Convolutional neural networks for structural damage localization on digital twins

Marco Parola[1][0000−0003−4871−4902], Federico A. Galatolo[1][0000−0001−7193−3754], Matteo Torzoni[2][0000−0003−0027−9788], and Mario G.C.A. Cimino[1][0000−0002−1031−1959]

[1] Department of Information Engineering, University of Pisa,
Largo L. Lazzarino 1, Pisa, Italy
{marco.parola, federico.galatolo, mario.cimino}@ing.unipi.it

[2] Department of Civil and Environmental Engineering, Politecnico di Milano,
Piazza L. da Vinci 32, Milano, Italy
matteo.torzoni@polimi.it

Abstract. Structural health monitoring (SHM) using IoT sensor devices plays a crucial role in the preservation of civil structures. SHM aims at performing an accurate damage diagnosis of a structure, that consists of identifying, localizing, and quantify the condition of any significant damage, to keep track of the relevant structural integrity. deep learning (DL) architectures have been progressively introduced to enhance vibration-based SHM analyses: supervised DL approaches are integrated into SHM systems because they can provide very detailed information about the nature of damage compared to unsupervised DL approaches. The main drawback of supervised approach is the need for human intervention to appropriately label data describing the nature of damage, considering that in the SHM context, providing labeled data requires advanced expertise and a lot of time. To overcome this limitation, a key solution is a digital twin relying on physics-based numerical models to reproduce the structural response in terms of the vibration recordings provided by the sensor devices during a specific events to be monitored. This work presents a comprehensive methodology to carry out the damage localization task by exploiting a convolutional neural network (CNN) and parametric model order reduction (MOR) techniques to reduce the computational burden associated with the construction of the dataset on which the CNN is trained. Experimental results related to a pilot application involving a sample structure, show the potential of the proposed solution and the reusability of the trained system in presence of different loading scenarios.

Keywords: Convolutional Neural Network · IoT · Digital Twin · Structural Health Monitoring · Damage Localization.

1 Introduction and background

Civil structures, whether buildings, bridges, oil and gas pipelines, are subject to several external actions and sources of degradation that might compromise their structural performance. This can happen due to a faulty construction process, lack of quality control, or unexpected loadings, environmental actions and natural hazards such as earthquakes. In order to keep track of the structural health, and to quickly react before a major damage occurs, autonomus damage identification systems provide a suitable framework to perform systematic diagnostic and prognostic activities, ultimately allowing for timely maintenance actions with a direct impact on reducing the operating costs.

In the last years, increasingly sophisticated structural health monitoring (SHM) systems have been developed to keep under control the structural health state, through the implementation of different levels of damage identification, such as detection, localization and quantification, possibly with a quantification of the impact of relevant environmental effects, see e.g. [1,2]. A SHM system is usually comprised by different components: a sensor network deployed to collect vibration and environmental data; a data transmission unit; a storage unit; a SHM data analysis software. The outcome of the monitoring procedure is then displayed to the user via reports or web platforms.

Recently, many vibration-based SHM strategies relying upon deep learning (DL) architectures have been proposed to address different SHM tasks, see e.g. [3,4,5], by exploiting their capability to automatize the selection and extraction of damage-sensitive features, that traditional algorithms would fail to detect [6]. Focusing on these data-driven strategies, the SHM problem can be addressed either in a supervised or an unsupervised way. The former exploits labeled input-output pairs, with vibration structural response data being the inputs and the values of the sought damage parameters being the corresponding outputs. On the other hand, unsupervised algorithms are often adopted to discover damage-sensitive patterns in the input data, without the need of providing a corresponding output label. The implementation of an unsupervised damage detection strategy usually involves two phases: (*i*) the identification of stable key parameters reflecting the undamaged structural health state by means of appropriate behavioral model of the structure; (*ii*) the detection of a persistent variation in such parameters over time, by relying upon the underlying idea that when the structure suffers damage, a deviation from the reference condition can be observed in terms of vibration response [7].

Since unsupervised learning methods can only be effective in detecting the presence of a structural damage, without allowing to obtain clear and explicit information about location, severity and type of damage, a supervised learning strategy is adopted in this paper. However, dealing with civil structures, experimental labeled data referred to the possible damage states can not be obtained in practice. To overcome this limitation, a key solution is provided by the digital twin (DT) paradigm, and in particular by the physics-based numerical models comprising the DT of the structure to be monitored. Indeed, the latter enable to systematically simulate the vibration recordings provided by IoT devices for specific damage and operational conditions [8].

A DT is built upon three components: a physical asset in the real-world (physical twin), a digital model of the structure in a computerized environment, and the integration of data and information that tie the physical entities and their virtual representation together [9]. For a successful DT implementation, it is crucial to properly identify all the involved physical entities and processes in the real-world and their digital counterparts, as well as the interconnection between them, in terms of the exchanged data. The process of implementing a DT is called digital transformation, and in this work it is (partially) achieved by means of a physics-based numerical model of the monitored structure relying on the finite element (FE) method and of a DL-based model for damage identification.

