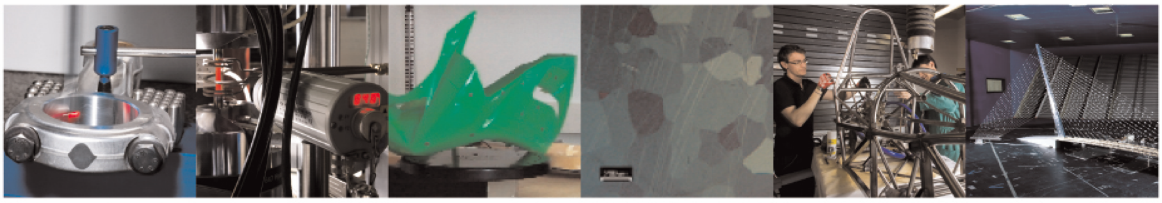
POLITECNICO
MILANO 1863

DIPARTIMENTO DI MECCANICA

mecc



Towards a deep learning-based unified approach for structural damage detection, localisation and quantification

Lomazzi L.;Giglio M.;Cadini F.

This is a post-peer-review, pre-copyedit version of an article published in Engineering Applications of Artificial Intelligence. The final authenticated version is available online at: https://doi.org/10.1016/j.engappai.2023.106003

© <2023>

This content is provided under CC BY-NC-ND 4.0 license

DIPARTIMENTO DI MECCANICA ■ POLITECNICO DI MILANO
via G. La Masa, 1 ■ 20156 Milano ■ EMAIL (PEC): pecmecc@cert.polimi.it
http://www.mecc.polimi.it
Rev. 0

Towards a deep learning-based unified approach for structural damage detection, localisation and quantification

Luca Lomazzi, Marco Giglio, Francesco Cadini*

Politecnico di Milano, Department of Mechanical Engineering, Via La Masa n.1, Milano, Italy

*Corresponding author: francesco.cadini@polimi.it

E-mail addresses: luca.lomazzi@polimi.it (Luca Lomazzi), marco.giglio@polimi.it (Marco Giglio), francesco.cadini@polimi.it (Francesco Cadini).

Abstract

Ultrasonic guided waves have been extensively employed for characterising structural damage thanks to their sensitivity to defects. Although they are easy to excite and acquire, heavy processing is often required to extract single-valued indicators of damage presence, or damage indices, from the acquired signals. Traditionally, damage indices have been elaborated through tomographic algorithms to generate damage probability maps, even though limitations affect the performance of such approach. Recently, the potentialities of machine learning have been leveraged to improve the accuracy of frameworks processing guided waves for damage diagnosis. However, most methods still require extracting damage indices from the acquired signals, which may bring to loss of diagnostic information and reduced accuracy. Furthermore, damage position and extent are usually roughly estimated through classification, while regression should be employed instead. In this context, this work aims (i) to test the capabilities of different supervised machine learning algorithms to localise and quantify damage through regression and (ii) to carry out a critical discussion about possible limitations of using damage indices instead of unprocessed signals. Results are compared to identify which algorithm performs better and if machine learning can improve the accuracy of damage diagnosis compared to traditional imaging methods. An experimentally validated numerical case study was used to test the capabilities of the proposed machine learning-based framework and to bring evidence of the accuracy of the algorithms involved to characterise damage with properties not seen during training.

Keywords: Damage detection; damage localization; damage quantification; machine learning; ultrasonic guided wave; SHM.

1 Introduction

Damage may affect structures during their life, eventually decreasing their safety level and increasing the failure probability of the system. Historically, preventive and/or corrective maintenance strategies were considered to allow engineering structures to operate in safe conditions. However, such strategies turned out to be economically inefficient, due to unnecessary shutdowns and replacements of components. Recently, more advanced maintenance techniques have been developed, such as those in the condition-based maintenance category. Those policies drive condition-based, or even predictive, maintenance actions according to diagnostic information retrieved from a network of sensors permanently installed on the structure. This damage diagnosis practice, which is commonly known as structural health monitoring (SHM), has been successfully applied in several engineering fields [1–8]. In particular, focusing on thin-walled structures, which are widely employed in engineering applications, satisfactory performance is provided by methods processing ultrasonic guided waves. Such waves are typically excited and sensed by installing a network of piezoelectric (PZT) devices on the structure [9–15]. Usually, damage diagnosis is performed in two steps. First, signals are acquired when the structure is not affected by damage. This set of signals is typically referred to as healthy baseline. Then, current measurements are compared to the healthy baseline to identify deviations indicating damage [16,17]. Most of the methods based on this strategy leverage tomographic reconstruction-based diagnostic algorithms to identify and localise damage [18,19]. One of the most successful approaches of this kind is the reconstruction algorithm for probabilistic inspection of damage (RAPID) [20]. However, features, or damage indices (DIs), must be extracted from the acquired signals to drive tomographic algorithms [21–30]. Although DIs provide easily interpretable single-valued indicators of the structural health state, they include less information than that carried by unprocessed signals. Besides this loss of information, tomographic reconstruction equations usually present some limitations: (i) artifacts may be introduced in damage probability maps due to uneven sensing network density [27,31], (ii) subjective parameters need to be set up, the variation of which strongly affects the damage diagnosis performance, and (iii) no information about damage quantification is easily retrievable. To solve the issues mentioned above, machine learning (ML) has been recently considered. To date, most of the contributions have proposed feed-forward neural networks (FFNNs) to identify and characterise damage. However, such simple networks cannot deal with complex and multi-dimensional signals, and DIs still need to be extracted in order to achieve good performance [32,33]. Moreover, so far, only feed-forward neural networks for classification have been used to localise damage, despite damage position is a continuous variable and would be better described through regression [34]. The performance of ML-based damage diagnosis frameworks has been further improved by leveraging the potentialities of deep learning. Among deep learning algorithms, convolutional neural networks (CNNs) have been extensively used to characterise damage, thanks to their capability of dealing with unprocessed multi-dimensional signals. However, damage position and extent have usually been characterised through classification networks [35–39]. Although using classification to localise and quantify damage makes the task easier, prediction accuracy is limited by the resolution used to discretise damage position and extent. This limitation has been overcome in Ref. [40], where a CNN for regression has been proposed to localise damage in Aluminium plates. Despite the satisfactory performance, the CNN used in this framework does not deal with unprocessed acquired signals, but it requires preliminarily extracting DIs. More complex algorithms, such has model assisted convolutional and recurrent neural networks, have been proposed in the literature to avoid extracting DIs and simultaneously deal with localisation and quantification through networks for regression [41]. Given the complexity of the task, the capabilities of the method have only been demonstrated through a 1D case study. That is, two PZT devices were installed on a waveguide, and damage position was described by the distance of the anomaly from one of the PZT devices. No information about the extension of the framework to 2D structures, such as plates, was reported in Ref. [41]. Moreover, the generalisation capabilities of the methods proposed so far have not been addressed. That is, the prediction accuracy of damage positions and extents not seen during training has not been discussed [37,41,42].

Hence, the research trend in the field of SHM using guided waves seems to be moving towards the implementation of ML models to improve the performance of damage diagnosis. This trend is mainly described by the adoption of models to overcome the limitations of traditional methods. That is, new models need to be developed to allow avoiding the extraction of DIs from acquired signals, and to perform damage localisation and quantification through regression. Moreover, such data-driven models should also be used to predict unseen damage characteristics, in order to test the damage diagnosis performance out of the training domain. In this context, this work has a two-fold aim. First, this work aims to test the capabilities of different supervised ML algorithms to localise and quantify damage through regression. Furthermore, results are compared (i) to identify whether ML methods provide better performance than traditional imaging methods and to find out which ML algorithm performs better, and (ii) to carry out a critical discussion about possible limitations of using DIs instead of unprocessed signals. Second, this work aims to present a unified ML-based framework using guided waves for damage diagnosis. In particular, classification is used to detect damage, and regression is employed to quantify and localise it.

This paper is organised as follows. The supervised ML methods considered are introduced in Section 2. A case study is carried out in Section 3 to test and compare the performance of those methods, and to verify their generalisation capabilities on damage with position and extent non seen during training. Moreover, in Section 3 the performance of the ML methods is compared with that of a traditional imaging approach. Finally, Section 4 draws out the conclusions from this work and presents possible further developments.

2 Machine learning approaches for damage diagnosis using guided waves

Traditional ML methods and deep learning approaches were considered in this work to perform damage diagnosis using guided waves. Specifically, regression trees, FFNNs and CNNs were set up to localise and quantify damage through regression. Support vector machines and Gaussian processes were also evaluated, but they were discarder because of the computational cost required for training. Moreover, a classification CNN was employed to detect damage and drive the subsequent regression step. No other methods were tested to perform this task, since damage detection is not the main focus of this work. In fact, identifying the presence of damage is already a consolidated process, and both traditional ML and non-ML techniques could have been used to perform this task instead.

Regression trees are decision trees used to perform regression. Differently from global models for regression, such as traditional linear regression, trees allow recursively partitioning the prediction space into independent smaller regions, each of which can be described through a simple regression model. This is particularly useful when non-linearities and interaction between independent variables are present. Terminal nodes, or leaves, of regression trees represent the identified independent regions, and a simple regression model is attached to each leaf. During training, tree branches are generated so that inputs corresponding to different regression regions are assigned to different leaves, and the corresponding output is then predicted according to simple regression models. Such regression models are tuned during training. The interested reader is referred to Ref. [43] for additional information. Similarly to regression trees, FFNNs are approximators for complex and generally non-linear functions $\mathbf{y}=f(\mathbf{x})$, where $\mathbf{x}$ is the vector of independent variables, or inputs, while $\mathbf{y}$ is the vector of dependent variables, or outputs. FFNNs consist of nodes collected in layers. Nodes in adjacent layers are fully connected. That is, each node in a layer is connected to all the nodes in the subsequent layer. The information received by the j-th node is aggregated and elaborated according to Equation 1 [44]:

$$z_j = g\left(\sum_{i=1}^{N_j}\left(w_{ij}\cdot z_i\right) + b_j\right) \tag{1}$$

where z_j is j-th node output, $g(\cdot)$ a generally non-linear activation function, N_j the number of nodes passing information to the j-th node, z_i the value of the i-th node in the set of N_j nodes, w_{ij} the weight of the connection between the i-th node and the j-th node, while b_j is a bias parameter. This message passing and elaboration process is repeated from the input layer through the output layer, where predictions are provided. During training, weights w_{ij} and biases b_j are tuned by minimising the error between expected and predicted outputs. The interested reader is referred to the vast literature about the topic for additional information. Finally, CNNs are deep learning architectures typically involved in classification or regression problems requiring heavy image processing. The architecture of CNNs is inspired by the visual perception mechanism of living creatures, which was first studied in macaque and spider monkeys [45], and it was first formalised in the works in Ref. [46,47], in which the authors developed the LeNet-5 artificial neural network. This type of architecture started growing exponentially thanks to the improvements in the image classification task shown in Ref. [48], that presented the successful AlexNet network. All the CNNs presented in the literature so far share the same basic components, i.e., convolutional, pooling and fully connected layers Ref. [49]. Convolutional layers aim to extract features from the input data through convolutional kernels, which allow computing different feature maps. Inside feature maps, the value of each neuron comes from mathematical operations performed on the neuron's receptive field, which represents a set of neighbouring neurons in the previous layer. Such mathematical operations involve, in sequence, the convolution with a convolutional kernel and the application of a non-linear activating function. Pooling layers are typically introduced between two consecutive convolutional layers and allow achieving the shift-invariance property. Max pooling is considered in this work, i.e., pooling layers extract the largest value in each patch of feature maps [50]. Fully connected layers have already been introduced above for FFNNs, and elaborate the information through Equation 1. In CNNs, these layers are stacked after the sequence of convolutional and pooling layers, with the aim of performing reasoning-like procedures [51]. The last layer of a CNN can be a classification layer or a regression layer. A regression layer is a normal fully connected layer with one node per desired output. Instead, a classification layer is a layer with one node per predictable class and usually involves the softmax operator, which allows attributing to each class the probability that the input to the network belongs to that class [52]. During training, CNN parameters, i.e., weights and biases, are optimised by minimising an appropriately defined loss function.

