



# Model-free damage detection of a laboratory bridge using artificial neural networks

Aaron Ruffels<sup>1,2</sup> · Ignacio Gonzalez<sup>1,3</sup> · Raid Karoumi<sup>1</sup>

Received: 9 June 2019 / Revised: 14 December 2019 / Accepted: 27 December 2019 / Published online: 4 February 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

This paper investigates a model-free damage detection method using a laboratory model of a steel arch bridge with a five-metre span. The efficiency of the algorithm was studied for various damage cases. The structure was excited with a rolling mass and seven accelerometers were used to record its response. An artificial neural network (ANN) was trained to predict the bridge accelerations based on data collected from the undamaged structure. Damage-sensitive features were defined as the root mean squared errors between the measured data and the ANN predictions. A baseline healthy state was established with which new data could be compared to. Outliers from the reference state were taken as an indication of damage. Two outlier detection methods were used: Mahalanobis distance and the Kolmogorov–Smirnov test. The method showed promising results and damage was successfully detected for four out of the five single damage cases. The gradual damage case was also detected, however, for some instances, greater damage did not result in an increase in the damage index. The Kolmogorov–Smirnov test performed best at detecting small single damage cases, while Mahalanobis distance was better at tracking gradual damage.

**Keywords** Model-free · Damage detection · Structural health monitoring · Machine learning · Artificial neural network

## 1 Introduction

Bridges are a key component of a country's transportation infrastructure and studies have shown that a strong link exists between transportation infrastructure and economic development [1]. However, throughout the world bridges are ageing—for example in Europe nearly 66% of the existing railway bridges are more than 50 years old [2]. In addition to this, approximately 10% of all bridges in Europe are considered to be structurally deficient, while in the United States the percentage is even higher at 28% [3]. This poses a safety risk to the end users and may cause unnecessary delays in the transportation network. Therefore, to improve or maintain a high level of service, an effective bridge management system is required. Ideally, these systems would help to improve the lifespan of bridges through proper maintenance, thus resulting in lower life cycle costs as well as increased safety to the end user.

Important to bridge management is the ability to monitor a structure's health. Structural health monitoring (SHM) is defined as the implementation of a damage identification strategy to engineering structures [3]. The development of techniques and algorithms to detect damage is currently a very active field of research in structural engineering [4].

The aim of this study was to use a small-scale, steel, truss bridge to expand upon the methods developed by Gonzalez and Karoumi [4], Neves et al. [5] and Chalouhi et al. [6]. These methods used artificial neural networks (ANN) to learn the baseline healthy state of a structure and then make predictions to determine whether new measured data came from the same state. ANNs are well suited to damage detection of bridges, as they can be trained using only healthy data. This is important because one cannot freely introduce damage to an operational bridge as would be required for fully supervised learning methods.

In this study, the model bridge was excited using a rolling mass and accelerations were recorded at seven nodes for each crossing of the mass. The scope of the study was limited to a single axle released from a constant drop height. The accelerations as well as the velocity and temperature measurements were provided as inputs to an ANN. The ANN was trained from data recorded in the undamaged state. New unseen data,

✉ Aaron Ruffels  
ruffels@kth.se

<sup>1</sup> KTH Royal Institute of Technology, Stockholm, Sweden

<sup>2</sup> ELU Konsult AB, Stockholm, Sweden

<sup>3</sup> Sweco AB, Stockholm, Sweden

also from the undamaged state, were then used to statistically characterise the healthy, baseline state of the structure. The characterisation was based on the root mean squared errors (RMSE) between the measured and predicted values of a time-delayed acceleration time series. Once the baseline state was established, unseen data from both the undamaged and damaged states were compared to this baseline state. Mahalanobis distance and Kolmogorov–Smirnov-based damage index were used to detect outliers from the baseline state. These outliers were taken as an indication of damage.

This method was applied to six different damage cases. The first five damage cases were studied to determine whether the algorithm was capable of detecting damage for the given structure and type of excitation. The sixth damage case involved gradually increasing the damage to the structure. Using this case, the aim was to determine whether the algorithm could track gradual changes in the structural state.

## 1.1 Literature review

Many different damage detection methods have been established, however, these can commonly be classified into two categories: model-based, and model-free damage detection [4]. Model-based methods are those that require an accurate finite element model of the target structure to assess the level of damage. Model-free methods do not require a finite element model but instead use techniques such as statistical regression, artificial intelligence or other signal processing procedures to make predictions from measured data.

Model-based and model-free methods are complementary approaches. Each method provides different insights into a structure's health. ASCE [7] highlights the strengths and weaknesses of model-based and model-free damaged detection for various contexts. According to Laory et al. [8], model-based methods can help support decisions related to the long-term structural management such as estimation of reserve capacity and repair. On the other hand, model-free methods are better suited to continuous monitoring of structures, because they only involve tracking changes in time series signals. Farrar and Worden [9] have written extensively on the application of machine learning and statistical pattern recognition techniques to SHM. Their book provides a summary of the field with applications ranging from civil infrastructure to telescopes and aircraft.

Some common model-free damage detection methods include: wavelet analysis, principal component analysis, extraction of vibration parameters and the use of artificial neural networks.

Wavelets are mathematical functions that decompose a signal into its individual frequency components. Moyo and Brownjohn [10] presented an application of wavelet analysis to identify events and changes in a bridge during its construction. Strain data from the Singapore–Malaysia Second

Link Bridge were used in the analysis. The study found that abrupt changes were easy to detect, however, the signal required de-noising to minimise the detection of false events.

Lee et al. [11] proposed a continuous relative wavelet entropy-based reference-free damage detection algorithm for truss bridge structures. The method did not require measurements from a reference 'healthy' state. Comparisons instead were made by referring to damage signatures from other locations of the structure. Both a finite element model and a laboratory structure were used to test the method. Damage cases involved loosening bolts at various nodes. The proposed method was able to identify the location of damage for single- and multi-damage scenarios as well as progressive damage. However, the method required a sensor to be at the location of damage and no particular differences were observed when all sensors were located at healthy joints.

Principal component analysis (PCA) is used to reduce the dimensionality of a data set. Values described by  $n$  dimensions are transformed to a new space with fewer dimensions described by linearly uncorrelated variables called principal components. Laory et al. [8] investigated a model-free data interpretation method which combines moving principal component analysis (MPCA) with four different regression analysis methods: robust regression analysis (RRA), multiple linear regression analysis, support vector regression and random forest. The methods were applied on three case studies. This involved two numerical studies on a railway truss bridge and a concrete frame, and a full-scale test on the Ricciolo Viaduct. A fixed-size window which moved along the measurement time series was used to extract the data and calculate its principal components. The correlations between the principal components were chosen as damage-sensitive features and analysed using the various regression functions to detect damage in the structure. The minimum detectable damage for a numerical model of a railway truss bridge was approximately a 2% reduction in axial stiffness.