As the number of involved degrees of freedom increases, the computational cost associated to the solution of a FE model grows, and the assembly of synthetic datasets accounting for different input parameters easily becomes prohibitive [10]. To this aim, a reduced-order modeling strategy for parametrized systems is adopted by relying on the reduced basis method [11] in order to set a cheaper, yet accurate, reduced-order model (ROM), ultimately allowing to speed up data generation phase required to train the DL model in a supervised fashion.

This work deepens and extends a former research activity presented in [12], by proposing a comprehensive approach to solve the damage localization task. The obtained results testify the capabilities of the proposed approach to perform real-time damage localization, as well as the effectiveness of such diagnostic framework against operational variability and measurement noise. In addition, the beneficial effect of implementing a hyperparameter optimization strategy is also considered, as reported to yield a fair improvement in the damage localization performance.

The reminder of the paper organized as follows: the methods for developing the digital twin, and involving the generation of synthetic datasets and the use of DL-based architectures for SHM purposes are described in Section 2; the application of the methodology to the case study of a two-story shear building is discussed in Section 4; conclusions and future developments are finally reported in Section 5.

2 Methodology and methods

The proposed methodology is described in the following. Specifically: in Section 2.1, we detail the numerical models comprising the DT, adopted to populate the synthetic dataset; in Section 2.2, we frame the damage localization task as a classification problem handling a set of predefined damage scenarios by means of a convolutional neural network (CNN).

2.1 Digital twin design

Physics-based numerical model The virtual representation of the structure to be monitored is obtained by relying upon a high-fidelity full-order model (FOM), describing its dynamic response under the applied loadings, according to the Newton's second law of motion and under the assumption of a linearized kinematic. By modeling the structure as a linear-elastic continuum and by space-discretizing the governing equation by means of a FE mesh, its dynamic response is described by the following semi-discretized form of the elasto-dynamic problem:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{d}}(t) + \mathbf{C}\dot{\mathbf{d}}(t) + \mathbf{K}(\mathit{\Delta}, l)\mathbf{d}(t) = \mathbf{f}(t, \boldsymbol{\eta}) \ , t \in (0, T) \\ \mathbf{d}(0) = \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) = \dot{\mathbf{d}}_0 \ , \end{cases} \tag{1}$$

where: $t \in (0, T)$ denotes time; $\mathbf{d}(t), \dot{\mathbf{d}}(t), \ddot{\mathbf{d}} \in \mathbb{R}^{\mathscr{M}}$ are the vectors of nodal displacements, velocities and accelerations, respectively; $\mathscr{M}$ is the number of degrees of freedom (dofs); $\mathbf{M} \in \mathbb{R}^{\mathscr{M} \times \mathscr{M}}$ is the mass matrix; $\mathbf{C} \in \mathbb{R}^{\mathscr{M} \times \mathscr{M}}$ is the damping matrix, assembled according to the Rayleigh's model; $\mathbf{K}(\mathit{\Delta}, l) \in^{\mathscr{M} \times \mathscr{M}}$ is the stiffness matrix, with $\mathit{\Delta}$ and l the parameters providing its dependence on damage as specified below; $f(t, \boldsymbol{\eta}) \ \in \mathbb{R}^{\mathscr{M}}$ is the vector of nodal forces associated to the operational conditions ruled by means of N_η parameters through the vector $\boldsymbol{\eta} \in \mathbb{R}^{N_\eta}$; $\mathbf{d}_0 \in \mathbb{R}^{\mathscr{M}}$ and $\dot{\mathbf{d}}_0 \in \mathbb{R}^{\mathscr{M}}$ are the initial conditions at $t = 0$, in terms of nodal displacements and velocities, respectively. The relevant parametric input space is assumed to display a uniform probability distribution for each parameter.

As typically done in simulation-based SHM, damage is modeled as a selective degradation of the material stiffness of amplitude $l \in \mathbb{R}$, taking place within the pre-designated region labeled by $\mathit{\Delta} \in \{\mathit{\Delta}_0, \ldots, \mathit{\Delta}_{N_d}\}$, with $\mathit{\Delta}_0$ identifying the damage-free baseline and all the others being referred to specific damage scenarios undergone by the structure among a set of predefined N_d damage states. These latter are defined on the basis of structural response, loading conditions, and aging processes of materials. In this work, l is not considered part of the label, as only the localization of damage is addressed.

Dataset generation The generation of the training instances is carried out by advancing in time the solution of the physics-based model of the structure using the Newmark time integration scheme. Either nodal displacements or accelerations recordings $\boldsymbol{\delta}_i = \boldsymbol{\delta}_i(\mathit{\Delta}, l, \boldsymbol{\eta}) \in \mathbb{R}^L$ in $(0, T)$, each including L measurements, are collected at N_s predefined locations where sensing devices are supposed to be installed, with $i = 1, \ldots, N_s$. The measurements are acquired with a sampling frequency $\mathscr{f}$ and for an observation time window $(0, T)$, short enough to assume constant operational and damage conditions, such that $T = (L - 1)/\mathscr{f}$. The training set $\mathbf{D} \in \mathbb{R}^{L \times N_s \times N_o}$ is then built from the assembly of N_o instances, each one shaped as a multivariate time series comprised by N_s arrays of L measurements and obtained by sampling the parametric input space of the numerical model via latin hypercube rule.

