness (k_T)	Normal	2.5 E6	4.5 E6	3.5 E6	0.25 E6	N/m

Table 5 V2 model properties

Parameter	Probability distribution	Min	Max	Mean	Standard deviation	Unit
Body mass (m_{B1})	Normal	5000	16,000	10,500	500	kg
Axle mass (m_{Gj})	Normal	600	1200	900	100	kg
Body moment of inertia (I_{B1})	Normal	35,000	65,000	50,000	5000	kg m^2
Suspension stiffness (k_{Sj})	Normal	4 E6	8 E6	6 E6	0.5 E6	N/m
Suspension damping (c_{Sj})	Normal	0.5 E4	1.5 E4	1 E4	0.1 E4	Ns/m
Tyre stiffness (k_{Tk})	Normal	1.25 E6	2.25 E6	1.75 E6	0.125 E6	N/m
Distance (d_1)	Constant	–	–	2.5	–	M
Distance (d_2)	Constant	–	–	2.5	–	M

Table 6 V5 model properties

Parameter		Probability distribution	Min	Max	Mean	Standard deviation	Unit
Tractor	Body mass (m_{B1})	Normal	2800	3400	3100	50	kg
	Axle mass (m_{Gj})	Normal	500	1000	750	50	kg
	Body moment of inertia (I_{B1})	Normal	4000	6000	5000	200	kg m^2
	Suspension stiffness (k_{Sj})	Normal	4 E6	8 E6	6 E6	0.5 E6	N/m
	Tyre stiffness (k_{Tk})	Normal	1.25 E6	2.25 E6	1.75 E6	0.125 E6	N/m
	Suspension damping (c_{Sj})	Normal	0.5 E4	1.5E4	1 E4	0.1 E4	Ns/m
Trailer	Body mass (m_{B2})	Normal	15,000	25,000	20,000	500	kg
	Axle mass (m_{Gj})	Normal	800	1400	1100	100	kg
	Body moment of inertia (I_{B2})	Normal	100 E3	140 E3	120 E3	10 E3	kg m^2
	Suspension stiffness (k_{Sj})	Normal	5 E6	15 E6	10 E6	1 E6	N/m
	Tyre stiffness (k_{Tk})	Normal	2.5 E6	4.5 E6	3.5 E6	0.25 E6	N/m
	Suspension damping (c_{Sj})	Normal	1 E4	3E4	2 E4	0.2 E4	Ns/m
Distances	b_1	Constant	–	–	3	–	m
	a_2	Constant	–	–	5	–	m
	d_1	Constant	–	–	1	–	m
	d_2	Constant	–	–	3.5	–	m
	d_3	Constant	–	–	2	–	m
	d_4	Constant	–	–	3.2	–	M
	d_5	Constant	–	–	4.4	–	m

Appendix B

Data and reproducibility

The dataset NuBe-DBBM (Numerical Benchmark for Drive-By Bridge Monitoring methods) described in Sect. 2 is publicly available in [18]. In addition, Table 7 provides the links to the repositories to reproduce the work of some of the results sections.

Table 7 Repository to reproduce results

Section	Link to repository
Section 3	Scripts available upon request
Section 4	https://github.com/muhmmadzs/DRB-NTNU
Section 5	https://github.com/pche123/VBI_UMAP_HDBSCAN
Section 6	https://github.com/Aggarwaljatin20/BeVBI

Acknowledgements The UNSW authors would like to thank Australian Research Council (ARC) for the provision of support under Discovery Early Career Researcher Award (DECRA) scheme with Grant Number DE210101625. The UCD authors wish to express their gratitude for the financial support received from the Irish Research Council (IRC) GOIPD/2023/1588 and the financial support of Science Foundation Ireland under Grant Number 20/FFP-P/8706.