The ML methods described above were used to process ultrasonic guided waves. Such waves are elastic waves propagating in solid media, and interact with discontinuities generating reflection and refraction phenomena [53]. Specifically, Lamb Waves (LWs) were considered in this work. LWs are plane strain waves propagating in free plate-like structures described by long-range propagation and high-sensitivity to damage properties [54]. According to the classical theory of elasticity, particles displacement in a structure made of isotropic material is given by Equation 2 [55]:

$$\mu u_{i,jj}+(\lambda+\mu)u_{j,ji}+\rho f_i=\rho\ddot{u}_i \tag{2}$$

where μ is the shear modulus, u_i the displacement in the x_i direction, λ the Lamé constant, ρ the mass density and f_i the body force in the x_i direction. Note that Equation 2 is written according to the Einstein's summation convention, $\ddot{()}$ is a shorthand for $\frac{\partial^2()}{\partial t^2}$, and $(),_{ij}$ stands for $\frac{\partial^2()}{\partial x_i \partial x_j}$. Considering a free plate, i.e., a plate with traction-free upper and lower surfaces, under the hypothesis of plane strain the exact solution to Equation 2 can be computed, which describes LWs governing equations. According to the wave motion in the thickness direction, two vibration modes exist, i.e., symmetric (S) and anti-symmetric (A) modes, which are shown in Figure 1 and described by the Rayleigh-Lamb frequency relations (or dispersion equations) reported in Equation 3 ([14,56]).

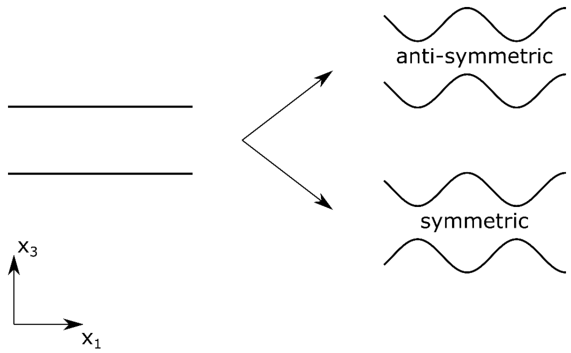


Figure 1. Symmetric and anti-symmetric Lamb modes.

$$\begin{aligned} &\text{S modes:} \quad \frac{\tan(qh)}{\tan(ph)}=-\frac{4k^2pq}{(q^2-k^2)^2} \\ &\text{A modes:} \quad \frac{\tan(qh)}{\tan(ph)}=-\frac{(q^2-k^2)^2}{4k^2pq} \end{aligned} \tag{3}$$

where h is the half-thickness of the plate, ω is the circular frequency and c_L, c_T and c_P represent the longitudinal, transverse and phase wave velocity, respectively. Moreover, p^2, q^2 and the angular wavenumber k are variables related to the circular frequency and to the wave velocities, as reported in Equation 3 ([14,56]).

$$p^2=\frac{\omega^2}{c_L^2}-k^2, \quad q^2=\frac{\omega^2}{c_T^2}-k^2, \quad k=\frac{\omega}{c_P} \tag{3}$$

The Rayleigh-Lamb frequency relations are commonly used to determine the velocity of LWs propagating in plate-like structures of thickness $2h$ at a specific wavenumber k. An infinite number of complex k values exists that satisfy these equations for any given frequency ω, but only a finite set of these values are physical and experimentally observable, i.e., the pure real angular wavenumber values, which represent undamped propagating LWs [53]. Hence, possible Lamb modes can be determined solving the Rayleigh-Lamb frequency relations for those modes related to real k values only, e.g., A0, S0, A1, S1, A2. Normally, the basic symmetric and anti-symmetric modes, i.e., S0 and A0, are employed to perform damage diagnosis. The former is sensitive to damage anywhere in the plate thickness and can travel longer distances than the latter, which is only sensitive to surface damage, but can detect smaller anomalies and normally shows wider transverse oscillations than the S0 mode [14]. As guided waves encounter damage, refraction and reflection phenomena occur, and mode conversion may take place. These phenomena introduce differences in the time and in the frequency domain between LWs sensed in a healthy structure and those acquired in presence of damage. In the interest of brevity, no additional information about LWs is reported here. The interested reader is referred to the work in Ref. [18] for further details.

The framework considered in this work to test the diagnostic performance of the ML methods presented above is shown in Figure 2. It is composed of three main steps, namely the Acquisition step, the Data preparation step and the Damage diagnosis step.

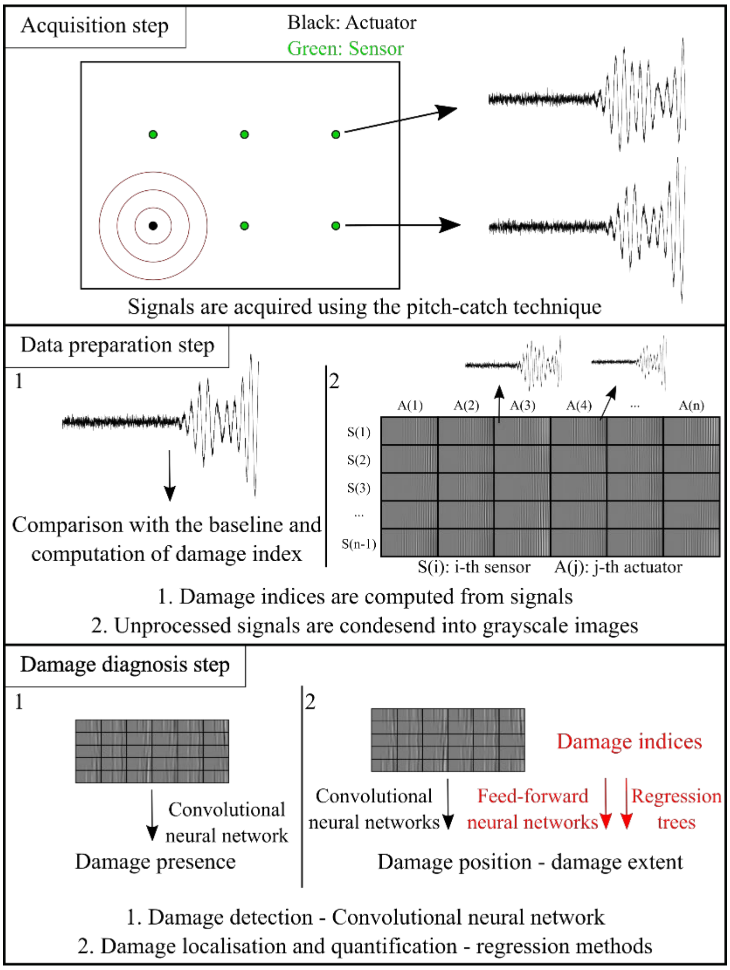


Figure 2. Diagram of the framework considered in this work.

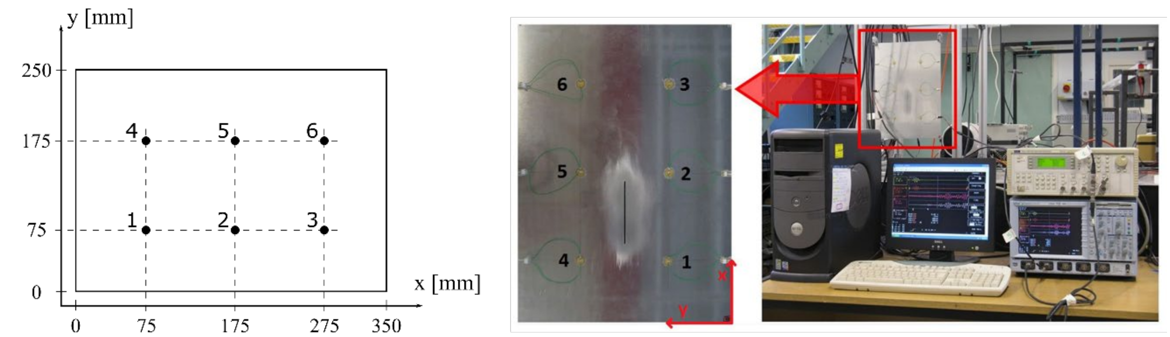
Within the Acquisition step, a network of PZT sensors permanently installed on the structure was employed to excite and sense LWs. Identifying with n the number of sensors, n acquisitions per analysis were performed, each of which consisting of the excitation from one PZT device, while the remaining *n-1* PZTs worked as sensors. Hence, $n \cdot (n-1)$ signals were gathered during each analysis. Indicating with f_s the sampling frequency and with t_a the time duration of the single acquisition, each signal is then composed of $n_s = f_s \cdot t_a$ samples.