In order to obtain a high quality dataset $\mathbf{D}$ to train the DL model, the number of required instances may be extremely high, thus making the computational cost associated to the data generation process potentially very high. To this aim, the FOM is replaced by a cheaper, yet accurate, projection-based ROM by relying on the reduced basis method [11], following the same strategy adopted in [13,4,10].

By relying upon the proper orthogonal decomposition (POD)-Galerkin approach, the solution to Problem (1) is approximated, in terms of displacements, as $\mathbf{d}(t, \varDelta, l, \boldsymbol{\eta}) \approx \mathbf{W}\mathbf{d}_R(t, \varDelta, l, \boldsymbol{\eta})$, which is a linear combination of $\mathscr{M}_R \ll \mathscr{M}$ basis functions $\mathbf{w}_r \in \mathbb{R}^{\mathscr{M}}$, $r = 1, \dots, \mathscr{M}_R$, gathered in the projection matrix $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_R] \in \mathbb{R}^{\mathscr{M} \times \mathscr{M}_R}$, with $\mathbf{d}_R(t, \varDelta, l, \boldsymbol{\eta}) \in \mathbb{R}^{\mathscr{M}_R}$ being the vector of unknown POD-coefficients

By enforcing the orthogonality between the residual and the subspace spanned by the first $\mathscr{M}_R$ POD-modes through a Galerkin projection, the following $\mathscr{M}_R$-dimensional dynamical system is obtained:

$$\begin{cases} \mathbf{M}_R\ddot{\mathbf{d}}(t) + \mathbf{C}_R\dot{\mathbf{d}}(t) + \mathbf{K}_R(\varDelta, l)\mathbf{d}_R(t) = \mathbf{f}_R(t, \boldsymbol{\eta}) \ , t \in (0, T) \\ \mathbf{d}_R(0) = \mathbf{W}^\top \mathbf{d}_0 \\ \dot{\mathbf{d}}_R(0) = \mathbf{W}^\top \dot{\mathbf{d}}_0 \ , \end{cases} \tag{2}$$

Here, the reduced arrays play the same role of their HF counterparts, yet with dimension ruled by $\mathscr{M}_R$ instead of $\mathscr{M}$, according to:

$$\mathbf{M}_R \equiv \mathbf{W}^\top \mathbf{M}\mathbf{W} \ , \ \mathbf{C}_R \equiv \mathbf{W}^\top \mathbf{C}\mathbf{W} \ , \ \mathbf{K}_R \equiv \mathbf{W}^\top \mathbf{K}\mathbf{W} \ , \ \mathbf{f}_R \equiv \mathbf{W}^\top \mathbf{f}. \tag{3}$$

The approximated solution is then recovered by back-projecting the ROM solution, via $\mathbf{d}(t) \approx \mathbf{W}\mathbf{d}_R(t)$, or $\ddot{\mathbf{d}}(t) \approx \mathbf{W}\ddot{\mathbf{d}}_R(t)$ depending on the handled measurements.

The projection matrix $\mathbf{W}$ is obtained by performing a singular value decomposition of a snapshot matrix $\mathbf{S} = [\mathbf{d}_1, \dots, \mathbf{d}_{\mathscr{S}}] \in \mathbb{R}^{\mathscr{M} \times \mathscr{S}}$, assembled from $\mathscr{S}$ snapshots of the FOM, namely solutions in terms of time histories of nodal displacements, obtained for different values of the parameters, as

$$\mathbf{S} = \mathbf{P}\boldsymbol{\Sigma}\mathbf{Z}^\top \ , \tag{4}$$

where: $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_{\mathscr{M}}] \in \mathbb{R}^{\mathscr{M} \times \mathscr{M}}$ is an orthogonal matrix, whose columns are the left singular vectors of $\mathbf{S}$; $\boldsymbol{\Sigma} \in \mathbb{R}^{\mathscr{M} \times \mathscr{S}}$ is a pseudo-diagonal matrix collecting the singular values of $\mathbf{S}$, arranged so that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\mathscr{R}} \geq 0$, $\mathscr{R} = min(\mathscr{S}, \mathscr{M})$ being the rank of $\mathbf{S}$; $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_{\mathscr{S}}] \in \mathbb{R}^{\mathscr{S} \times \mathscr{S}}$ is an orthogonal matrix, whose columns are the right singular vectors of $\mathbf{S}$.

The ROM order $\mathscr{M}_R$ is set by adopting a standard energy-content criterion by prescribing a tolerance ϵ on the fraction of energy content to be disregarded in the approximation, according to:

$$\frac{\sum_{m=1}^{\mathscr{M}_R} (\sigma_m)^2}{\sum_{m=1}^{\mathscr{R}} (\sigma_m)^2} \geq 1 - \epsilon^2 \ , \tag{5}$$

that is the energy retained by the last $\mathscr{R} - \mathscr{M}_R$ POD-modes is equal or smaller than ϵ^2.

2.2 Deep learning for the damage localization

In this paper, we propose the use of one-dimensional (1D) CNNs to solve the damage localization task, framed as a multiclass classification problem. A classification task involves the prediction of an output class label based on a given input. In this case, the output labels to be predicted identify a set of predefined damage scenarios, each referring to a different damage location.