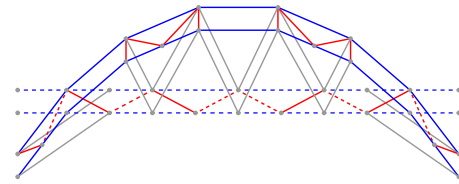
Moughty and Casas [12] investigated a number of damage-sensitive features extracted directly from the acceleration time series. These parameters included: RMS acceleration, cumulative absolute acceleration, Arias intensity, peak-to-peak acceleration and vibration intensity measured in Vibrars. Outlier detection methods such as Mahalanobis squared distance (MSD), PCA and singular value decomposition (SVD) were used to detect changes in the data between progressive damage tests. Data were taken from the S101 Bridge in Vienna, Austria. The results showed that vibration parameters associated with vibration energy—i.e. squared amplitude of vibration—performed best. These included the cumulative absolute acceleration, Arias intensity and the vibration intensity. Mahalanobis squared distance produced the highest indication of structural condition variation, while it was noted that the other methods also performed well.

Wu et al. [13] and Pandey and Barai [14] were some of the first authors to apply neural networks to the field of damage detection for civil engineering structures. Both studies used a supervised learning approach. The algorithms could correctly predict damage extent and location. Wu et al. [13] used a frequency response spectrum as the input to train an ANN with a single hidden layer. Three output nodes were used, each corresponding to the percentage reduction of stiffness in a different member. On the other hand, Pandey and Barai [14] trained a neural network with the vertical displacement at each node. The output nodes then predicted the cross-sectional area of each member in a 21-bar truss structure. Two configurations were used with one and two hidden layers, respectively.

Niu [15] and Niu and Qun [16] proposed a damage detection method using a time delay neural network. Niu used the same 21-bar truss structure as Pandey and Barai [14]. A comparison was made between a traditional neural network as studied by Pandey and Barai [14] and a neural network which was given both the original and a delayed signal as input. The time-delayed neural network took longer to train because it contained more input nodes, however, the performance was better at identifying damage.

Gonzalez and Karoumi [4] developed a new method using a two-stage machine learning algorithm trained using vibration data (deck accelerations) and bridge weigh-in-motion (BWIM) data (load magnitude and position). The first stage of the algorithm used an artificial neural network to predict future accelerations given a number of previous accelerations and the BWIM data. Thereafter, a Gaussian process was used to classify the prediction errors from the ANN to determine the probability of damage. A simply supported Euler–Bernoulli beam discretised into 30 finite elements was studied. A single damage case was investigated by applying a 30% stiffness reduction to one of the elements. This method showed very promising results. By setting the probability of correctly labelling damage at 90%, the false positive rate could then be calculated for different cases—the best case giving a 6% probability of false positives.

Neves et al. [5] and Chalouhi et al. [6] applied the method developed by Gonzalez and Karoumi [4] to two different cases. Neves et al. [5] used a three-dimensional finite element model of a single-track railway bridge. The structure consisted of a concrete deck of constant thickness, two steel girder beams that support the deck and steel cross-bracing that connect the girders. Chalouhi et al. [6] applied the method to the San Michele Bridge in Northern Italy (constructed in 1889). The effect of temperature variations was also taken into account by including this information as an input to the ANN. Both studies showed promising results. Using a Gaussian process, the prediction errors of the ANN were classified according to train velocity. The reliability of the detection method was shown to increase with the number of tested train passages.



**Fig. 1** Different cross-sections used for the bridge. Solid blue: arch, dashed blue: beam, gray: truss, dashed red: bracing A, solid red: bracing B

**Table 1** Member cross-section dimensions

Category	Section	Dim. (mm)
Arch	Rectangular hollow	30 × 25 × 2
Beam	Rectangular hollow	25 × 20 × 2
Truss	Circular hollow	17.2 × 2
Bracing A	Circular solid	6
Bracing B	Circular solid	8

## 2 Methods

### 2.1 The bridge structure

The laboratory model used in this study was a steel arch truss bridge with a 5 m span, 1 m width and a height of 1.741 m. The model was constructed from S235JR steel and all members were connected using M8 bolts.

This model was considered relevant for the current study due to its relative complexity. The structure included members which were subjected to various stress states such as bending, tension and compression. Steel arch trusses are a common structural system used for bridges. These were very popular in the late nineteenth to mid-twentieth centuries. However, some more recent examples include the New River Gorge Bridge completed in 1977 and the Chaotianmen Bridge completed in 2009 [17].

The bridge consisted of five different cross-section types as highlighted in Fig. 1. This included the arch, truss, main beam and two types of bracing—giving a total of 62 individual members. The dimensions of the cross-sections are given in Table 1.

### 2.2 Experimental setup

To implement the damage detection method, the following aspects were considered important: (1) the bridge should be excited in a realistic manner, (2) the sensors should be placed in such a way as to collect as much information as possible and (3) operational and environmental measures should be monitored such as temperature and velocity. Figures 2 and 3 show an overview of the experimental setup.

### 2.2.1 Method of excitation

Model-free damage detection is best suited to continuous structural monitoring as discussed by [7, 8]. Forced excitation methods were not considered as these would usually require a bridge to be closed to traffic while the investigation is taking place. Thus, a suitable method of excitation should mimic that of the traffic or ambient loading.

A rolling mass was therefore chosen as the method of excitation. This loading case is more analogous to railway traffic than automobile or pedestrian traffic, because only a single vehicle is on the bridge at a time. If used on a real bridge, the ANN could be trained for a specific loading that is frequent—for example a passage of a certain commuter or freight train on the bridge. Given the weight of the mass was 12.742 kg, a scaling factor was chosen so that the applied load corresponded to the weight of a typical single 20.4 ton axle load, giving a length scaling factor of  $\lambda_l = 40$ . The dimensions of the mass are shown in Fig. 4.

Realistic vehicles do not consist of a single axle. However, past studies which have used a single rolling mass to study the dynamic response of beams have obtained satisfactory results (Bilello et al. [18] and Stancioiu et al. [19]). Therefore, the single rolling mass assumption was considered acceptable.

### 2.3 Damage detection method

The proposed damage detection method made use of an artificial neural network (ANN) to predict the accelerations at various sensor locations. The prediction errors, which were defined as the difference between the predicted and measured accelerations, were used as damage-sensitive features. Two statistical outlier detection methods were then considered to distinguish between the damaged and undamaged data. Figure 5 shows a flow chart of how the damage detection method was implemented.

#### 2.3.1 Data acquisition and sensing

The variables given as an input to the damage detection algorithm included: bridge acceleration, vehicle velocity

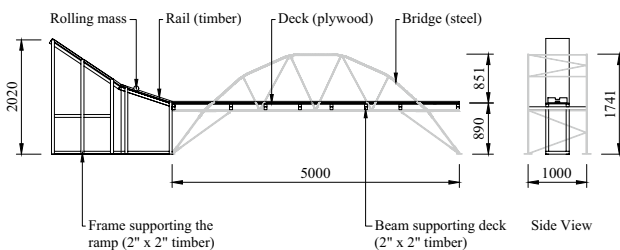


Fig. 2 Experimental setup



(a) Overview of the bridge set up.



(b) Rolling mass on the bridge.

Fig. 3 Bridge setup in the laboratory

and surrounding air temperature. Table 2 gives an overview of the measurement equipment used. All data used in this study were stored in an online data repository [20].