Within the Data preparation step, two different databases were generated: (i) the database including DIs and (ii) that of grayscale images (GSIs). DIs were computed by comparing acquired signals with the healthy baseline. Specifically, a DI was assigned to each signal within each analysis, thus generating a database where each analysis was described through $n \cdot (n-1)$ single-valued DIs. Instead, GSIs were generated by collecting all the signals retrieved during the analysis, without any prior feature extraction process. Each GSI was represented through a matrix with $n-1$ rows, i.e., one row per sensor, and n columns, i.e., one column per actuator, where the element in position (i, j) was the signal gathered by actuating from the *j-th* PZT device and sensing from the *i-th* sensor and consisted of n_s values. GSIs were generated according to a global scale, i.e., acquired signals were mapped to GSIs so that maximum and minimum values among all the GSIs correspond to the maximum and minimum values among all the acquired signals. This was made to preserve wave amplitude relationships among acquisitions in different scenarios, so that possible amplitude modifications due to damage would be captured by the ML algorithms.

Within the Damage diagnosis step, a CNN for classification was used to detect damage. That is, a GSI was processed by the CNN to identify if damage was present or not in the corresponding analysis. In case damage was detected, different ML methods for regression were used to localise and quantify the anomaly. Specifically, regression trees and FFNNs were employed to process DIs, while CNNs were used to characterise damage through GSIs. Note that only damage localisation and quantification were performed, while the identification of damage type (e.g., delamination, hole, crack) was not subject matter of this work. Damage type classification could be classified through already consolidated methods, such as that presented in Ref. [57].

3 Case study

The framework presented in Section 2 was employed to test and compare the performance of different ML methods in diagnosing structural damage. The case study was carried out considering a 2mm thick rectangular aluminium plate with Young's modulus $E = 74\text{GPa}$, density $\rho = 2600\text{kg/m}^3$ and Poisson's ratio $\nu = 0.33$. The plate had size $350\text{mm}\ x\ 250\text{mm}$ in the xy plane, as shown in Figure 3. Six PZT SONOX-P5 transducers with 8mm diameter and 0.5mm thickness were installed on the plate to form a rectangular-shaped network of sensors. Damage was simulated by introducing a crack aligned with the x axis. The crack was introduced through saw cut and its length was gradually varied from 5mm to 70mm. A waveform generator TTi-TGA1242 and a DELL 2.8GHz computer were used to drive the PZT devices. Signals from PZT sensors were acquired through a LeCroy-LT264DSO oscilloscope with sampling frequency 500 Msamples/s [32]. Signals were acquired at each crack length using the pitch-catch technique. That is, one PZT at a time was driven through the waveform generator, while the remaining PZT devices were used to sense LWs. LWs were averaged over 15 consecutive excitations of the same actuator. No additional information about the experimental database are reported here, since no experimental test was conducted for this work. The interested reader is referred to Ref. [32] for a detailed description of the experimental setup and procedure. Experimental data was used in Ref. [32] to validate numerical simulations. In this work, the experimentally validated numerical simulations of A0 mode LWs presented in the work in Ref. [32] were reproduced through Abaqus. The plate was modelled using 4-node doubly curved general-purpose shell elements with 5 thickness integration points. Mesh size was set to 0.67mm. A linear elastic material model was employed, using the parameters reported above. The Abaqus dynamic explicit solver was used, with maximum time increment 0.1μs. The case study was kept fully numerical so to generate a coherent large database with enough samples to train the ML models. Furthermore, the experimental dataset was generated only considering one crack position, and it could not have been used to train regression methods to localise damage.

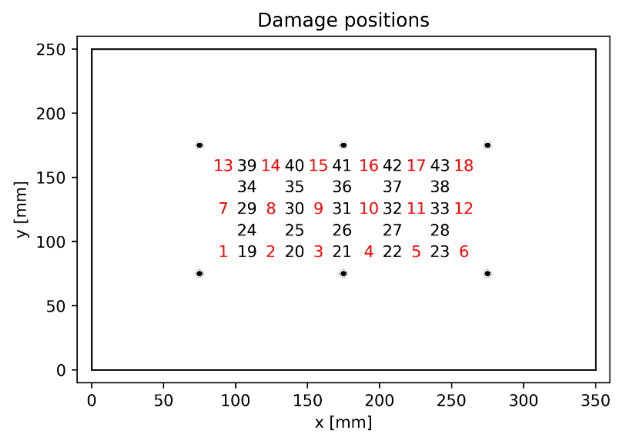


(a) (b)

Figure 3. Plate considered in the case study. (a) plate geometry and sensors – (b) experimental setup (reprint from Ref. [32]*).*

3.1 Numerical simulations

Numerical simulations with no damage were performed to acquire the healthy baseline. Instead, damage scenarios were simulated by introducing a crack at a time in the plate. Damage was modelled by assigning the properties of air to the crack. The 43 crack positions shown in Figure 4 were considered, along with 12 equally spaced crack lengths in the range [5mm, 60mm], with step 5mm. PZT actuators were driven with a 150kHz 5-cycle toneburst modulated with Hann window. Signals were sensed with a sampling frequency $f_s = 20\text{MHz}$ over a time window of $t_a = 150\mu\text{s}$. Further details about the database are reported in Table 1.



Position	x [mm]	y [mm]	Position	x [mm]	y [mm]
1	91.5	91.5	23	242.0	91.5
2	124.5	91.5	24	108.0	108.0
3	157.5	91.5	25	141.0	108.0
4	192.5	91.5	26	174.0	108.0
5	225.5	91.5	27	209.0	108.0
6	258.5	91.5	28	242.0	108.0
7	91.5	125.0	29	108.0	125.0
8	124.5	125.0	30	141.0	125.0
9	157.5	125.0	31	174.0	125.0
10	192.5	125.0	32	209.0	125.0
11	225.5	125.0	33	242.0	125.0
12	258.5	125.0	34	108.0	142.0
13	91.5	158.5	35	141.0	142.0
14	124.5	158.5	36	174.0	142.0
15	157.5	158.5	37	209.0	142.0
16	192.5	158.5	38	242.0	142.0
17	225.5	158.5	39	108.0	158.5
18	258.5	158.5	40	141.0	158.5
19	108.0	91.5	41	174.0	158.5
20	141.0	91.5	42	209.0	158.5
21	174.0	91.5	43	242.0	158.5
22	209.0	91.5			

Figure 4. Damage positions.

	Healthy baseline	Damage scenarios
Excited mode	A0	
Sensed signal	Out-of-plane displacement at sensors	
Number of analyses	1	516 (one per crack)
Number of crack lengths	~	12
Number of crack positions	~	43

Table 1. Details about the database.

3.2 Data preparation

Acquired signals were corrupted with noise to increase the dataset size and improve training performance. Specifically, 100 different replicas of each LW acquired with damage were generated by adding artificial noise. Noise was sampled from a Gaussian distribution $N(0, \sigma_N)$ with $\sigma_N = 2.1 \cdot 10^{-3}$ [normalised units], so that a signal-to-noise ratio (SNR) of 20db was achieved [38,58]. Following the same procedure, the healthy baseline was also corrupted with noise to produce 51,600 replicas of the LWs sampled without damage. This was made to have balanced sets to train the classification CNN with. Finally, signals were normalised with respect to the global maximum and minimum, i.e., considering the maximum and the minimum amplitudes over the whole database. Figure 5 compares a sampled signal with one of its noisy replicas.

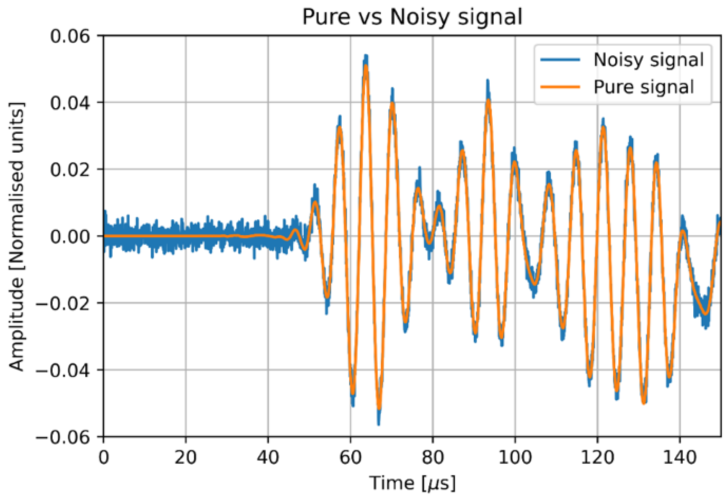


Figure 5. Comparison of sampled signal and one of its noisy replicas.

After introducing noise, three different datasets were generated from the acquired signals. That is, (i) signals acquired with damage were further processed to create a dataset of DIs, (ii) all unprocessed signals were collected in GSIs and GSIs were labelled to distinguish between damaged and undamaged scenarios, and (iii) unprocessed signals acquired in presence of damage were collected in GSIs, which were labelled according to the position and extent of damage. Specifically, DIs were computed using the Pearson correlation coefficient [17]. That is, within each analysis, each actuator-sensor pair was given a score based on Equation 4 [17]:

$$\mathrm{DI} = 1 - \frac{\mathrm{cov}(\mathrm{h}, \mathrm{d})}{\sigma_{\mathrm{h}} \sigma_{\mathrm{d}}} \tag{4}$$

where h and d indicate corresponding signals acquired with and without damage, respectively, cov(·) is the covariance operator and σ the standard deviation. The higher the DI, the more relevant the difference between the signal acquired without damage and that sensed in presence of the crack. Hence, since each analysis consisted of 30 signals, i.e., 5 acquired signals per actuation, each analysis was described by a feature vector of 30 DIs. Instead, GSIs were generated by following the procedure already described in Section 2. An example of GSI is shown in Figure 6, where a grid is superimposed on the image to identify the signals pertaining to each actuator-sensor pair. The signal corresponding to each actuator-sensor pair is composed of $n_s = 3000$ samples, which leads to an overall size of 18000 columns and 5 rows per GSI. In this study, during each acquisition sensors were numbered in clockwise order starting from the PZT actuator. This is shown in Figure 7, where actuators are highlighted in black, while sensors are represented by green circles

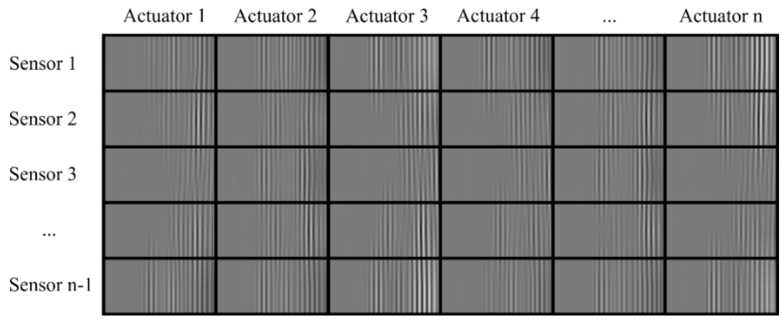


Figure 6. Example of GSI.