Originally developed within the computer vision community, convolutional layers have quickly become a first choice to solve several problems, outperforming alternative methods [14,15]. They feature good relational inductive biases such as the locality and translational equivariance (parameter sharing) of convolutional kernels, which prove highly effective to analyze multivariate time series while improving the relevant computational efficiency. In a convolutional layer, the kernel filters feature a charcteristic size controlling the width of the local receptive field on its input. Each convolutional layer simultaneously applies multiple kernel filters throughout its input, resulting in multiple activation maps, called feature maps, each one providing the location and strength of the relevant convolutional kernel in the input.

In a deep learning framework, a multi-label classification task can be addressed by prescribing number of computational neurons in the last layer of the adopted neural network equal to the number of possible target labels. The resulting architecture is typically trained to solve the underlying classification task by minimizing the categorical cross-entropy between the predicted and target label classes:

$$H(\boldsymbol{b}, \widehat{\boldsymbol{b}}) = -\sum_{j=0}^{N_d} b_j \ log \widehat{b}_j \ , \tag{6}$$

where $\boldsymbol{b} = \{b_0, \dots, b_{N_d}\}^\top \in \mathbb{B}^{N_d}$ and $\widehat{\boldsymbol{b}} = \{\widehat{b}_0, \dots, \widehat{b}_{N_d}\}^\top \in \mathbb{R}^{N_d}$ are two vectors gathering the Boolean indexes b_j, whose value is 1 if the target class for the current instance is j and 0 otherwise, and the confidence levels $\widehat{b}_j$ by which the current instance is assigned to the j-th damage class, respectively.

To evaluate the performance of the adopted DL model against the considered multi-class classification problem, the accuracy, precision, and recall indicators are used, as follows:

$$\texttt{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \ ; \tag{7}$$

$$\texttt{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \ ; \tag{8}$$

$$\texttt{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \ . \tag{9}$$

Herein, TP denotes the amount of true positives, FP denotes the amount of false positives, TN denotes the amount of true ngatives, FN the amount of false negatives.

3 Case study

The damage location capabilities of the proposed methodology are assessed with reference to the virtual health monitoring of the building reported in Fig. 1a. Within a digital twin perspective, this physical asset is equipped with a network of sensors, for instance using commercial IoT devices as the one reported in Fig. 1b. This latter is a mono-axial wireless device, useful to acquire displacement measurements with an accuracy of 0.01 mm[1] A crucial aspect to take into account in the practice is the synchronization between IoT devices, which is a critical requirement for system operation. This latter is not a zero-cost process and different protocols can be adopted to meet the prescribed requirements, depending on the system type [16].

(a) (b)

Fig. 1: Pilot example: (a) physical twin to be monitored; (b) exemplary IoT device suitable for dynamic monitoring.

In the following, the corresponding digital twin is developed in order to suitably perform real-time damage detection and localization under the action of seismic loads. To provide a faithful virtual description of the considered framework, the digital transformation process is carried out by means of a DT comprised by three components: (*i*) a physics-based model of the structure to be monitored, either the FOM governed by Problem (1) or the corresponding reduced-order representation described by Problem (2); (*ii*) the generation of site-specific accelerograms, compatible with pseudo-real seismic loads and exploited to force the structural model; (*iii*) the extraction of relevant dofs recordings, in terms of multi-variate time series to mimic the deployed monitoring system, and the contamination of these signals by adding an independent, identically distributed Gaussian noise to allow for measurement noise.

The building to be monitored is modeled as a two-dimensional frame, see Fig. 2, adopting a plane stress formulation with an out of plane thickness of 0.1 m.

[1] Deck, Dynamic Displacement Sensor. Move Srl, Italy. `https://www.movesolutions.it/deck/`.

The structure is assumed to be made of concrete, with mechanical properties: Young's modulus $E = 30$ GPa, Poisson's ratio $\nu = 0.2$, density $\rho = 2500$ kg/m^3.

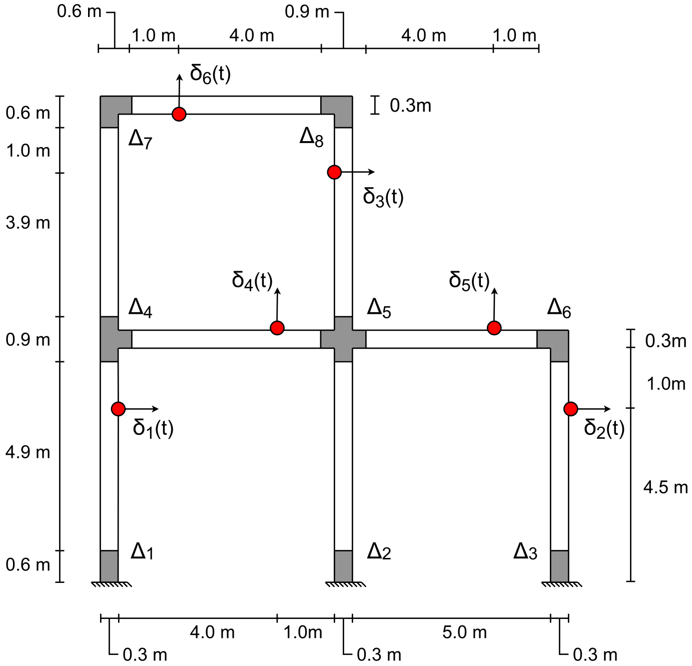


Fig. 2: Physics-based digital twin of the monitored structure, with details of the loading conditions and synthetic recordings related to displacements $\delta_1(t), \dots, \delta_6(t)$ and target damage locations associated to $\varDelta_1, \dots, \varDelta_8$.