**2.3.1.1 Sensor placement** Some studies such as Niu and Qun [16] only considered the bottom chord nodes for measuring the structural response, while others—such as Laory et al. [8]—included sensors along the top chord, bottom chord and vertical trusses. To ensure that as much information as possible was obtained from the chosen

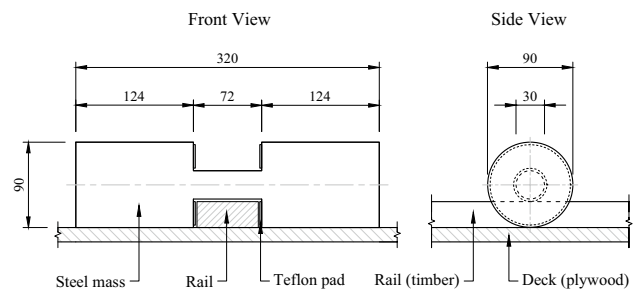
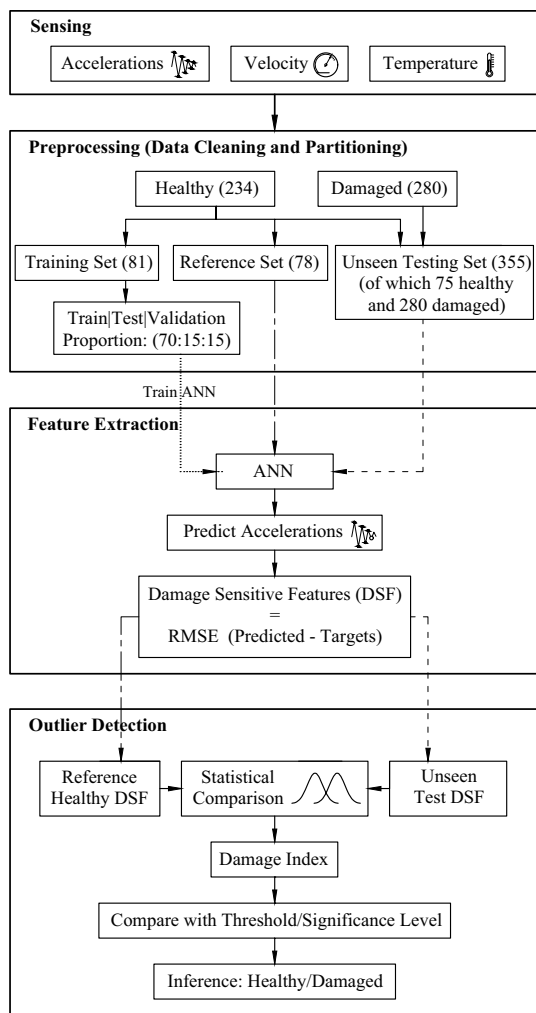


Fig. 4 Dimensions (in mm) of the rolling mass used for excitation

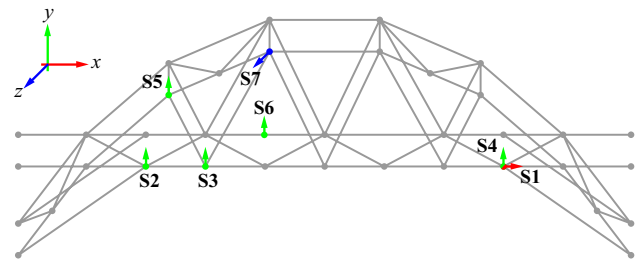


**Fig. 5** Flow chart of the damage detection process followed, from data collection to determination of damage index

**Table 2** Measurement equipment used

Equipment	Unit	Mfr.	Model
Accelerometers	m/s <sup>2</sup>	Colibrys	SF1500S.A
Signal amplifier	–	HBM	Spider8
DAQ software	–	HBM	Catman 4.6
Thermometer	deg C	Fluke	52 K/J

number of sensors, an optimisation algorithm was used. The algorithm described by Sun and Büyüköztürk [21] was chosen due to its relative simplicity as well having a number of clearly described case studies where the algorithm was applied with successful results. The chosen sensor layout was shown in Fig. 6 and the measurement directions for each sensor was summarized in Table 3.



**Fig. 6** Chosen accelerometer layout

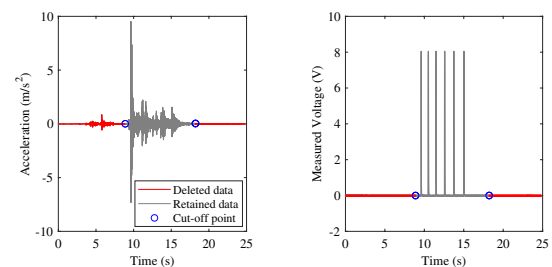
**Table 3** Sensor measurement directions

Measurement direction	Accelerometer
x (longitudinal)	S1
y (vertical)	S2, S3, S4, S5, S6
z (lateral)	S7

**2.3.1.2 Data measurements** The analysis of acceleration time signals was a central part of the damage detection method. Accelerations were measured using seven SF1500S.A/SF1500SN.A, uni-axial accelerometers connected to a Spider8 signal amplifier. The accelerometers were placed as described in Fig. 6. The sampling frequency was set at 400 Hz and no filters were applied to the signal. Figure 7a shows an example of the the acceleration time signal for a single run. Note that the mass only enters the bridge after approximately 9 s. The portion of the recorded signal retained for analysis is shown between the two cutoff points. This included measurements taken while the mass was on the bridge as well as approximately 3 s of free vibrations.

The velocity was calculated from the voltage peaks as shown in Fig. 7b. These were obtained from the steel mass crossing pairs of copper strips placed at 1 m intervals.

The data were divided into a number of sets for training and testing as shown in Fig. 5. The data from the healthy



**(a)** Signal from accelerometer 4. **(b)** Voltage signal to calculate velocity.

**Fig. 7** Portion of acceleration and voltage signals retained for analysis

state were divided randomly into three equally sized groups. One for training the ANN, another to use as a reference state for statistical comparison and the third was included in the unseen test set. All the data from the damaged state were also added to the unseen test set.

The data used to train the ANN were further subdivided into training, testing and validation sets in the following proportions 70:15:15. The testing and validation sets help to avoid overfitting the data when training the ANN as some of the data was kept unseen. The ANN with the best performance on the test set was retained.

### 2.3.2 Feature extraction from ANN predictions

Artificial neural networks (ANNs) have been used extensively in past damage detection studies [4–6, 14, 15, 22]. They are said to be universal approximators which can uniformly approximate any continuous function on a compact input domain to arbitrary accuracy provided the network has a sufficiently large number of hidden units [23, p. 230]. Thus, ANNs are suitable for the proposed damage detection method which attempts to predict acceleration signals.

In this study, an ANN was trained to curve fit acceleration time series in the undamaged state for a single crossing of the moving mass. To do this, the healthy data were arranged appropriately and given as an input to the ANN. Corresponding to each input is a target output—which also comes from the healthy data. The ANN was provided with many input–target pairs to learn the structure’s undamaged response as best as possible.

The data were arranged so that  $n$  accelerations (from  $m$  sensors) before acceleration  $p$ , and  $n$  accelerations (from  $m$  sensors) after acceleration  $p$  were given as an input to the ANN. The target was then to predict acceleration  $p$  for a particular sensor.

Expression 1 and 2 show an example of how the inputs and targets were arranged to predict the mid-accelerations. In this example, the number of sensors ( $m$ ) is 7. The number of accelerations before and after acceleration  $p$  was set to  $n = 3$ . The acceleration at time  $p$  was set as the target. Starting from  $p = n + 1$ , each acceleration in the acceleration time series is set as the target, while the surrounding  $n$  accelerations before and after  $p$  are given as inputs. Here  $v$  and  $T$  represent the velocity and temperature for each particular run.