Figure 7. Sensors numbering in each acquisition. Actuators are coloured in black, sensors in green and LWs are represented through concentric red circles.

Datasets were split into three groups, i.e., training, validation and test sets, with a percentage of 70%, 28% and 2%, respectively. However, while training regression methods, crack positions number 2 and 12 (Figure 4) and cracks with length $l = 25$mm and $l = 45$mm were left out of the training and validation sets. This choice was made to allow testing the generalisation capabilities of the ML methods on never-seen-before damage positions and extents.

3.3 ML algorithms setup

Regression trees and FFNNs were optimised through the optimisation module of the regression learner matlab toolbox. Default optimisation parameters were adopted. That is, the architecture of regression trees and FFNNs was determined through Bayesian optimisation, using the expected improvement per second plus acquisition function and 30 iterations. The only regression trees hyperparameter selected for optimisation was minimum leaf size, which was searched in the range [1, max(2,floor(n_o/2))], where n_o is the number of observations. Instead, the number and dimension of fully connected layers, the activation function and the regularisation strength were optimised for FFNNs. Particularly, the number of fully connected layers was searched in the range [1, 3], their dimension in the range [1, 300], the activation function was selected among ReLU, tanh and sigmoid, while regularisation strength was searched in the range [$1e^{-5}/n_o$, $1e^{5}/n_o$]. FFNNs were trained using a limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton (LBFGS) algorithm [59] by minimising mean squared error (MSE). Crack centre x coordinate, crack centre y coordinate and crack length were independently characterised through different decision trees and FFNNs. The optimised architecture and training performance of regression trees and FFNNs are summarised in Table 2 and

Table 3, respectively. Note that crack centre x and y coordinates were standardised to make training easier, while crack length was not standardised. Instead, CNNs were set up using the deep learning API Keras, which is written in Python and runs on top of the ML platform TensorFlow [60,61]. The architecture of CNNs was selected through trial and error method, without recurring to automatic optimisation procedures. CNNs were trained resorting to the Adam optimiser. Default parameters were set to train regression CNNs, i.e., $learning\ rate = 1 \cdot 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 1 \cdot 10^{-7}$ [62], while non-default values of some parameters were required to improve convergence during the training of the classification CNN, i.e., $learning\ rate = 5 \cdot 10^{-4}$ and $\beta_1 = 0.95$ [63]. Regression CNNs were trained by minimising the MSE, while categorical crossentropy was considered for the classification CNN. Batch size of 50 was selected for training. According to the architecture described in Table 4, 588,138 parameters were optimised for the classification CNN, while regression CNNs required to train 587,881 parameters. Not that, as already mentioned above for regression trees and FFNNs, crack centre x coordinate, crack centre y coordinate and crack length were independently characterised by three distinct CNNs.

Parameter	Dependent variable		
	Crack centre x coordinate	Crack centre y coordinate	Crack length
Minimum leaf size	2	3	1
MSE (validation)	3.1426e-6 (standardised)	1.9527e-5 (standardised)	1.2946
Prediction speed	~1,100,000 obs/s	~1,200,000 obs/s	~1,300,000 obs/s
Optimisation time	15.70 s	12.56 s	11.20 s

Table 2. Architecture of regression trees.

Parameter	Dependent variable		
	Crack centre x coordinate	Crack centre y coordinate	Crack length
First layer size	23	100	52
Second layer size	179	275	299
Third layer size	~	~	~
Activation function	ReLU	ReLU	Sigmoid
Regularisation strength	2.5026e-10	2.8101e-10	1.1845e-9
MSE (validation)	7.9763e-6 (standardised)	3.2774e-5 (standardised)	1.2119
Prediction speed	~540,000 obs/s	~490,000 obs/s	~430,000 obs/s
Optimisation time	998.84 s	872.31 s	1272.90 s

Table 3. Architecture of FFNNs.

Layer	Classification CNN	Regression CNN
Input	Size: 5 x 18000 x 1	
Convolutional	Filters: 8 Kernel Size: 2 x 500	

	Stride: 1 x 100 Activation: ReLU	
Convolutional	Filters: 16 Kernel Size: 2 x 4 Stride: 1 x 2 Activation: ReLU	
Max pooling	Pool size: 2 x 2 Stride: 1 x 2	
Convolutional	Filters: 32 Kernel Size: 2 x 4 Stride: 1 x 1 Activation: ReLU	
Max pooling	Pool size: 1 x 2 Stride: 1 x 2	
Fully connected	Size: 640 Activation: Sigmoid	
Fully connected	Size: 256 Activation: Sigmoid	
Fully connected	Size: 2 Activation: softmax	Size: 1 Activation: linear

Table 4. Architecture of CNNs.

The accuracy and loss function evolution during the training of the classification CNN are shown in Figure 8. Ten epochs were enough to get satisfactory accuracy and loss function values. Instead, the training performance of regression CNNs are shown in Figure 9. Sixty epochs were required to achieve satisfactory training performance for all the three networks. Figure 8 and Figure 9 also bring evidence that no overfitting occurred during training. Note that output data standardisation was not required to train CNNs. Average training time for regression CNNs was 602.16 s, and average prediction speed was around 570,000,000 obs/s.

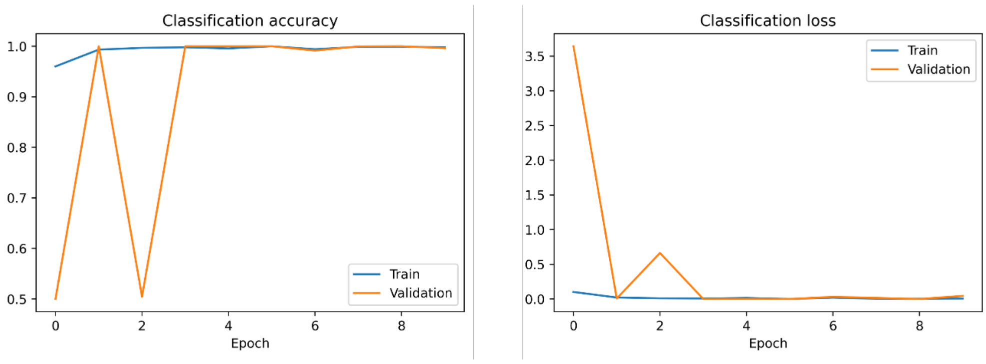


Figure 8. Accuracy and Categorical cross entropy loss function of the classification CNN.

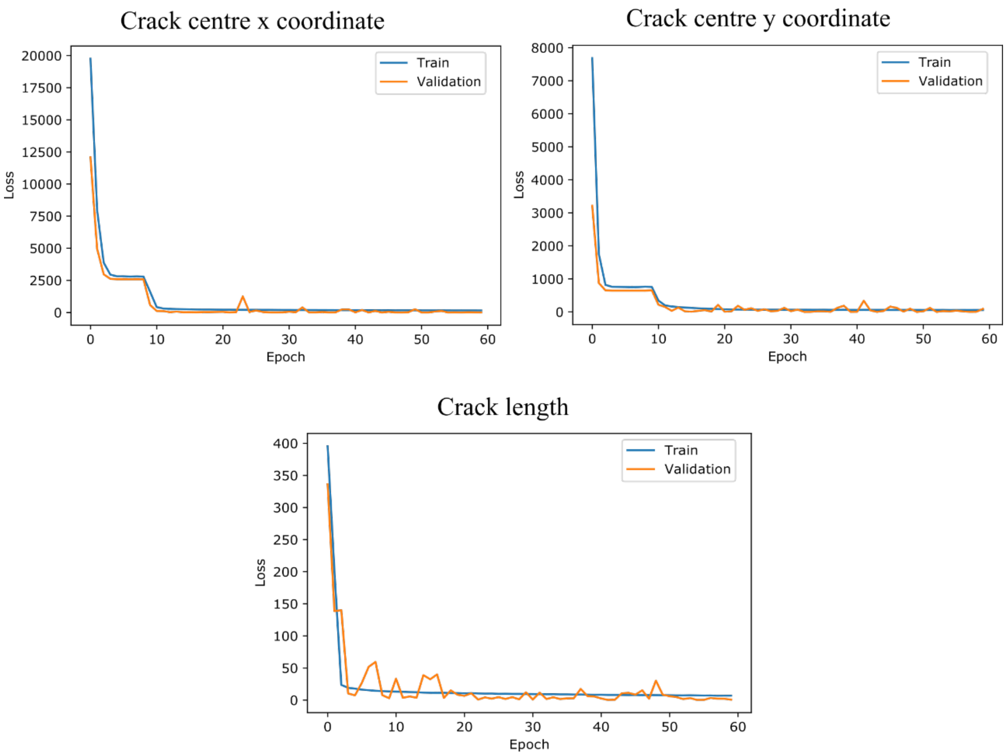


Figure 9. Mean square error loss functions of regression CNNs.

Finally, a sensitivity analysis was performed to identify the influence of the training database size on the performance of regression CNNs. Here, the number of GSIs describing each damage scenario was progressively reduced from 100 to 16, with step 14. Note that the performance of the regression ML methods described above were all achieved using 100 GSIs per scenario. This analysis was only conducted for CNNs in the interest of brevity. The variation of the final training loss (MSE) value with respect to the database size is shown in Figure 10 for all the regression CNNs. Results showed that the database size could be reduced of 70% without significantly decreasing the performance of the regression CNNs.

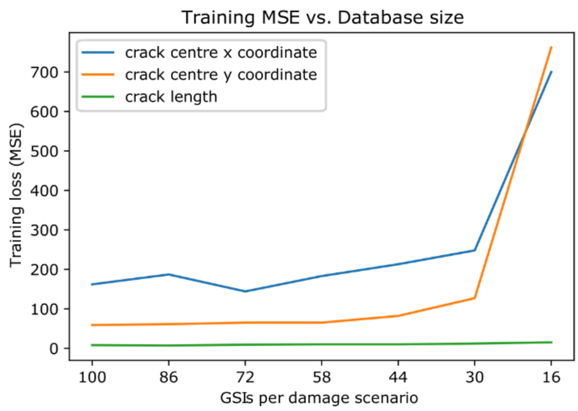


Figure 10. Training MSE vs. training database size for the regression CNNs.

3.4 Results

The performance of the ML methods considered in this work was evaluated by means of indicators computed over the test splits. Specifically, the performance of the classification CNN was described by classification accuracy, while boxplots were considered to evaluate the predictions of regression methods.

The classification CNN was able to detect damage with an accuracy of 99.6%. No additional details about damage detection through classification are reported here, since this is not a novelty of this work. The performance of regression methods is shown in Figure 11. Recall that crack positions number 2 ($x = 124.5$mm, $y = 91.5$mm) and 12 ($x = 258.5$mm, $y = 125.0$mm) and cracks with $l = 25$mm and $l = 45$mm were left out of the training and validation sets with the purpose of testing the generalisation capabilities of regression methods on never-seen-before damage positions and extents.