The structure is excited by seismic loads simulated by means of ground motion prediction equations adapted from [17,18] and allowing to generate spectrum-compatible accelerograms as a function of: local magnitude $Q \in [4.6, 5.3]$, epicentral distance $R \in [80, 100]$ km, and site geology; with the previous notation used to specify the ranges in which Q and R can take value, implicitly denoting that a uniform probability distribution is adopted to describe them, while having considered a rocky condition as site geology.

Structural displacement time histories $\boldsymbol{\delta}_i = \boldsymbol{\delta}_i(\varDelta, l, \boldsymbol{\eta}) \in \mathbb{R}^L$, with $i = 1, \dots, N_s$, are reocorded in $(0, T)$ from $N_s = 6$ dofs arranged as depicted in Fig. 2. Recordings are provided for a duration ($T = 70$ s) with an acquisition frequency of $\not{f} = 25$ Hz, thus consisting of $L = 1751$ measurements each. The FOM in Eq. (1) is obtained from a FE discretization using linear tetrahedral elements and resulting in $\mathscr{M} = 4326$ dofs. The damping matrix is assembled according to the Rayleigh's model to account for a 5% damping ratio on the first four structural modes. The stiffness matrix $\mathbf{K}(\mathit{\Delta}, l)$ is parameterized to account for $N_d = 9$ damage scenarios, simulated by reducing the material stiffness within the corresponding subdomain labeled by $\mathit{\Delta} \in \{\mathit{\Delta}_0, \ldots, \mathit{\Delta}_{N_d}\}$ and highlighted in dark grey in Fig. 2, with $\mathit{\Delta}_0$ identifying the undamaged case and all the others being referred to specific damage state undergone by the structure, as described in Tab. 1. The damage level $l \in [5\%, 25\%]$, representing the amplitude of the stiffness degradation is held constant within the time instance $(0, T)$.

Table 1: Considered damage scenarios.

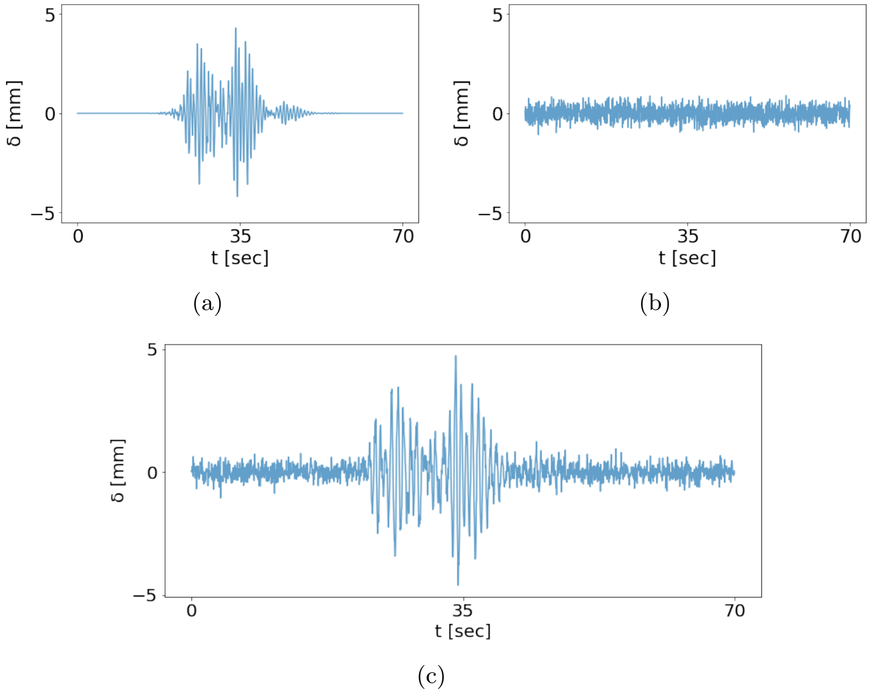
Class label	Damage Location
$\mathit{\Delta}_0$	Undamaged condition
$\mathit{\Delta}_1$	Ground floor – left
$\mathit{\Delta}_2$	Ground floor – mid
$\mathit{\Delta}_3$	Ground floor – right
$\mathit{\Delta}_4$	First floor – left
$\mathit{\Delta}_5$	First floor – mid
$\mathit{\Delta}_6$	First floor – right
$\mathit{\Delta}_7$	Roof – left
$\mathit{\Delta}_8$	Roof – mid

The projection basis $\mathbf{W}$ ruling the ROM in Eq. (2) is instead computed from a snapshot matrix $\mathbf{S}$ comprising $\mathscr{S} = 630,360$ snapshots, obtained through 360 evaluations of the FOM for different values of the input parameters sampled via latin hypercube rule. By prescribing a tolerance $\epsilon = 10^{-4}$, the order of the ROM is set to $\mathscr{M}_{\mathtt{R}} = 31$, in place of the original $\mathscr{M} = 4326$ dofs. The population of the dataset $\mathbf{D} \in \mathbb{R}^{L \times N_s \times N_o}$ is carried out by systematically evaluating the ROM for $N_o = 9999$ instances at varying input parameters values.