In this study, seven accelerometers were used so  $m = 7$ , as in the example. However, the number of delays was set to 10 so  $n = 5$ . A separate ANN was trained to predict the accelerations at each sensor. The same inputs were given to train each ANN, however, the targets varied as shown in Expression 2. A graphic example of the inputs and predicted target values is illustrated in Fig. 8a. This method of arranging the inputs and targets for training the ANN was

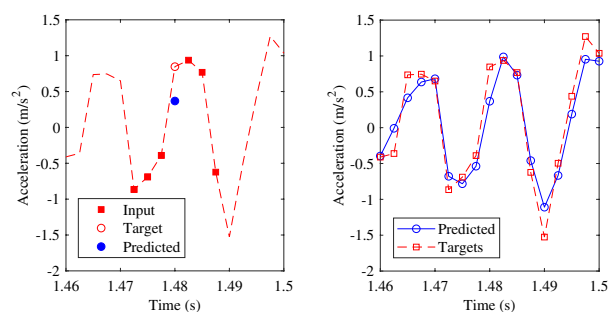
found to give better performance than the prediction of next accelerations for most cases.

$$\text{Inputs} = \begin{bmatrix} a_{\text{acc}1,t=1} & a_{\text{acc}1,t=2} & \dots \\ a_{\text{acc}1,t=2} & a_{\text{acc}1,t=3} & \\ a_{\text{acc}1,t=3} & a_{\text{acc}1,t=4} & \\ a_{\text{acc}1,t=5} & a_{\text{acc}1,t=6} & \\ a_{\text{acc}1,t=6} & a_{\text{acc}1,t=7} & \\ a_{\text{acc}1,t=7} & a_{\text{acc}1,t=8} & \\ a_{\text{acc}2,t=1} & a_{\text{acc}2,t=2} & \\ a_{\text{acc}2,t=2} & a_{\text{acc}2,t=3} & \\ a_{\text{acc}2,t=3} & a_{\text{acc}2,t=4} & \\ a_{\text{acc}2,t=5} & a_{\text{acc}2,t=6} & \\ \vdots & \vdots & \\ a_{\text{acc}7,t=7} & a_{\text{acc}7,t=7} & \\ v & v & \\ T & T & \dots \end{bmatrix} \tag{1}$$

$$\begin{aligned} \text{Targets}_{\text{ANN1}} &= \{a_{\text{acc}1,t=4} \ a_{\text{acc}1,t=5} \ \dots\} \\ \text{Targets}_{\text{ANN2}} &= \{a_{\text{acc}2,t=4} \ a_{\text{acc}2,t=5} \ \dots\} \\ &\vdots \\ \text{Targets}_{\text{ANN7}} &= \{a_{\text{acc}7,t=4} \ a_{\text{acc}7,t=5} \ \dots\} \end{aligned} \tag{2}$$

**2.3.2.1 Evaluation of ANN performance** Using this method, a good fit could be obtained between the predicted accelerations and targets as shown in Fig. 8b. Nevertheless, the predictions were not perfect. Even in the healthy state, prediction errors existed and contained some variability from one run to the next.

The performance of the predictions made by the network was calculated by determining the total deviation between the predicted and target values for an entire run. The root mean square error (RMSE) between each output–target was summed for a particular run to calculate the total prediction error for that run as shown in Eq. (3).



(a) Illustration of inputs to predict a single acceleration. (b) Comparison of predicted and target acceleration-time series.

Fig. 8 Predicted and target acceleration time series

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^T (\text{output}_i - \text{target}_i)^2}{N}} \quad (3)$$

where for each instant  $i$ ,  $\text{target}_i$  is the measured acceleration,  $\text{output}_i$  is the acceleration predicted by the ANN, and  $N$  is the total number of samples for a particular run. Compared to the simple mean absolute error, the RMSE magnifies and severely penalizes large errors [5] making this a more suitable choice of error function.

Once the network was trained, it was presented with new unseen data. This could consist either of healthy or damaged data. The ANN makes predictions assuming the structure will respond according to the relationships learned from the undamaged training data. The RMSEs between the predicted and target accelerations are therefore expected to increase when the structure is damaged. Thus, the RMSEs were deemed to be suitable damage-sensitive features.

Finally, to detect damage, the magnitude and/or the distribution of the damage-sensitive features (i.e. RMSEs) could be compared. Outliers from the normal condition were flagged as damage as described in the following section.

### 2.3.3 Outlier detection

Mahalanobis distance was used to calculate a damage index (DI) for each run. This measure has been widely used in past damage detection studies [24–27], as it is fairly simple and allows multivariate data to be reduced to a single damage index. A 95% threshold was applied over the healthy data. Anything above the threshold was considered to be damage. A requirement for using Mahalanobis distance is that the data should be normally distributed—or Gaussian. To ensure this, both Lilliefors and Anderson–Darling tests were used.

Mahalanobis distance was good at giving a damage index for each individual run. However, every now and then, healthy data were flagged as an outlier but thereafter returned to below the threshold.

Damage is expected to cause a permanent change to the structure and one would expect more than one point to indicate abnormal behaviour if damage was present. Therefore, a statistical test comparing a moving window of ten points from the unknown data with a reference healthy distribution was used to detect changes to the structure which presented consistent indications of damage. The two-sample Kolmogorov–Smirnov test was therefore chosen as the second DI, as it does not require any assumptions about the distribution of the data [28], samples of different sizes can be compared and smaller sample sizes are acceptable [29]. Furthermore, the test can detect differences in both the means as well as the variances between the two samples [30] while the  $t$  test is only capable of detecting differences in the means [31]. When using the Kolmogorov–Smirnov-based damage index,

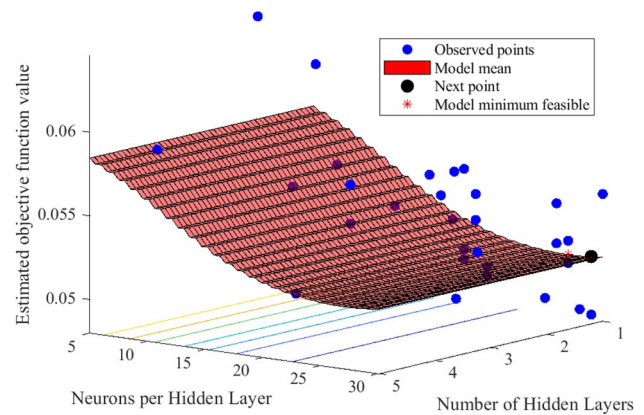


Fig. 9 Objective function model for the hyperparameter search

Table 4 Hyperparameters considered for optimisation

Hyperparameter	Range of inputs considered
Transfer function	Log-sigmoid (logsig) Tan-sigmoid (tansig) ReLU (poslin)
Training function	traingdx(*) trainbr (†) trainlm (‡)
Network depth	1–6 hidden layers
Network width	5–40 neurons per hidden layer
Number of delays	1–14 delays in the input time signal

(\*) Gradient descent backpropagation, (†) Bayesian regularisation backpropagation, (‡) Levenberg–Marquardt backpropagation

the damage threshold was defined as the 0.05 significance level.