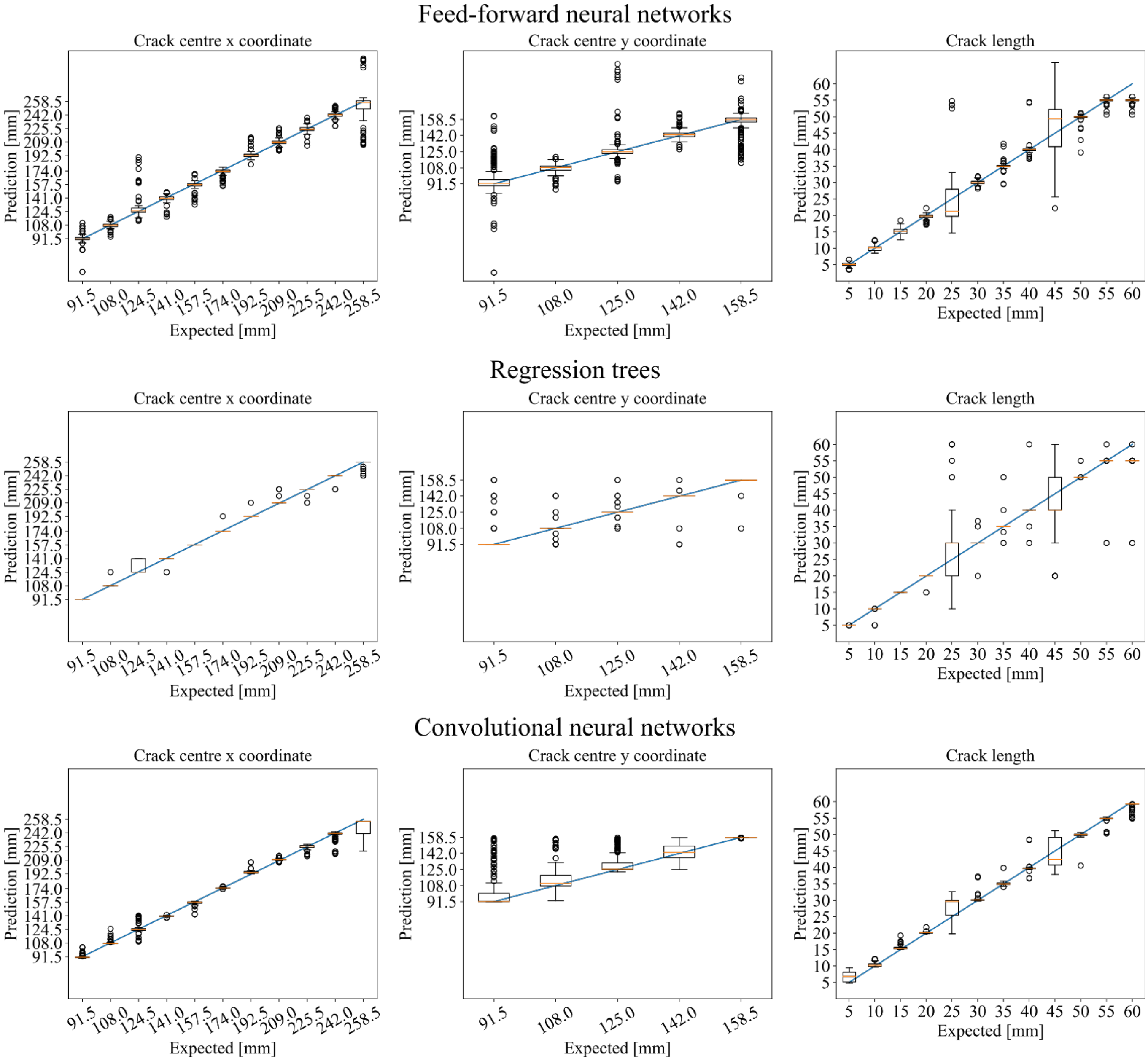


Figure 11. Prediction distributions of regression methods.

All the regression methods could satisfactorily predict the crack centre x coordinate. In fact, most of the predictions lay on the bisector line, while only few outliers (circles in the boxplots) were shifted towards the closest training positions. The regression tree provided slightly more accurate predictions, since boxplots were more localised to the expected values. As expected, predictions of cracks at $x = 124.5$mm and $x = 258.5$mm were characterised by higher dispersion around the expected value. This was mainly determined by the exclusion of positions number 2 ($x = 124.5$mm) and 12 ($x = 258.5$mm) from training splits, which reduced the number of training examples at those x coordinates. Similarly, predictions of the crack centre y coordinate were satisfactorily accurate. That is, most of the predictions lay on the bisector line, with a limited number of outliers. Instead, the regression methods performance in predicting crack length was not comparable. In fact, the regression

CNN performed well, and predictions were mainly localised to the expected lengths, while considerably higher dispersion characterised the predictions of the regression tree and of the FFNN. This was particularly evident for crack lengths $l = 25\text{mm}$ and $l = 45\text{mm}$, which were left out of training splits. This last consideration brought evidence that regression trees and FFNNs could not generalise on never-seen-before damage extents, while CNNs satisfactorily characterised them. This different is most likely related to the different input data processed by the methods. In fact, regression trees and FFNNs provide predictions by analysing DIs, while CNNs use GSIs to that purpose. To sum up, the information lost in condensing time signals into single-valued DIs could prevent ML methods from learning the relationship between LWs and damage extent, while the localisation performance was not influenced by pre-processing the acquired signals.

Representative test scenarios were selected to provide an ocular proof of the predicted damage positions and extents. Specifically, crack positions 1, 2 (never-seen-before damage position), 9 and 17 were selected. Figure 12 shows predictions of cracks with length $l = 5\text{mm}$, Figure 13 refers to $l = 25\text{mm}$ (never-seen-before damage size), while cracks with length $l = 50\text{mm}$ are shown in Figure 14. The localization of 5mm-long cracks was generally accurate, even though the crack at position 2 was characterised with less accuracy by all the methods involved. This was expected, since that position was left out of training splits. Increasing the excitation frequency may improve the accuracy of predictions in presence of small damage, since (i) higher frequency modes carry less compressed damage-related information than lower frequency waves and (ii) LWs interact with damage with size greater or equal to half their wavelength [9]. More accurate predictions were provided in case of longer cracks. Specifically, 25mm-long cracks were correctly localised and quantified by all methods, with CNNs providing the most accurate predictions, even in case of position 2. Similar results were shown for 50mm-long cracks. CNNs and regression trees provided similar performance, while FFNNs could not accurately characterise position 2. Such results may point out that both CNNs and regression trees could satisfactorily characterise never-seen-before damage positions, while FFNNs found that task more difficult.

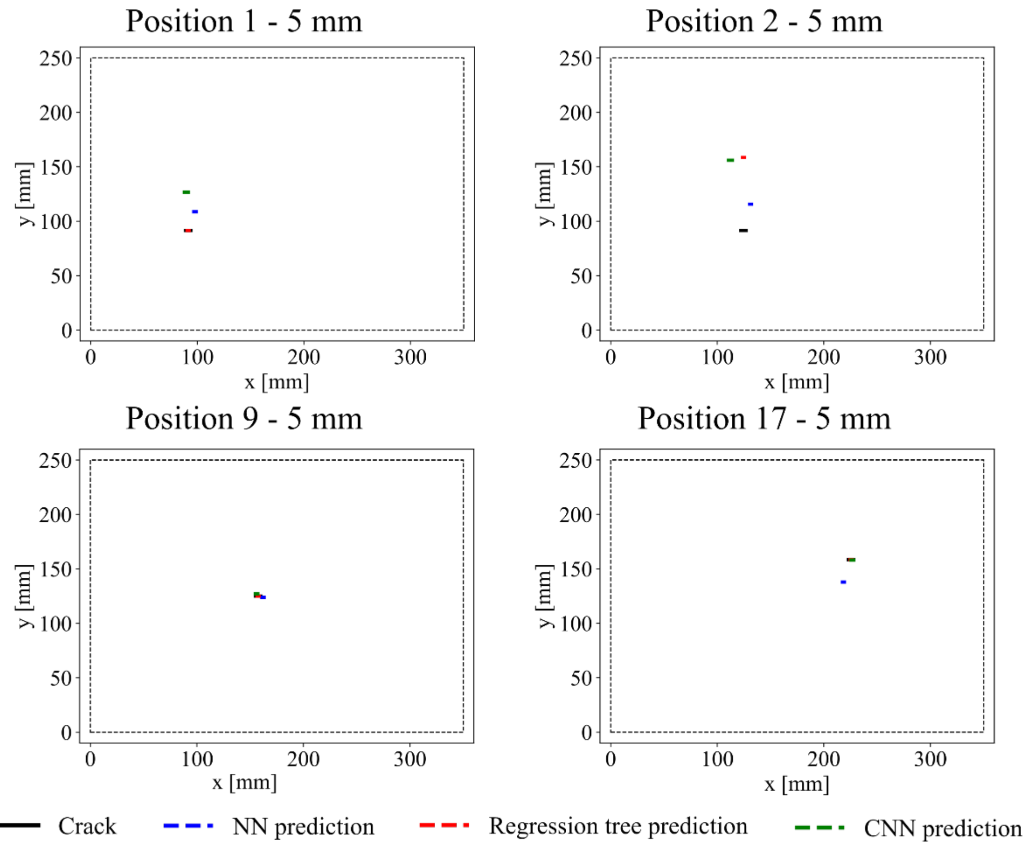


Figure 12. Damage diagnosis results with 5mm-long cracks. Black dashed lines represent plate boundaries.