In this work, the signals are corrupted by assuming an additive Gaussian noise uncorrelated in time, to represent measurement noise and those environmental and ambient components potentially affecting the structural response, such as traffic, temperature, humidity, rain, wind [19]. As typically done in signal processing, the amount of meaningful information carried by a signal with respect to the amount of noise components is measured by adopting a signal-to-noise ratio (SNR). The SNR of a generic signal $\boldsymbol{\delta}^S$ is defined as the ratio between the power $\mathcal{P}_{\text{signal}}$ of the signal itself over the power $\mathcal{P}_{\text{noise}}$ of the relevant noise components $\boldsymbol{\delta}^N$, in logarithmic decibel scale as follows:

$$\mathrm{SNR} = 10 \log_{10}\left(\frac{\mathcal{P}_{\text{signal}}}{\mathcal{P}_{\text{noise}}}\right) = 10 \log_{10}\left(\frac{\mathbb{E}[(\boldsymbol{\delta}^S)^2]}{\mathbb{E}[(\boldsymbol{\delta}^N)^2]}\right) \ , \tag{10}$$

where $\mathbb{E}[\,\cdot\,]$ denotes the expectation operator. A SNR higher than 0 dB denotes more information than noise, while a ratio equal to infinity indicates the absence of noisy components. In this work, an independent, identically distributed Gaussian noise, yielding a SNR $= 10$ dB is adopted to corrupt both the training and testing data. An exemplary $\delta_1(t)$ displacement time history is reported in Fig. 3a, a as obtained from a FOM simulation for a sample seismic event; in Fig. 3b, is reported an instance of the noise components mentioned above, while resulting noisy recordings is reported Fig. 3c. Before training the DL model, the data are preprocessed sensor-by-senor by standardizing all the data, so that the entire amount of data gathered bu the same sensor are normalized to feature zero mean and unit variance. Moreover, the hold-out method is used to split dataset $\mathbf{D}$ into training sets and test sets, respectively amounting to 90% and 10% of the data, while the validation set is obtained by taking 20% of the data in the training set.



(a)

(b)

(c)

Fig. 3: Exemplary $\delta_1(t)$ displacement time histories: (a) instance of recorded signal, obtained from a FOM simulation for a sample seismic event; (b) sample of independent, identically distributed Gaussian noise affecting the sensor; (c) resulting noisy recordings. Extracted from [12].

4 Experiments and results

The overall methodology is developed on `Google Colab` [20], a free platform based on the open-source `Jupyter` project and featuring an `NVIDIA Tesla K80` GPU card. In Fig. 4, is illustrated a high level flowchart of the methodology. Both the adopted dataset and code are publicly released and made available at [21].

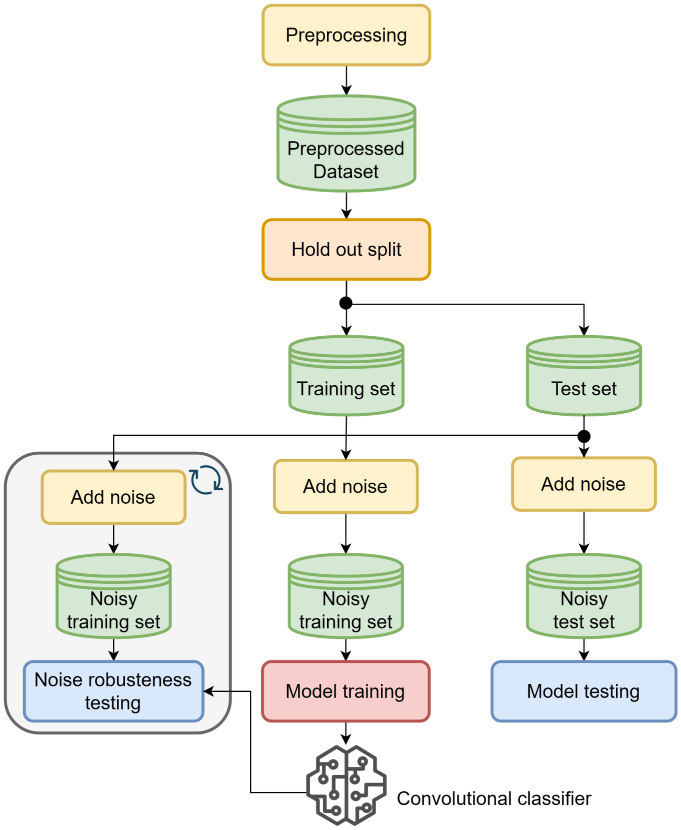


Fig. 4: High level workflow of experiments.

In the following, we detail the adopted CNN, which is designed to automatically and adaptively learn feature hierarchies through backpropagation, using multiple building blocks such as convolution layers, pooling layers, and fully connected layers. Then, an analysis of the performance of the DL model is presented, also considering the beneficial effect of exploiting a hyperparameter optimization strategy for DL models. Tab. 2 summarizes the numerosity of the target classes for the training and test sets.