All tests were performed from a constant drop height resulting in a small variation in velocity. Therefore, a Gaussian process was not required to characterise the damage-sensitive features according to the velocity.

### 2.3.4 Hyperparameter search

A hyperparameter search was conducted to determine a suitable network architecture for the ANN. Matlab provides a Bayesian optimisation approach, `bayesopt`, for this problem which is in line with recommendations from the literature [32]. `bayesopt` uses the ARD Matérn 5/2 kernel function. The hyperparameters considered as well as the constraints of possible inputs are shown in Table 4.

The validation performance of the network was used as an objective function for the optimisation. Here the validation performance refers to minimising the error defined in Eq. (3) over the validation set (i.e. 15% of the training set as shown in Fig. 5). Figure 9 shows the objective function

**Table 5** Chosen ANN configurations (see Table 4 for abbreviation definitions)

Hidden layers	Neurons/layer	Delays	Transfer function	Training function
1	40	10	poslin	trainbr

model obtained from optimising the number of hidden layers as well as the number of neurons per layer for the ANN to predict mid-accelerations. Smaller values indicate better performance. From this figure, one can see that the performance increases rapidly up to approximately 25 neurons per layer. There is also no significant increase in performance from using more layers for the given problem. The chosen parameters are indicated in Table 5.

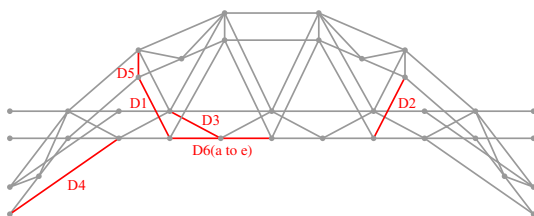
### 3 Damage cases investigated

Six damage cases were investigated. These were chosen to investigate various phenomena of interest. Table 6 gives a brief description of each damage case, while the corresponding damage locations are shown in Fig. 10. For some cases, the selected members were influenced by the sensor locations. The reader is therefore referred to Fig. 6 for a diagram of the accelerometer locations.

**Table 6** Description of damage cases investigated

Case	Member	Description
D1	Truss	Member removed
D2	Truss	Member removed
D3	Bracing—deck	Member removed
D4	Support	Member removed
D5	Bracing—arch	Member removed
D6a	Beam	5 mm deep cut
D6b	Beam	10 mm deep cut
D6c	Beam	15 mm deep cut
D6d	Beam	15 mm deep cut + widening(§)
D6e	Beam	15 mm deep cut + widening(§)

(§) see Fig. 12

**Fig. 10** Members corresponding to the damage cases in Table 6

D1 and D2 were selected to study the effect of damage in one of the main trusses. D1 was chosen at a location with a high density of sensors, while D2 was chosen at a location with a lower sensor density. This configuration was chosen to investigate the effect of sensor density on the detection capability of the algorithm. When investigating damaged joints, Lee et al. [11] found that a particular difference could not be detected if all sensors were placed at healthy joints using a continuous relative wavelet entropy method. The effect of sensor density on detection capability was therefore investigated for the proposed algorithm.

D4 was chosen to investigate damage to one of the supporting members. This member was located at a densely instrumented portion of the structure and thus damage was expected to be easily detected at this location. Some initial cases which were easy to detect were included to help validate and improve the algorithm before considering more complex and subtle damage scenarios.

D3 and D5 were selected to study the damage detection capability of the algorithm for braces in the transverse direction. Many previous studies such as Pandey and Barai [14], Niu [15], Laory et al. [8], Niu and Qun [16] and Lee et al. [11] only investigated two-dimensional truss structures, while Neves et al. [5] investigated damage to the bracing of a concrete–steel composite railway bridge. It was therefore deemed of interest to include members in the transverse direction for this study.

D6(a to e) was selected to investigate damage in the main beam. Previous studies such as Laory et al. [8] have often focused on the bottom chords of truss structures. Neves et al. [5] recommended that the smallest detectable damage should be investigated to understand the limitations of the method proposed by Gonzalez and Karoumi [4]. This damage case was therefore divided into five different subcategories with different degrees of damage. This was chosen to simulate gradual damage as well as to help identify the smallest detectable damage.

## 4 Damage detection

Damage cases 1–5 involved the removal of one member from the structure. The aim was therefore to identify a deviation in the structural behaviour from the reference healthy state. The results were presented for both Mahalanobis distance and Kolmogorov–Smirnov damage index using an ANN trained to predict mid-accelerations.

### 4.1 Effect of sensor density on damage detection ability

Damage cases 1 and 2 were used to investigate the effect of sensor density on the ability of the algorithm to detect



**Fig. 11** Damage detection results. For the Mahalanobis distance (MD) damage index, the blue +’s represent the reference set, the blue x’s represent healthy data in the unseen set, while the red o’s represent damaged runs in the unseen set. The horizontal dash-dotted line indicates the 95th percentile of the reference set. For the Kolmogorov–Smirnov (K–S) damage index, the dashed blue line represents a lower estimate of the index’s performance, while the solid blue line represents an upper estimate. The two estimates differ depending on the sequence the data are presented in. The dash-dotted line shows the 0.05 significance level

damage. Both cases were identical from a structural point of view. However, damage case 1 was located in a densely instrumented portion of the structure with sensors placed on either end of the member. In contrast, damage case 2 had far fewer sensors in its vicinity.

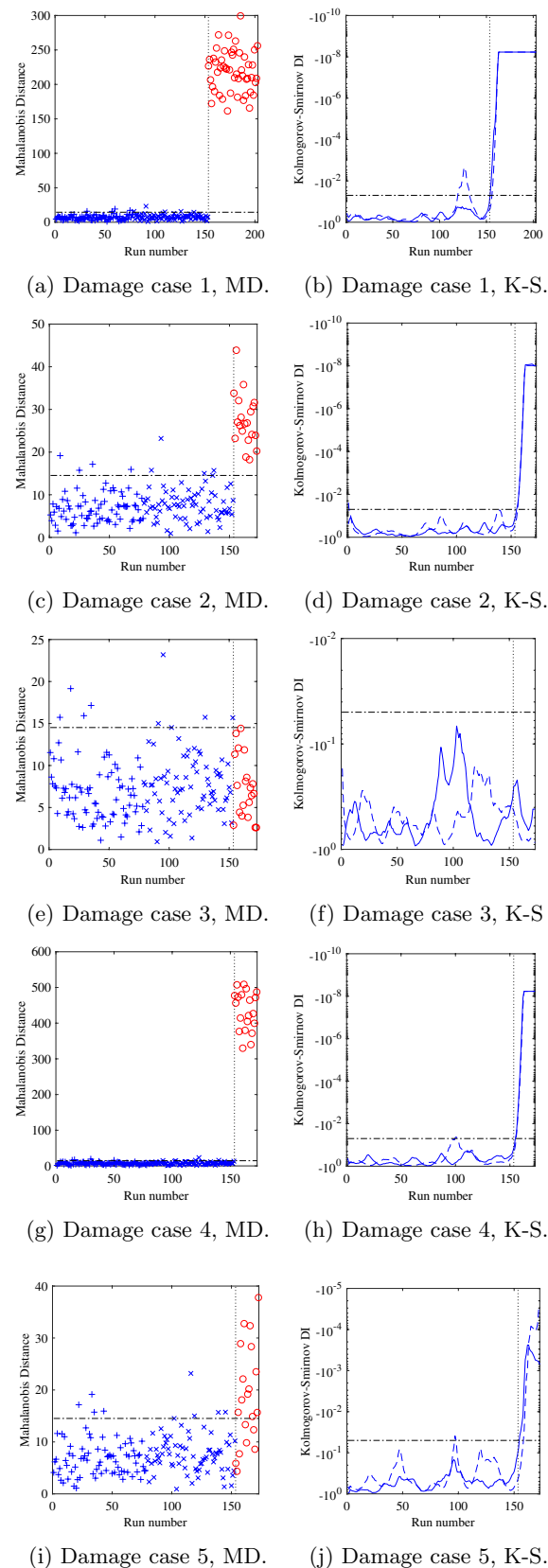
The damage indices are presented graphically in Fig. 11, while the confusion matrices for each method are shown in Table 7.