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital). Parts of this study was funded by Australian Research Council (ARC) (Grant Number DE210101625). Also, parts of this study was funded by Irish Research Council (IRC) GOIPD/2023/1588 and the financial support of Science Foundation Ireland (Grant Number 20/FFP-P/8706).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Gkoumas K, Gkoktsi K, Bono F, Galassi MC, Tirelli D. The way forward for indirect structural health monitoring (iSHM) using connected and automated vehicles in Europe. Infrastructures. 2021;6:43. https://doi.org/10.3390/infrastructures6030043.
2. The International Transport Forum (2023) Preparing infrastructure for automated vehicles. Report, ITF Research Report, OECD Publishing, Paris
3. Singh P, Mittal S, Sadhu A. Recent advancements and future trends in indirect bridge health monitoring. Pract Period Struct Des Constr. 2023;28:03122008. https://doi.org/10.1061/PPSCFX.SCENG-1259.
4. Malekjafarian A, Corbally R, Gong W. A review of mobile sensing of bridges using moving vehicles: progress to date, challenges and future trends. Structures. 2022;44:1466–89. https://doi.org/10.1016/j.istruc.2022.08.075.
5. Wang ZL, Yang JP, Shi K, Xu H, Qiu FQ, Yang YB. Recent advances in researches on vehicle scanning method for bridges. Int J Struct Stab Dyn. 2022. https://doi.org/10.1142/S0219455422300051.
6. Yang YB, Wang ZL, Shi K, Xu H, Wu YT. State-of-the-art of the vehicle-based methods for detecting the various properties of highway bridges and railway tracks. Int J Struct Stab Dyn. 2020;20:13. https://doi.org/10.1142/S0219455420410047.
7. Shokravi H, Shokravi H, Bakhary N, Heidarrezaei M, Koloor SSR, Petru M. Vehicle-assisted techniques for health monitoring of bridges. Sensors. 2020;20:3460. https://doi.org/10.3390/s20123460.
8. Yang YB, Yang JP. State-of-the-art review on modal identification and damage detection of bridges by moving test vehicles. Int J Struct Stab Dyn. 2018;18:1850025. https://doi.org/10.1142/S0219455418500256.
9. Mei Q, Gül M, Shirzad-Ghaleroudkhani N. Towards smart cities: crowdsensing-based monitoring of transportation infrastructure using in-traffic vehicles. J Civ Struct Heal Monit. 2020;10:653–65. https://doi.org/10.1007/s13349-020-00411-6.
10. US Department of Transportation (2020) Identifying real-world transportation applications using artificial intelligence (AI). Real-world AI scenarios in transportation for possible deployment. Report no. FHWA-JPO-20–810, US
11. International Transport Forum (2021) Data-driven transport infrastructure maintenance. Report, International Transport Forum Policy Papers, No. 95, OECD Publishing, Paris
12. National cooperative highway research program (2020) Strategic issues facing transportation, vol 7: preservation, maintenance, and renewal of highway infrastructure. Report, Transportation research board, US, Washington
13. World road association (2023) Use of big data for road condition monitoring. Report, PIARC, France, Paris
14. Corbally R, Malekjafarian A. A data-driven approach for drive-by damage detection in bridges considering the influence of temperature change. Eng Struct. 2022. https://doi.org/10.1016/j.engstruct.2021.113783.
15. Sarwar MZ, Cantero D. Deep autoencoder architecture for bridge damage assessment using responses from several vehicles. Eng Struct. 2021;246:113064. https://doi.org/10.1016/j.engstruct.2021.113064.
16. Cheema P, Alamdari MM, Chang K, Kim CW, Sugiyama M. A drive-by bridge inspection framework using non-parametric clusters over projected data manifolds. Mech Syst Signal Process. 2022;180:109401.
17. Liu J, Xu S, Berge M, Noh HY. HierMUD: hierarchical multi-task unsupervised domain adaptation between bridges for drive-by damage diagnosis. Struct Health Monit. 2023;22:1941–68.
18. Cantero D. NuBe-DBBM: numerical benchmark for drive-by bridge monitoring methods. 2023. Zenodo Repos. https://doi.org/10.5281/zenodo.7741092.
19. Cantero D. VBI-2D—road vehicle-bridge interaction simulation tool and verification framework for Matlab. SofwareX. 2024;26:101725. https://doi.org/10.1016/j.softx.2024.101725.
20. Li Y (2006) Factors affecting the dynamic interaction of bridges and vehicle loads. Dissertation, University College Dublin
21. ISO 8608 (1995) Mechanical vibration—road surface profiles—reporting of measure data
22. Harris NK, Obrien EJ, González A. Reduction of bridge dynamic amplification through adjustment of vehicle suspension damping. J Sound Vib. 2007;302(3):471–85. https://doi.org/10.1016/j.jsv.2006.11.020.
23. Corbally R, Malekjafarian A. Examining changes in bridge frequency due to damage using the contact-point response of a passing vehicle. J Struct Integr Maint. 2021;6:148–58. https://doi.org/10.1080/24705314.2021.1906088.

24. Sarwar MZ, Cantero D. Vehicle assisted bridge damage assessment using probabilistic deep learning. Measurement. 2023;206:112216. https://doi.org/10.1016/j.measurement.2022.112216.
25. Percival DB, Walden AT. Wavelet methods for time series analysis. Cambridge: Cambridge University Press; 2000.
26. McInnes L, Healy J, Melville J (2018) Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint: 1802.03426. https://doi.org/10.48550/arXiv.1802.03426
27. McInnes L, Healy J, Astels S. Hdbscan: hierarchical density based clustering. J Open-Source Softw. 2017;2:205. https://doi.org/10.21105/joss.00205.
28. Liu RY, Singh K. Using iid bootstrap inference for general non-iid models. J Stat Plan Inference. 1995;43:67–75. https://doi.org/10.1016/0378-3758(94)00008-J.
29. Liu J, Bergés M, Bielak J, Garret JH, Kovacevic J, Noh HY (2018) A damage localization and quantification algorithm for indirect structural health monitoring of bridges using multi-task learning. In: 45th Annual review of progress in quantitative nondestructive evaluation, Vermont, USA, 15–19 July, paper no. 090003. https://doi.org/10.1063/1.5099821
30. Liu J, Chen B, Chen S, Berges M, Bielak J, Noh HY (2020) Damage-sensitive and domain-invariant feature extraction for vehicle-vibration-based bridge health monitoring. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain, 4–8 May, pp.3007–3011. https://doi.org/10.1109/ICASSP40776.2020.9053450
31. McKinley S, Levine M. Cubic spline interpolation. Eureka: College of the Redwoods; 1998.
32. Efron B. Bootstrap methods: another look at the jackknife. Ann Stat. 1979;7:1–26. https://doi.org/10.1214/aos/1176344552.
33. Zhu XQ, Law SS. Moving loads identification through regularization. J Eng Mech. 2002;128:989–1000.
34. Moore RC, deNero J (2011) L1 and L2 regularization for multiclass hinge loss models. In: Symposium on Machine Learning in Speech and Natural Language Processing
35. Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res. 2008;9:2579–605.