Table 2: Classes numerosity in the training and test sets.

Damage class	Training set	Test set
Δ_0	994	117
Δ_1	1003	108
Δ_2	1008	103
Δ_3	998	113
Δ_4	992	119
Δ_5	1001	110
Δ_6	998	113
Δ_7	1005	106
Δ_8	1001	110

4.1 Damage localization via CNN

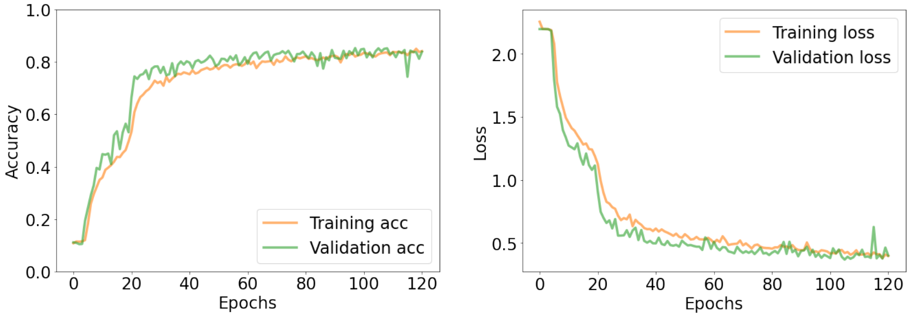
The architecture adopted to address the damage localization task, along with its relevant hyperparameters are summarised in Tab. 3; specifically, it is a convolutional model made of 4 blocks. The first three deal with feature extraction, whereas the last one performs the classification task. Each of the first three blocks consists of a 1D convolutional layer, a 1D max pooling layer, and a dropout layer. The output features are then reshaped through a flatten layer and run through the classifier block, which is composed of two dense layers and a dropout one.

Table 3: Adopted CNN architecture and selected hyperparameters.

Layer type	Hyperparameter	Output shape	# parameters
Input	-	[1751, 6]	0
Conv-1D	kernels=6, kernel size=32	[1751, 6]	1158
MaxPool-1D	pooling size=8	[219, 6]	0
Dropout	rate=0.15	[219, 6]	0
Conv-1D	kernels=32, kernel size=20	[219, 32]	3872
MaxPool-1D	pooling size=6	[37, 16]	0
Dropout	rate=0.15	[37, 16]	0
Conv-1D	kernels=16, kernel size=12	[37, 16]	6160
MaxPool-1D	pooling size=4	[10, 16]	0
Dropout	rate=0.15	[10, 16]	0
Flatten	-	[160]	0
Dense	units=64	[64]	10304
Dropout	rate=0.15	[10, 16]	0
Dense (output)	units=9	[9]	585

Adopting the Xavier's weight initialization, the loss function is minimized using the Adam algorithm, a first-order stochastic gradient descent optimizer, for a maximum of 200 allowed epochs. To ensure that the CNN does not learn the training dataset, but a possible model behind it, an early-stopping strategy is used to interrupt the learning process, whenever overfitting shows up. Whenever the loss function value computed on the validation set does not decrease for 10 epochs in a row, this latter terminates the CNN training before reaching the number of allowed epochs.

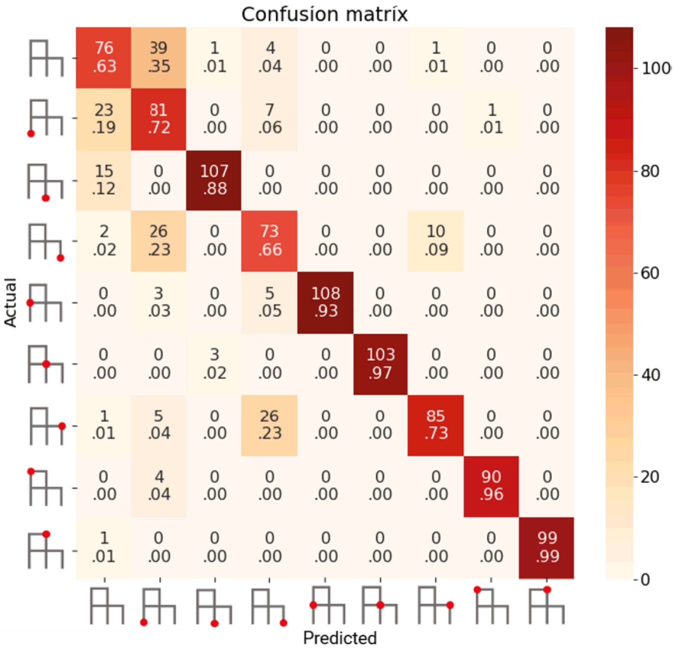
The evolution of the loss function and of the accuracy metric over the training and validation sets, obtained while training the classifier, is reported in Fig 5a and Fig 5b, respectively. The training process ends after 121 epochs due to the early stopping condition, after which the value of the tunable parameters yielding the best perfomance in terms of loss function are restored. From both graphs, it is clear that most of the gains deriving from tuning the CNN parameters are attained during the first portion of the training. The slightly irregular trend is due to the presence of dropout layers and to the stochastic nature of the minimization algorithm, for which different values of loss and classification accuracy are obtained on different mini-batches. After completing the training, the CNN achieves a global accuracy of 83%. The relevant results are gathered in the confusion matrix of Fig. 6, and in Tab. 4, which reports the precision and recall values class by class. We can observe that the lowest values of the assessment metrics are obtained for labels $\varDelta_1$, $\varDelta_2$ and $\varDelta_3$, suggesting that in some scenarios it may not be easy to distinguish the presence of damage on the ground floor with the undamaged state of the structure. Regarding the upper floor damage, experimental results indicate that the convolutional architecture is able to localize it with a high accuracy level.