Both cases were successfully identified as damage. However, the Mahalanobis distance DI indicated a much higher level of damage for damage case 1 as can be seen when comparing Fig. 11a, c. This shows that sensor density does impact the damage detection ability. Another observation is that the damage index is not proportional to the magnitude of damage but instead indicates the degree of discordance between the two data sets.

Some other damage detection methods have also encountered difficulties when trying to detect damage away from the measurement locations. For example, Lee et al. [11] found that if all sensors were located at a healthy joint, damage at neighbouring joints could not be detected.

The confusion matrices in Table 7 show that the Mahalanobis distance DI is more susceptible to type I errors (false positive detection), where healthy data are detected as damage. This is because a cutoff value was set at the 95th percentile of the reference healthy data. The Kolmogorov–Smirnov test is more robust to these errors, as this index relates to the likelihood of a ten-window set of runs belonging to the same set as the reference healthy data.

For damage case 2, the Kolmogorov–Smirnov test gave a type II error (false negative detection). These errors are typically seen as more serious than type I errors as a damaged run was detected as healthy. The run that was assigned as healthy was the first of the damaged runs to be presented to the algorithm. Because this damage index considers a 10-point window, a single damaged run may not raise the alarm. However, after each successive damaged run is added to the window the index indicates damage with a greater and greater certainty.



**Table 7** Confusion matrices for damage cases 1–5 (note that only upper limit of the K–S DI was presented here)

		Predicted	
		D	H
Actual	D	50	0
	H	9	144

(a) Damage case 1, MD.

		Predicted	
		D	H
Actual	D	50	0
	H	0	153

(b) Damage case 1, K-S.

		Predicted	
		D	H
Actual	D	20	0
	H	9	144

(c) Damage case 2, MD.

		Predicted	
		D	H
Actual	D	19	1
	H	0	153

(d) Damage case 2, K-S.

		Predicted	
		D	H
Actual	D	0	20
	H	9	144

(e) Damage case 3, MD.

		Predicted	
		D	H
Actual	D	0	20
	H	0	153

(f) Damage case 3, K-S.

		Predicted	
		D	H
Actual	D	20	0
	H	9	144

(g) Damage case 4, MD.

		Predicted	
		D	H
Actual	D	19	1
	H	0	153

(h) Damage case 4, K-S.

		Predicted	
		D	H
Actual	D	13	7
	H	9	144

(i) Damage case 5, MD.

		Predicted	
		D	H
Actual	D	19	1
	H	0	153

(j) Damage case 5, K-S.

## 4.2 Effect of the damage location on detection ability

Damage cases 3, 4 and 5 were used to investigate how the algorithm performed for different damage locations. Graphs for each damage case are shown in Fig. 11, while the confusion matrices are given in Table 7.

Damage case 3 could not be detected by either method. Almost all damaged runs lie below the 95% threshold for the Mahalanobis distance DIs in Fig. 11e and the Kolmogorov–Smirnov DI does not cross the 0.05 significance level in Fig. 11f. The member removed for damage case 3 was located between the two main beams. One explanation for the poor damage detection performance was that the rolling mass crossing the bridge (in its centre) did not provide adequate excitation in the lateral direction for a difference to be observed between the healthy and damaged states.

Damage case 4 was clearly detected with all damage lying above the 95% threshold in Fig. 11g. There was only one damage run which was detected as healthy in the Kolmogorov–Smirnov confusion matrix for this case. However, as discussed in the previous section, the first damaged run is often not identified by the Kolmogorov–Smirnov-based DI. This is because more evidence is needed to ensure that there has been a significant change between the sample and reference distributions.

The member removed for damage case 5 was a lateral brace, similar to that in damage case 3. While damage case 3 was not detected, there was a clear distinction between the healthy and damaged runs for this case. The main difference was that the brace in damage case 5 was located on the arch, above the deck. This portion of the structure was much more sensitive to vibrations. In fact, the first two mode shapes of the structure related to the lateral vibration of the arch. Damage case 5 was therefore more straightforward to detect, as it was located in a portion of the structure more sensitive to vibrations. Thus, these results indicate that for damage to be detected the effect of the damage should influence an important global mode of vibration.

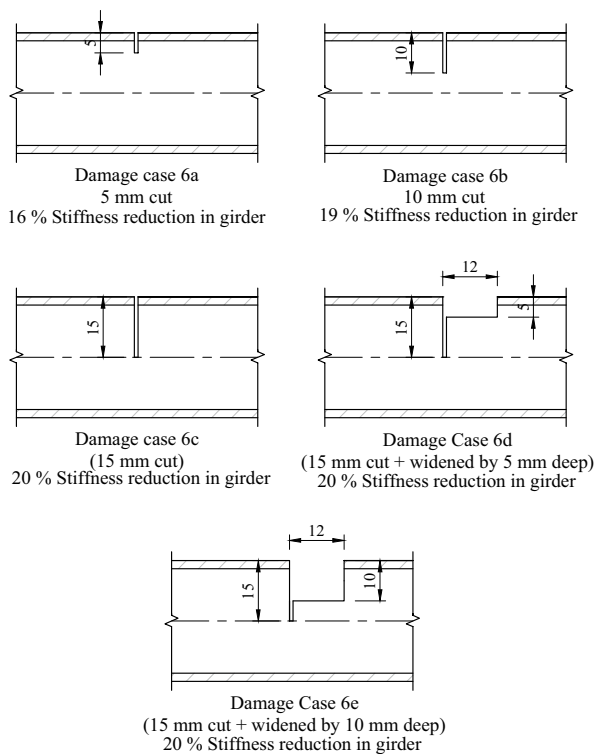
The Kolmogorov–Smirnov-based DI outperformed the Mahalanobis distance DI in damage case 5 as can be seen when comparing the confusion matrices between Table 7i and j. The subtle change in the data did not result in all the damage runs lying above the 95% threshold in Fig. 11i. However, the Kolmogorov–Smirnov-based DI detects a significant change in the mean and is therefore able to detect the damage with more certainty. Figure 11j shows a sharp increase in the damage index as soon as the damage is introduced.

## 5 Gradual damage detection

Damage case 6 was used to investigate the algorithm's ability to detect gradual damage. This damage case was divided into five different levels, each with progressively more damage as shown in Fig. 12.