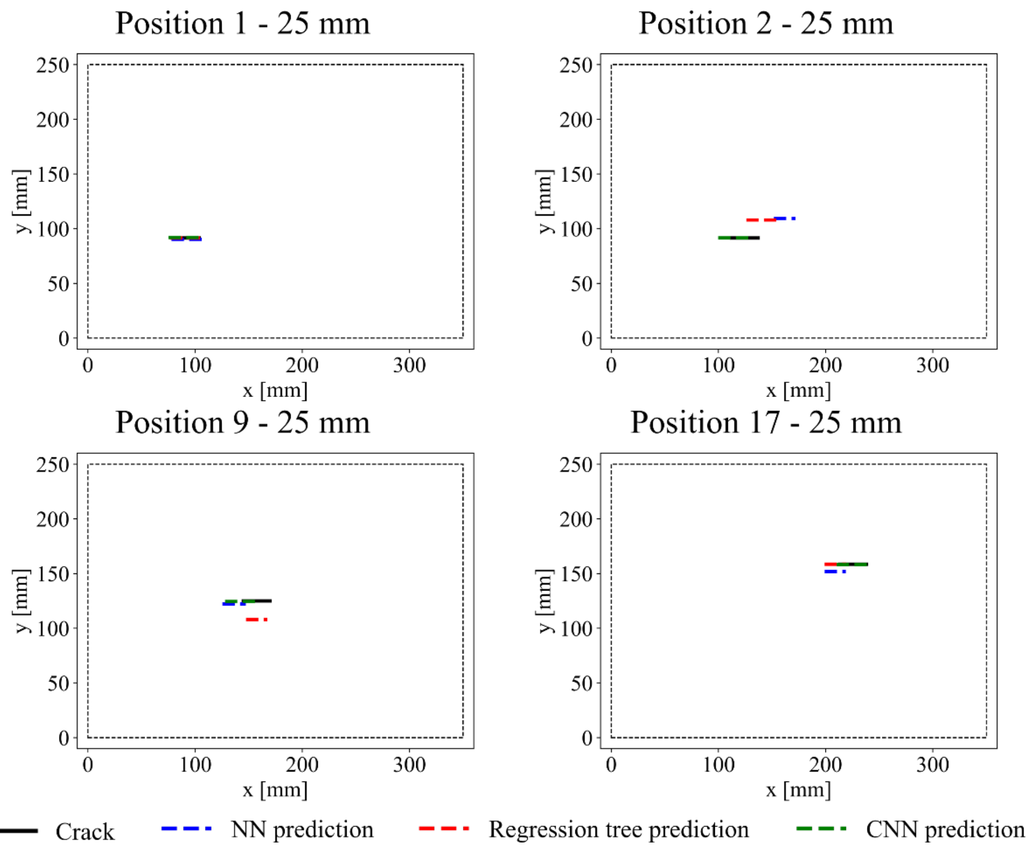


Figure 13. Damage diagnosis results with 25mm-long cracks. Black dashed lines represent plate boundaries.

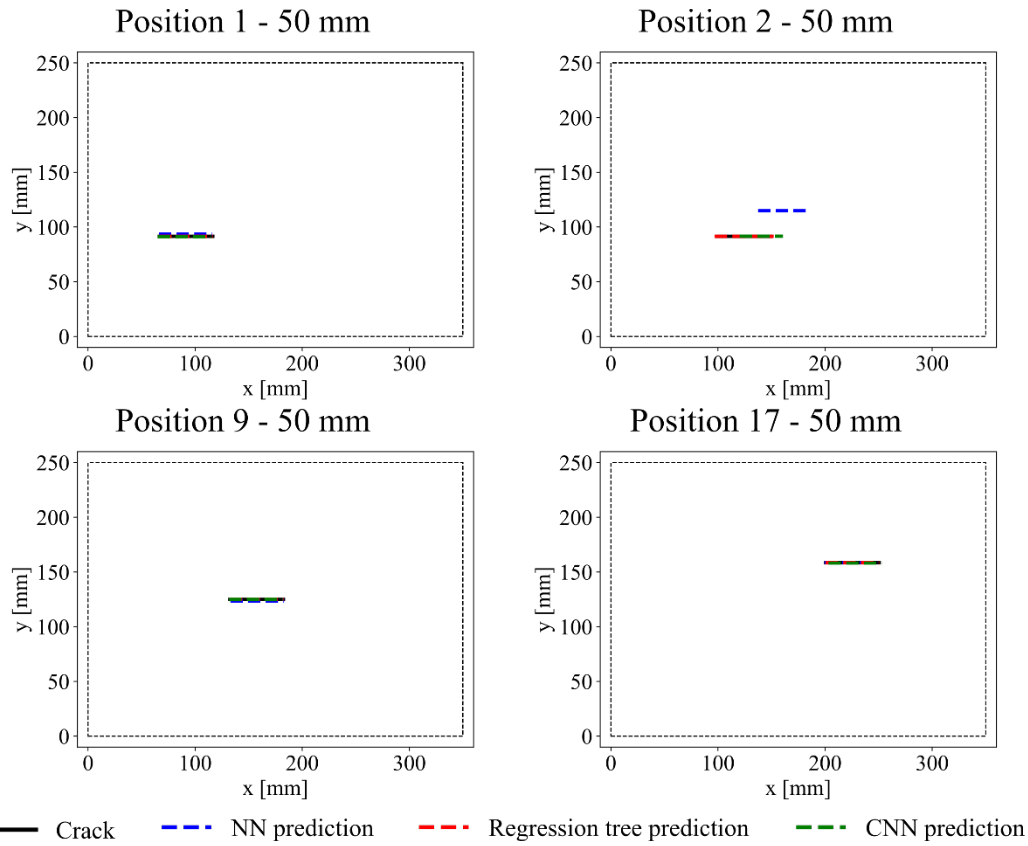


Figure 14. Damage diagnosis results with 50mm-long cracks. Black dashed lines represent plate boundaries

Finally, the performance of the proposed ML methods was compared to that of traditional tomographic image reconstruction algorithms. Among the methods present in the literature, the RAPID was selected to provide an unbiased comparison, since it computes damage probability maps processing the same information already processed by the ML algorithms. The RAPID leverages on the idea that the damage probability at a certain point (x, y) in the structure can be reconstructed by combining the information on the point relative position to the sensor pairs with the signals change with respect to the healthy baseline. Hence, the damage probability map in the reconstructed region can be computed as a linear summation of the contribution of all the sensor pairs, which is determined by combining a DI with geometrical properties. The probability at a point (x, y) in the plate is given by Equation 5 [64]:

$$P(x, y) = \sum_{k=1}^{N} p_k(x, y) = \sum_{k=1}^{N} A_k \frac{\beta - R_k(x, y)}{\beta - 1} \tag{9}$$

with [64]:

$$R_k(x,y) = \begin{cases} \dfrac{\sqrt{\left(x - x_i + \alpha(x_i - x_j)\right)^2 + \left(y - y_i + \alpha(y_i - y_j)\right)^2} + \sqrt{\left(x - x_j + \alpha(x_j - x_i)\right)^2 + \left(y - y_j + \alpha(y_j - y_i)\right)^2}}{\sqrt{\left(x_i - x_j - 2\alpha(x_i - x_j)\right)^2 + \left(y_i - y_j - 2\alpha(y_i - y_j)\right)^2}} & for\ R_k(x,y) < \beta \\ \beta & for\ R_k(x,y) \geq \beta \end{cases} \quad (1$$

where p_k represents the contribution of the k-th sensor pair, which is determined by the combination of the DI (A_k) and the geometrical weight $\frac{\beta - R_k(x,y)}{\beta - 1}$. Moreover, α and β are parameters driving the size of an effective elliptical distribution drawn around the k-th sensor pair, (x_i, y_i) and (x_j, y_j) are the coordinates of the sensor and of the actuator, respectively, while N represents the number of sensor pairs in the network. The values of the coefficients α and β were set to 0.08 and 1.07, respectively, through trial and error method. DIs were computed according to Equation 4. Positions 2 and 9 and crack length $l = 25$mm were selected as representative scenarios to compare the performance of ML and traditional methods. Results are shown in Figure 15.

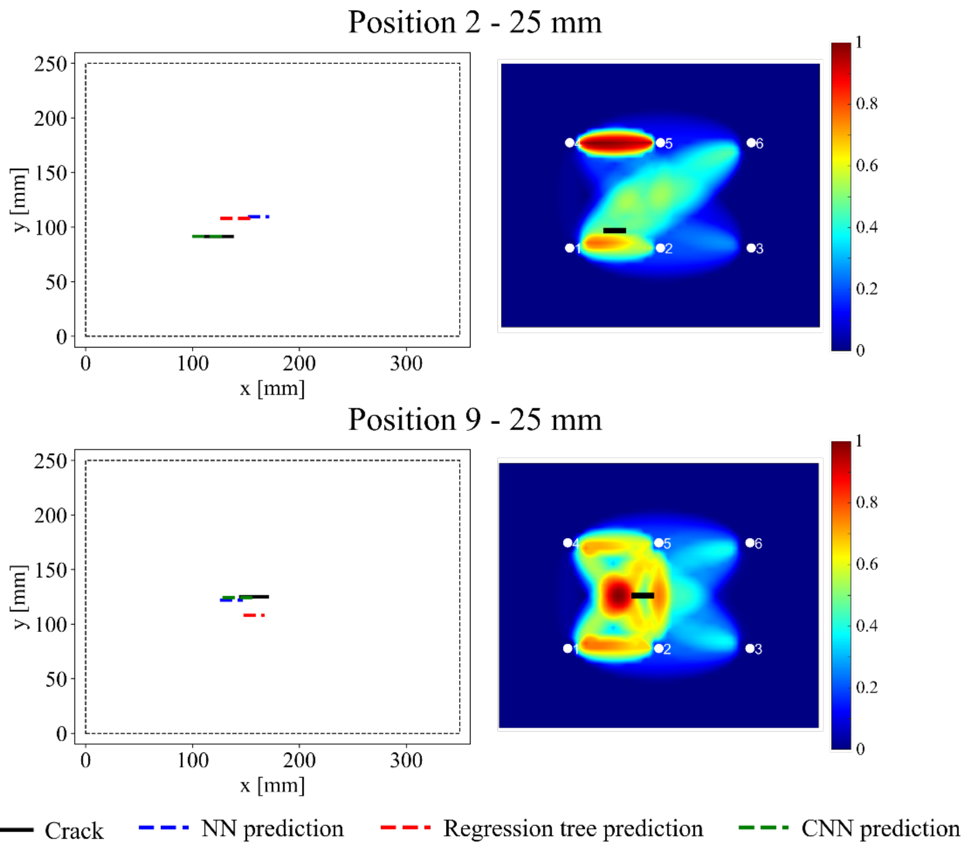


Figure 15. Damage diagnosis performance comparison: machine learning (left) vs. RAPID (right).

ML methods outperformed the tomographic algorithm both in case of position 2 (never-seen-before) and position 9. In fact, the RAPID algorithm showed limited capabilities in predicting damage size, while significant uncertainty was observed in the prediction of the damage position, which severely affected any diagnostic (localisation and quantification) conclusions. The outcome of the comparison did not strictly depend on the density of PZT devices considered in the network of sensors. In fact, increasing the number of sensors installed on the structure would boost the performance of both the RAPID and the ML methods. Additionally, a quantitative comparison may be performed by establishing a performance metrics for the RAPID algorithm, such as the extent of the high probability area, i.e., the area with damage probability greater than 50%, or the distance between the real damage position and that of the point with maximum damage probability, but this is out of the scope of this work. However, although ML methods outperformed the tomographic algorithm, an experimental and/or numerical database of LW signals in both healthy and damage conditions must be available to train ML algorithms. Even though commercial software packages may make generating numerical databases of LW signals prohibitive in terms of required time and computational resources, several methods have been proposed in the literature to overcome such problems, thus making it possible to rely on accurate and fast numerical simulations to generate realistic datasets [65–68]. To conclude, as a general guideline, if a numerical and/or experimental database of healthy and damage conditions can be set up, ML methods should be employed to boost the performance of frameworks for damage diagnosis, while in case this is not possible the RAPID algorithm should be preferred instead. However, note that the limitations of the ML methods considered in this work, especially the inability of dealing with limited amount of data, could be overcome by further improving the proposed framework exploiting the capabilities of more complex state-of-the-art solutions available in the literature [69–72]. Similarly, the proposed method could be made more general by including autoencoders to deal with varying environmental and/or operative conditions [44,73]. Such extensions of the framework are considered out of the scope of this work and will be addressed in future contributions.

4 Conclusions

In this work, ML methods for damage diagnosis using ultrasonic guided waves have been presented and compared. Specifically, a classification CNN has been employed to detect damage, and different regression methods, i.e., FFNNs, regression trees and CNNs, have been considered to predict damage position and extent. CNNs directly processed acquired signals without requiring any prior feature extraction processes, while regression trees and FFNNs worked with single-valued damage indices. Note that a CNN for damage detection was included in the framework to provide a method to conduct coherent full damage diagnosis assessments, without the need to detect damage through external sources. However, using classification networks for damage diagnosis is not a novelty of this work, and other traditional methods could be used instead. An experimentally validated numerical case study has been carried out to test and compare the ML methods. All regression methods could satisfactorily predict damage position and extent. Moreover, the ML methods also accurately identified crack positions not seen during training. However, only CNNs provided accurate characterisation of never-seen-before damage extents. Although no clear difference could be seen in the localisation performance, these results suggested that the loss of information determined by extracting DIs from acquired signals could limit the ability of regression methods to deal with unseen damage extents. Moreover, ML methods were compared to the RAPID, a traditional tomographic algorithm for damage diagnosis. Results showed that ML methods qualitatively outperformed the tomographic algorithm in terms of both localisation and quantification accuracy.

Although ML methods seemed to perform better than traditional tomographic algorithms, they also come with limitations, the main of which being the need for a database for training, that is usually costly and time consuming to generate. Two strategies may be used to overcome this limitation. On the one hand, the database may be fully generated starting from experimentally validated numerical simulations, as in the case of the application reported in this work. This would allow generating a big enough database without the often impractical need to test a real specimen for each damage scenario. This technique may also be combined with some experimental acquisitions to generate a numerically enhanced experimental database. On the other hand, if the available database is not large enough to train supervised ML algorithms, more complex unsupervised ML methods might be employed instead. In this context, the authors are working on an unsupervised convolutional autoencoder-based framework to perform damage diagnosis. This framework, which is inspired by the methodology presented in this work, only relies on healthy data to train neural networks, thus allowing reducing the number of acquisitions required to set up the damage diagnosis tool. Another limitation of the proposed method consists in the inability of identifying damage type. A solution to this issue may come from considering classification methods that process different representations of the acquired signals, e.g., hybrid time-frequency representations of LWs, such as spectrograms. Moreover, the framework still needs to be further developed and improved to overcome the common problem of false alarms in case of time varying operational and/or external conditions. This may be dealt with by using CNNs. Particularly, the temperature corresponding to each acquisition could be acquired and used as additional input of the networks. This would allow the CNNs to learn the influence of temperature on LWs propagation, thus reducing possible false alarms otherwise expected to be negatively affected by temperature variations. Future work will also involve (i) testing the proposed methodology on more complex structures, such as composite and hybrid plates, (ii) exploiting the potential of transfer learning to allow setting up the framework through numerical simulations and then using the acquired knowledge to adapt regression methods to experimental structures, and (iii) shifting from the supervised algorithms described in this study to unsupervised methods, provided that they are suitably developed and arranged.

5 Acknowledgements

The authors would like to acknowledge Prof. Claudio Sbarufatti for the precious help in building up the numerical framework to simulate the LWs propagation in the metallic plate.

6 Bibliography

1. Avci O, Abdeljaber O, Kiranyaz S, Hussein M, Gabbouj M, Inman DJ. A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. *Mechanical Systems and Signal Processing* 2021; **147**: 107077. DOI: 10.1016/j.ymssp.2020.107077.

2. Wang Y, Qiu L, Luo Y, Ding R, Jiang F. A piezoelectric sensor network with shared signal transmission wires for structural health monitoring of aircraft smart skin. *Mechanical Systems and Signal Processing* 2020; **141**: 106730. DOI: 10.1016/j.ymssp.2020.106730.

3. Qing X, Li W, Wang Y, Sun H. Piezoelectric transducer-based structural health monitoring for aircraft applications. *Sensors (Switzerland)* 2019; **19**(3): 1–27. DOI: 10.3390/s19030545.

4. Du C, Dutta S, Kurup P, Yu T, Wang X. A review of railway infrastructure monitoring using fiber optic sensors. *Sensors and Actuators, A: Physical* 2020; **303**: 111728. DOI: 10.1016/j.sna.2019.111728.

5. Ewald V, Groves RM. DESIGN OF A STRUCTURAL HEALTH MONITORING SYSTEM FOR A DAMAGE TOLERANCE FUSELAGE COMPONENT. *7th International Symposium on NDT in Aerospace*, 2015.

6. AlShorman O, Alkahatni F, Masadeh M, Irfan M, Glowacz A, Althobiani F, *et al.* Sounds and acoustic emission-based early fault diagnosis of induction motor: A review study. *Advances in Mechanical Engineering* 2021; **13**(2). DOI: 10.1177/1687814021996915/ASSET/IMAGES/LARGE/10.1177_1687814021996915-FIG2.JPEG.

7. Alshorman O, Irfan M, Saad N, Zhen D, Haider N, Glowacz A, *et al.* A Review of Artificial Intelligence Methods for Condition Monitoring and Fault Diagnosis of Rolling Element Bearings for Induction Motor. *Shock and Vibration* 2020; **2020**. DOI: 10.1155/2020/8843759.

8. Glowacz A. Thermographic Fault Diagnosis of Shaft of BLDC Motor. *Sensors 2022, Vol 22, Page 8537* 2022; **22**(21): 8537. DOI: 10.3390/S22218537.

9. Su Z, Ye L. *Identification of Damage Using Lamb Waves*. vol. 48. London: Springer London; 2009. DOI: 10.1007/978-1-84882-784-4.

10. Mitra M, Gopalakrishnan S. Guided wave based structural health monitoring: A review. *Smart Materials and Structures* 2016; **25**(5): 053001. DOI: 10.1088/0964-1726/25/5/053001.

11. Lee BC, Staszewski WJ. Lamb wave propagation modelling for damage detection: I. Two-dimensional analysis. *Smart Materials and Structures* 2007; **16**(2): 249–259. DOI: 10.1088/0964-1726/16/2/003.

12. Lee BC, Staszewski WJ. Modelling of Lamb waves for damage detection in metallic structures: Part II. Wave interactions with damage. *Smart Materials and Structures* 2003; **12**(5): 815–824. DOI: 10.1088/0964-1726/12/5/019.

13. Mori N, Biwa S, Kusaka T. Damage localization method for plates based on the time reversal of the mode-converted Lamb waves. *Ultrasonics* 2019; **91**: 19–29. DOI: 10.1016/j.ultras.2018.07.007.

14. Su Z, Ye L, Lu Y. Guided Lamb waves for identification of damage in composite structures:

A review. *Journal of Sound and Vibration* 2006; **295**(3–5): 753–780. DOI: 10.1016/j.jsv.2006.01.020.

15. Guan R, Lu Y, Duan W, Wang X. Guided waves for damage identification in pipeline structures: A review. *Structural Control and Health Monitoring* 2017; **24**(11): e2007. DOI: 10.1002/STC.2007.

16. Dawson AJ, Michaels JE, Michaels TE. Isolation of ultrasonic scattering by wavefield baseline subtraction. *Mechanical Systems and Signal Processing* 2016; **70**–**71**: 891–903. DOI: 10.1016/j.ymssp.2015.09.008.

17. Zhao X, Gao H, Zhang G, Ayhan B, Yan F, Kwan C, *et al.* Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. Defect detection, localization and growth monitoring. *Smart Materials and Structures* 2007; **16**(4): 1208–1217. DOI: 10.1088/0964-1726/16/4/032.

18. Hay TR, Royer RL, Gao H, Zhao X, Rose JL. A comparison of embedded sensor Lamb wave ultrasonic tomography approaches for material loss detection. *Smart Materials and Structures* 2006; **15**(4): 946–951. DOI: 10.1088/0964-1726/15/4/007.

19. Zhao X, Royer RL, Owens SE, Rose JL. Ultrasonic Lamb wave tomography in structural health monitoring. *Smart Materials and Structures* 2011; **20**(10). DOI: 10.1088/0964-1726/20/10/105002.

20. Gao H, Shi Y, Rose JL. Guided wave tomography on an aircraft wing with leave in place sensors. *AIP Conference Proceedings*, vol. 760, American Institute of PhysicsAIP; 2005. DOI: 10.1063/1.1916887.

21. Yuan S, Liang D, Shi L, Zhao X, Wu J, Li G, *et al.* Recent progress on distributed structural health monitoring research at NUAA. *Journal of Intelligent Material Systems and Structures*, vol. 19, Sage PublicationsSage UK: London, England; 2008. DOI: 10.1177/1045389X07087302.

22. Trendafilova I, Manoach E. Vibration-based damage detection in plates by using time series analysis. *Mechanical Systems and Signal Processing* 2008; **22**(5): 1092–1106. DOI: 10.1016/j.ymssp.2007.11.020.

23. Wu Z, Qing XP, Ghosh K, Karbhar V, Chang FK. Health monitoring of bonded composite repair in bridge rehabilitation. *Smart Materials and Structures* 2008; **17**(4): 045014. DOI: 10.1088/0964-1726/17/4/045014.

24. Qing XP, Chan HL, Beard SJ, Kumar A. An active diagnostic system for structural health monitoring of rocket engines. *Journal of Intelligent Material Systems and Structures* 2006; **17**(7): 619–628. DOI: 10.1177/1045389X06059956.