The results for the gradual damage are shown in Fig. 13a for the Mahalanobis distance DI. The data from the first 5 mm cut (case 6a) were not detected, and only 4 out of 29 points crossed the 95% threshold. The Kolmogorov–Smirnov DI in Fig. 13b only crossed the 0.05 significance level after the moving window reached the last 10 runs from damage case 6a. While the cut was only 5 mm deep, it involved the removal of the entire top flange of one of the main beams and thus resulted in a 16% decrease in the moment of inertia of the girder. It should be noted that the girder was made up of two steel beams as well as the timber deck.

After the 10 mm cut, there was a sharp increase in both damage indices, while the 15 mm deep cut only caused a



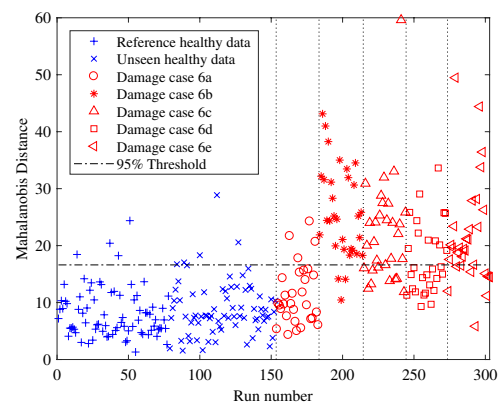
**Fig. 12** Gradual damage accumulation for damage case 6

sight increase in the Kolmogorov–Smirnov DI. The widening of the cut did not cause any significant increase as can be seen in damage case 6d and instead a decrease in the damage index was observed. This could be because the tests for cases 6a to 6c were carried out on a different day to 6d and 6e.

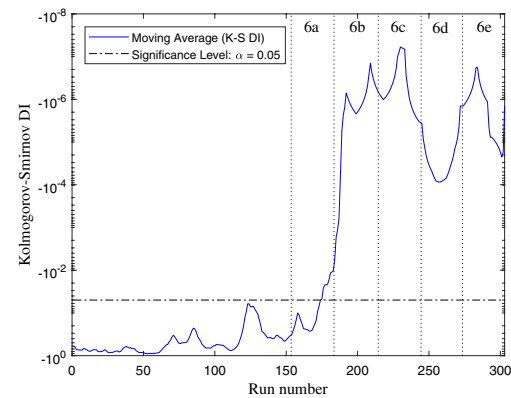
The Kolmogorov–Smirnov-based DI determines the likelihood of the test sample coming from the same distribution as the reference healthy sample. The index increases rapidly until damage is detected. Depending on the size of the samples being compared, there is a limit to the index. After the damage is determined with the highest certainty possible for the given sample size, the DI remains constant. This was observed for the single damage cases in Fig. 11b, d and h in Sect. 4. Thus, the Kolmogorov–Smirnov test is good at detecting subtle changes to the damage index such as a consistent change of the mean. However, on the other hand, the Mahalanobis distance DI would be a better indicator of gradual damage as the distance continues to increase as damaged data move further and further from the reference healthy data sample.

## 6 Conclusions and recommendations

The aim of this research was to apply the damage detection method studied by Gonzalez and Karoumi [4], Neves et al. [5] and Chalouhi et al. [6] to a laboratory model. This



**(a)** Mahalanobis distance DI



**(b)** Kolmogorov-Smirnov based DI

**Fig. 13** Results from the prediction of mid-accelerations

involved training an artificial neural network (ANN) on measured accelerations from the healthy structural state. A reference healthy state could then be established. By comparing new data with this reference healthy state, damage could be inferred. The objective was to apply this method to single and gradual damage detection.

The previous studies used a Gaussian process to classify the prediction errors with respect to vehicle speed. Since the drop height, and hence speed, of the rolling mass was kept constant this step was not required. Instead, two damage indices (DI) were investigated. This included Mahalanobis distance DI which has been previously been applied in a number of studies [24, 27, 33]. This method considered the distance between a measured result and the mean of a reference distribution. Another damage index was developed which made use of the Kolmogorov–Smirnov test. This test determines the likelihood that a test sample is from the same distribution as a reference sample. The benefit of comparing two samples is that the method is much less prone to type

I errors or false positive detections (where healthy data are classified as damaged).

## 6.1 Single damage detection

Five single damage cases were investigated. A significant distinction was made between the healthy and damaged data for four out of these five cases. The case which could not be detected involved the removal of a lateral brace between the two main beams. A possible reason for this case not being detected was that the member was located in a portion of the structure which was not excited by the rolling mass.

Greater sensor density improved the degree to which damage was detected. However, the algorithm was still able to detect the removal of a member even if no sensors were located at the end nodes of that member.

The Kolmogorov–Smirnov-based DI performed better than the Mahalanobis distance DI for cases with subtle differences between the healthy and damaged runs. This was because the method could detect changes in the distribution of the data compared with the reference healthy sample. Thus, a small but consistent change in the mean could be detected with a high degree of certainty.

When choosing the hyperparameters, a number of ANNs were trained. It was found that an improvement in the ANNs prediction performance over the healthy data generally resulted in an improvement in the damage detection capability. However, after a certain performance was achieved the gains in damage detection capability followed a law of diminishing returns, especially with respect to the greater computational time required.

## 6.2 Gradual damage detection

The gradual damage case investigated showed that it was possible to detect gradual changes. However, the damage index did not always progress as expected. For example, one of the cases where one would have expected an increase in the damage index, a decrease was instead observed. This was especially clear in Fig. 13b. Thus, further investigation into gradual damage detection is recommended. It was also concluded that Mahalanobis distance DI was better for tracking gradual damage as this DI would continue to increase as damage moved further and further from the reference state.

## 6.3 Recommendations

In the same way that weather forecasters do not rely on a single model to make predictions of the weather, engineers monitoring structural health should apply multiple damage detection algorithms across a range of damage-sensitive features. Human input could be used to interpret the results based on each model's strengths and weaknesses,

their knowledge of the structure as well as the prevailing environmental conditions. Confidence in the overall detection system could be improved when a number of models converge to the same prediction.

### 6.3.1 Applications for SHM on real structures

Based on the conclusions above the following comments could be made for applications of the method on real structures:

- Using an ANN to predict accelerations, damage can be detected especially if the damaged member is active in an important global mode of vibration.
- Damage can be detected by sensors away from the location of damage.
- A damage index comparing statistical significance (such as the Kolmogorov–Smirnov test) performs better at distinguishing subtle damage than a discordancy measure such as Mahalanobis distance.
- An improvement in the ANN's performance tends to improve the damage detection capability.
- A discordancy measure such as Mahalanobis distance is better at tracking gradual damage than a statistical significance test such as the Kolmogorov–Smirnov test.

### 6.3.2 Recommendations for future research

This study was conducted indoors with fairly constant temperatures. Furthermore, there were only small differences in the speeds of the mass crossing the bridge. Further research could build on the work by Chalouhi et al. [6] to study the impact of operational and environmental variations. Alternatively, the effect of excluding environmental variables such as temperature could also be investigated.

The first gradual damage case of a 5 mm cut was not identified. Therefore, improvements to detect more subtle damage could be investigated.

Real-world, continuous structural healthy monitoring systems generate vast amounts of data. The amount of computational power required to analyse this data using ANNs should be investigated. For example, the minimum number of training runs to accurately characterise the structure's behaviour under operational and environmental conditions could be evaluated. A more in-depth study on optimal hyperparameters could be conducted to assess training data from different structures.

Deep neural networks have made significant gains in image and speech recognition. There should therefore be an investigation into whether similar benefits can be obtained by applying deep neural networks to damage detection problems.

## References

1. Owen W (1959) Special problems facing underdeveloped countries. *Am Econ Rev* 49(2):179–187. <http://www.jstor.org/stable/1816113>
2. Bell B (2004) European railway bridge demography—deliverable D 1.2. Technical report, Sustainable Bridges Consortium
3. Wenzel H (2009) Health monitoring of bridges. Wiley, Vienna
4. Gonzalez I, Karoumi R (2015) BWIM aided damage detection in bridges using machine learning. *J Civ Struct Health Monit* 5(5):715–725. <https://doi.org/10.1007/s13349-015-0137-4>
5. Neves AC, González I, Leander J, Karoumi R (2017) Structural health monitoring of bridges: a model-free ANN-based approach to damage detection. *J Civ Struct Health Monit* 7(5):689–702. <https://doi.org/10.1007/s13349-017-0252-5>
6. Khouri CE, Ignacio G, Carmelo G, Raid K (2017) Damage detection in railway bridges using machine learning: application to a historic structure. *Proced Eng* 199:1931–1936. <https://doi.org/10.1016/j.proeng.2017.09.287>
7. ASCE (2013) Structural identification of constructed systems approaches, methods, and technologies for effective practice of St-Id. *Am Soc Civ Eng*. [https://doi.org/10.1061/41016\(314\)139](https://doi.org/10.1061/41016(314)139)
8. Laory I, Trinh TN, Posenato D, Smith IFC (2013) Model-free methodologies for data-interpretation during continuous monitoring of structures. *J Comput Civ Eng* 27(6):142. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000289](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000289)
9. Farrar Charles R, Keith W (2013) Structural health monitoring: a machine learning perspective, 1st edn. Wiley, Oxford. <https://doi.org/10.1002/9781118443118>
10. Moyo P, Brownjohn JMW (2001) Bridge health monitoring using wavelet analysis. In: Proceedings of SPIE. The International Society for Optical Engineering, 43 (June): 17. <https://doi.org/10.1117/12.429636>. URL <http://vibration.shef.ac.uk/doc/P00351.pdf>
11. Lee SG, Yun GJ, Shang S (2014) Reference-free damage detection for truss bridge structures by continuous relative wavelet entropy method. *Struct Health Monit* 13(3):307–320. <https://doi.org/10.1177/1475921714522845>
12. Moughy JJ, Casas JR (2017) Performance assessment of vibration parameters as damage indicators for bridge structures under ambient excitation. *Proced Eng* 199:1970–1975. <https://doi.org/10.1016/j.proeng.2017.09.306>
13. Wu X, Ghaboussi J, Garrett JH Jr (1992) Use of neural networks in detection of structural damage. *Comput Struct* 42(4):649–659. [https://doi.org/10.1016/0045-7949\(92\)90132-J](https://doi.org/10.1016/0045-7949(92)90132-J)
14. Pandey PC, Barai SV (1995) Multilayer perceptron in damage detection of bridge structures. *Comput Struct* 54(4):597–608. [https://doi.org/10.1016/0045-7949\(94\)00377-f](https://doi.org/10.1016/0045-7949(94)00377-f)
15. Niu L (2012) Monitoring of a frame structure model for damage identification using artificial neural networks. In: 2nd international conference on electronic and mechanical engineering and information technology, pp 438–441
16. Niu L, Qun C (2013) Structural health monitoring and damage detection using neural networks. In: Third international conference on intelligent system design and engineering applications, pp 1302–1304. <https://doi.org/10.1109/ISDEA.2012.307>. <http://ieeexplore.ieee.org/document/6455402/>
17. Gerard P, Nigel H (2008) ICE manual of bridge engineering, 2nd edn. Thomas Telford, London. <https://doi.org/10.1680/mobe.34525.0001>
18. Bilello C, Bergman LA, Kuchma D (2004) Experimental investigation of a small-scale bridge model under a moving mass. *J Struct Eng* 130(5):799–804. [https://doi.org/10.1061/\(ASCE\)0733-9445\(2004\)130:5\(799\)](https://doi.org/10.1061/(ASCE)0733-9445(2004)130:5(799))
19. Danut S, Simon J, Huajiang O, Mottershead John E (2009) Vibration of a continuous beam excited by a moving mass and experimental validation. *J Phys Conf Ser* 181:1. <https://doi.org/10.1088/1742-6596/181/1/012084>
20. Ruffels A (2019) Acceleration data from a laboratory bridge excited using a single rolling Axle. Mendeley Data, V1
21. Hao S, Oral B (2015) Optimal sensor placement in structural health monitoring using discrete optimization. *Smart Mater Struct* 24:12. <https://doi.org/10.1088/0964-1726/24/12/125034>
22. Fang X, Luo H, Tang J (2005) Structural damage detection using neural network with learning rate improvement. *Comput Struct* 83(25–26):2150–2161. <https://doi.org/10.1016/j.comps-truc.2005.02.029>
23. Bishop CM (2006) Pattern recognition and machine learning, 1st edn. Springer, Singapore
24. Figueiredo E, Park G, Farrar CR, Worden K, Figueiras J (2011) Machine learning algorithms for damage detection under operational and environmental variability. *Struct Health Monit* 10(6):559–572. <https://doi.org/10.1177/1475921710388971>
25. Khoa NLD, Zhang B, Wang Y, Chen Y, Mustapha S (2014) Robust dimensionality reduction and damage detection approaches in structural health monitoring. *Struct Health Monit* 13(4):406–417. <https://doi.org/10.1177/1475921714532989>
26. Santos A, Silva M, Santos R, Figueiredo E, Sales C, Costa JCWA (2016a) A global expectation-maximization based on memetic swarm optimization for structural damage detection. *Struct Health Monit* 15(5):610–625. <https://doi.org/10.1177/1475921716654433>
27. John JM, Joan RC (2018) Damage sensitivity evaluation of vibration parameters under ambient excitation, pp 1–3
28. Pratt JW, Gibbons JD (1981) Concepts of nonparametric theory. Springer, Berlin. <https://doi.org/10.1007/978-1-4612-5931-2>
29. Lilliefors HW (1967) On the Kolmogorov–Smirnov test for normality with mean and variance. *J Am Stat Assoc* 62(318):399–402
30. Magel RC, Wibowo SH (1997) Comparing the powers of the Wald–Wolfowitz and Kolmogorov–Smirnov tests. *Biomet J* 39(6):665–675
31. Kanji Gopal K (2006) 100 statistical tests, 3rd edn. SAGE Publications Ltd, Thousand Oaks. <https://doi.org/10.1177/09622802000900509>
32. Jasper S, Hugo L, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in neural information processing systems, vol 25. Curran Associates Inc, New York, pp 2951–2959. [arXiv:1206.2944S](https://arxiv.org/abs/1206.2944S)
33. Santos A, Figueiredo E, Silva MFM, Sales CS (2016b) Costa JCWA machine learning algorithms for damage detection: kernel-based approaches. *J Sound Vib* 363:584–599

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.