25. Michaels JE, Michaels TE. An integrated strategy for detection and imaging of damage using a spatially distributed array of piezoelectric sensors. *Health Monitoring of Structural and Biological Systems 2007*, vol. 6532, SPIE; 2007. DOI: 10.1117/12.715438.

26. Park S, Yun CB, Roh Y, Lee JJ. PZT-based active damage detection techniques for steel bridge components. *Smart Materials and Structures* 2006; **15**(4): 957–966. DOI: 10.1088/0964-1726/15/4/009.

27. Liu Z, Zhong X, Dong T, He C, Wu B. Delamination detection in composite plates by synthesizing time-reversed Lamb waves and a modified damage imaging algorithm based on RAPID. *Structural Control and Health Monitoring* 2017; **24**(5): e1919. DOI: 10.1002/stc.1919.

28. Gao F, Shao Y, Hua J, Zeng L, Lin J. Enhanced wavefield imaging method for impact damage

detection in composite laminates via laser-generated Lamb waves. *Measurement: Journal of the International Measurement Confederation* 2021; **173**: 108639. DOI: 10.1016/j.measurement.2020.108639.

29. Wang D, Ye L, Lu Y, Su Z. Probability of the presence of damage estimated from an active sensor network in a composite panel of multiple stiffeners. *Composites Science and Technology* 2009; **69**(13): 2054–2063. DOI: 10.1016/j.compscitech.2008.11.005.

30. Xia Q, Liu Y, Lu Y, Cao S, Zhang H, Ma S. A modified damage index probability imaging algorithm based on delay-and-sum imaging for synthesizing time-reversed lamb waves. *Journal of Vibroengineering* 2019; **21**(8): 2140–2147. DOI: 10.21595/jve.2019.20568.

31. Azuara G, Barrera E, Ruiz M, Bekas D. Damage Detection and Characterization in Composites Using a Geometric Modification of the RAPID Algorithm. *IEEE Sensors Journal* 2020; **20**(4): 2084–2093. DOI: 10.1109/JSEN.2019.2950748.

32. Sbarufatti C, Manson G, Worden K. A numerically-enhanced machine learning approach to damage diagnosis using a Lamb wave sensing network. *Journal of Sound and Vibration* 2014; **333**(19): 4499–4525. DOI: 10.1016/j.jsv.2014.04.059.

33. Dworakowski Z, Dragan K, Stepinski T. Artificial neural network ensembles for fatigue damage detection in aircraft. *Journal of Intelligent Material Systems and Structures* 2017; **28**(7): 851–861. DOI: 10.1177/1045389X16657428.

34. De Fenza A, Sorrentino A, Vitiello P. Application of Artificial Neural Networks and Probability Ellipse methods for damage detection using Lamb waves. *Composite Structures* 2015; **133**: 390–403. DOI: 10.1016/j.compstruct.2015.07.089.

35. Cui R, Azuara G, Lanza di Scalea F, Barrera E. Damage imaging in skin-stringer composite aircraft panel by ultrasonic-guided waves using deep learning with convolutional neural network: *Https://DoiOrg/101177/14759217211023934* 2021; **0**(0): 1–16. DOI: 10.1177/14759217211023934.

36. Hu C, Yang B, Yan J, Xiang Y, Zhou S, Xuan FZ. Damage localization in pressure vessel by guided waves based on convolution neural network approach. *Journal of Pressure Vessel Technology, Transactions of the ASME* 2020; **142**(6). DOI: 10.1115/1.4047213/1083797.

37. Su C, Jiang M, Lv S, Lu S, Zhang L, Zhang F, *et al.* Improved Damage Localization and Quantification of CFRP Using Lamb Waves and Convolution Neural Network. *IEEE Sensors Journal* 2019; **19**(14): 5784–5791. DOI: 10.1109/JSEN.2019.2908838.

38. Ewald V, Groves RM, Benedictus R. DeepSHM: a deep learning approach for structural health monitoring based on guided Lamb wave technique. In: Wang KW, Sohn H, Huang H, Lynch JP, editors. *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019*, vol. 10970, SPIE; 2019. DOI: 10.1117/12.2506794.

39. Chang CS, Lee YC. Ultrasonic touch sensing system based on lamb waves and convolutional neural network. *Sensors (Switzerland)* 2020; **20**(9): 1–15. DOI: 10.3390/s20092619.

40. Zhang S, Li CM, Ye W. Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network. *Mechanical Systems and Signal Processing* 2021; **147**: 107107. DOI: 10.1016/j.ymssp.2020.107107.

41. Rautela M, Gopalakrishnan S. Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks. *Expert Systems with Applications* 2021; **167**: 114189. DOI: 10.1016/J.ESWA.2020.114189.

42. Rautela M, Senthilnath J, Moll J, Gopalakrishnan S. Combined two-level damage identification strategy using ultrasonic guided waves and physical knowledge assisted machine

learning. *Ultrasonics* 2021; **115**: 106451. DOI: 10.1016/j.ultras.2021.106451.

43. Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. *Classification and Regression Trees* 2017: 1–358. DOI: 10.1201/9781315139470/CLASSIFICATION-REGRESSION-TREES-LEO-BREIMAN.

44. Cadini F, Lomazzi L, Ferrater Roca M, Sbarufatti C, Giglio M. Neutralization of temperature effects in damage diagnosis of MDOF systems by combinations of autoencoders and particle filters. *Mechanical Systems and Signal Processing* 2022; **162**: 108048. DOI: 10.1016/j.ymssp.2021.108048.

45. Hubel DH, Wiesel TN. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology* 1968; **195**(1): 215–243. DOI: 10.1113/jphysiol.1968.sp008455.

46. LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, *et al. Handwritten Digit Recognition with a Back-Propagation Network*. vol. 2. 1989.

47. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 1998; **86**(11): 2278–2323. DOI: 10.1109/5.726791.

48. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 2012; **25**(1097–1105).

49. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, *et al.* Recent advances in convolutional neural networks. *Pattern Recognition* 2018; **77**: 354–377. DOI: 10.1016/j.patcog.2017.10.013.

50. Boureau YL, Ponce J, Fr JP, Lecun Y. A Theoretical Analysis of Feature Pooling in Visual Recognition. *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel: 2010.

51. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8689 LNCS, Springer Verlag; 2014. DOI: 10.1007/978-3-319-10590-1_53.

52. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, *et al.* ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 2015; **115**(3): 211–252. DOI: 10.1007/s11263-015-0816-y.

53. Rose JL. *Ultrasonic guided waves in solid media*. vol. 9781107048. 2014. DOI: 10.1017/CBO9781107273610.

54. Lim HJ, Sohn H. Online Stress Monitoring Technique Based on Lamb-wave Measurements and a Convolutional Neural Network Under Static and Dynamic Loadings. *Experimental Mechanics* 2020; **60**(2): 171–179. DOI: 10.1007/s11340-019-00546-8.

55. Gurtin ME. The Linear Theory of Elasticity. *Linear Theories of Elasticity and Thermoelasticity*, Springer Berlin Heidelberg; 1973. DOI: 10.1007/978-3-662-39776-3_1.

56. Achenbach JD. *Wave Propagation in Elastic Solids*. Amsterdam: North-Holland Publishing Company; 1973. DOI: 10.1016/c2009-0-08707-8.

57. Sun H, Peng L, Wang S, Huang S, Qu K. Development of Frequency-Mixed Point-Focusing Shear Horizontal Guided-Wave EMAT for Defect Inspection Using Deep Neural Network. *IEEE Transactions on Instrumentation and Measurement* 2021; **70**. DOI: 10.1109/TIM.2020.3033941.

58. Hameed MS, Li Z, Zheng K. Damage Detection Method Based on Continuous Wavelet Transformation of Lamb Wave Signals. *Applied Sciences 2020, Vol 10, Page 8610* 2020; **10**(23): 8610. DOI: 10.3390/APP10238610.

59. Nocedal J, Wright SJ. *Numerical Optimization*. 2nd ed. Springer New York; 2006. DOI: 10.1007/978-0-387-40065-5.

60. Keras. Keras: the Python deep learning API. https://keras.io/ [accessed May 25, 2021].

61. TensorFlow.

62. Adam optimiser. https://keras.io/api/optimizers/adam/ [accessed May 25, 2021].

63. Kingma DP, Ba JL. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR; 2015.

64. BONET M, ECKSTEIN B, LOENDERSLOOT R, WIERACH P. Identification of Barely Visible Impact Damages on a Stiffened Composite Panel with a Probability-based Approach 2015. DOI: 10.12783/shm2015/290.

65. Duczek S. Higher order finite elements and the fictitious domain concept for wave propagation analysis. Otto-von-Guericke-Universität Magdeburg, 2014.

66. Gopalakrishnan S, Ruzzene M, Hanagud S. *Computational Techniques for Structural Health Monitoring*. 1st ed. London: Springer London; 2011. DOI: 10.1007/978-0-85729-284-1.

67. Willberg C, Duczek S, Vivar-Perez JM, Ahmad ZAB. Simulation methods for guided wave-based structural health monitoring: A review. *Applied Mechanics Reviews* 2015; **67**(1). DOI: 10.1115/1.4029539/443651.

68. Zhang J, Ankit A, Gravenkamp H, Eisenträger S, Song C. A massively parallel explicit solver for elasto-dynamic problems exploiting octree meshes. *CMAME* 2021; **380**: 113811. DOI: 10.1016/j.cma.2021.113811.

69. Zhang W, Li X, Ma H, Luo Z, Li X. Universal Domain Adaptation in Fault Diagnostics with Hybrid Weighted Deep Adversarial Learning. *IEEE Transactions on Industrial Informatics* 2021. DOI: 10.1109/TII.2021.3064377.

70. Li X, Xu Y, Li N, Yang B, Lei Y. Remaining useful life prediction with partial sensor malfunctions using deep adversarial networks. *IEEE/CAA Journal of Automatica Sinica* 2022. DOI: 10.1109/JAS.2022.105935.

71. Xing H, Xiao Z, Qu R, Zhu Z, Zhao B. An Efficient Federated Distillation Learning System for Multitask Time Series Classification. *IEEE Transactions on Instrumentation and Measurement* 2022; **71**. DOI: 10.1109/TIM.2022.3201203.

72. Xing H, Xiao Z, Zhan D, Luo S, Dai P, Li K. SelfMatch: Robust semisupervised time-series classification with self-distillation. *International Journal of Intelligent Systems* 2022; **37**(11): 8583–8610. DOI: 10.1002/INT.22957.

73. Parziale M, Lomazzi L, Giglio M, Cadini F. Vibration-based structural health monitoring exploiting a combination of convolutional neural networks and autoencoders for temperature effects neutralization. *Structural Control and Health Monitoring* 2022; **29**(11). DOI: 10.1002/STC.3076.