(a)

(b)

Fig. 5: Classifier training: loss function (a) and accuracy metric (b) evolution on the training and validation sets. Extracted from [12].



76 .63	39 .35	1 .01	4 .04	0 .00	0 .00	1 .01	0 .00	0 .00
23 .19	81 .72	0 .00	7 .06	0 .00	0 .00	0 .00	1 .01	0 .00
15 .12	0 .00	107 .88	0 .00	0 .00	0 .00	0 .00	0 .00	0 .00
2 .02	26 .23	0 .00	73 .66	0 .00	0 .00	10 .09	0 .00	0 .00
0 .00	3 .03	0 .00	5 .05	108 .93	0 .00	0 .00	0 .00	0 .00
0 .00	0 .00	3 .02	0 .00	0 .00	103 .97	0 .00	0 .00	0 .00
1 .01	5 .04	0 .00	26 .23	0 .00	0 .00	85 .73	0 .00	0 .00
0 .00	4 .04	0 .00	0 .00	0 .00	0 .00	0 .00	90 .96	0 .00
1 .01	0 .00	0 .00	0 .00	0 .00	0 .00	0 .00	0 .00	99 .99

Fig. 6: Confusion matrix on test set. Extracted from [12].

Table 4: Damage localization Precision and Recall by class on test set.

Damage class	Precision	Recall
Δ_0	.56	.59
Δ_1	.54	.61
Δ_2	.95	.88
Δ_3	.61	.71
Δ_4	.94	.92
Δ_5	.99	.97
Δ_6	.91	.78
Δ_7	1.0	.98
Δ_8	1.0	1.0

4.2 Noise tolerance evaluation

Since the amplitude of ambient-induced vibrations is often not known a priori as it can be due to a multitude of causes, in the following the performance of the CNN model are assessed considering multiple test sets featuring a different noise level; this is useful to provide some insights on the tolerance of the trained model against noise. Overall, the model is evaluated on 13 different test sets, each collecting recordings that feature a different SNR value between 1 dB and 25 dB.

The result of this analysis is reported in Fig. 7, which shows the performance of the CNN in terms of classification accuracy while varying the value of SNR. From this latter, it can be observed that the convolutional model is still able to detect damage-sensitive patterns, even when the recordings are contaminated with a very high level of noise. Specifically, when considering test sets featuring a SNR value higher than that characterizing the training data (equal to 10 dB), an improvement in the CNN performance is observed; on the other hand, when values of SNR lower than 10 DB are adopted, the CNN performance gets worse, however the attained classification accuracy attests on values largely above that one corresponding to the random guess, and still fairly remarkable for SNR values above than 5 dB.

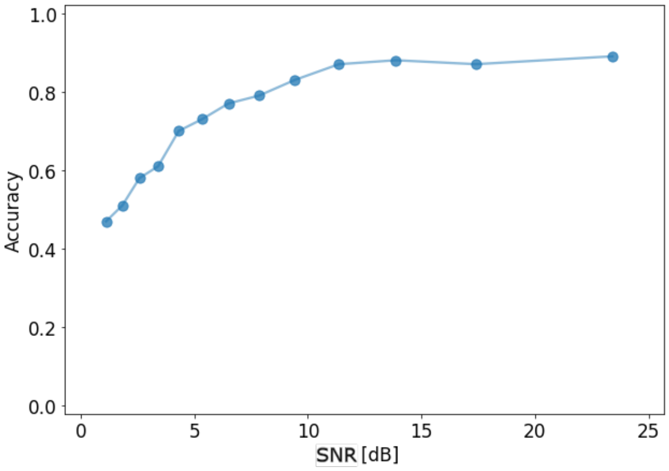


Fig. 7: Model accuracy on test set varying the EC values. Extracted from [12].

4.3 Random search algorithm for hyperparameter optimization

The design of deep neural network architectures requires to choose several parameters that are not learned during the training process but need to be selected by the user. These are the so called hyperparameters, which includes the network topology, the width/depth of each layer and the training options controlling the optimization algorithm. The performance of DL architectures critically depends on the specific choice of hyperparameters and, often, finding an optimal combination of them can make the difference between good and bad models. Therefore, to improve the performance of a DL model, a hyperparameter tuning can be carried out to find a set of optimal hyperparameters, for instance see [22]. When such a hyperparameter tuning is carried out via random search, several feasible hyperparameters configurations are iteratively instantiated to train the corresponding DL model, then the results obtained by means of each trained model are compared to determine the best set of parameters [23].

Since the DL model described in the previous section and detailed in Tab. 3 is defined through a simple trial-and-error heuristic, a random search optimization algorithm is here adopted to fine tune its relevant hyperprameters.

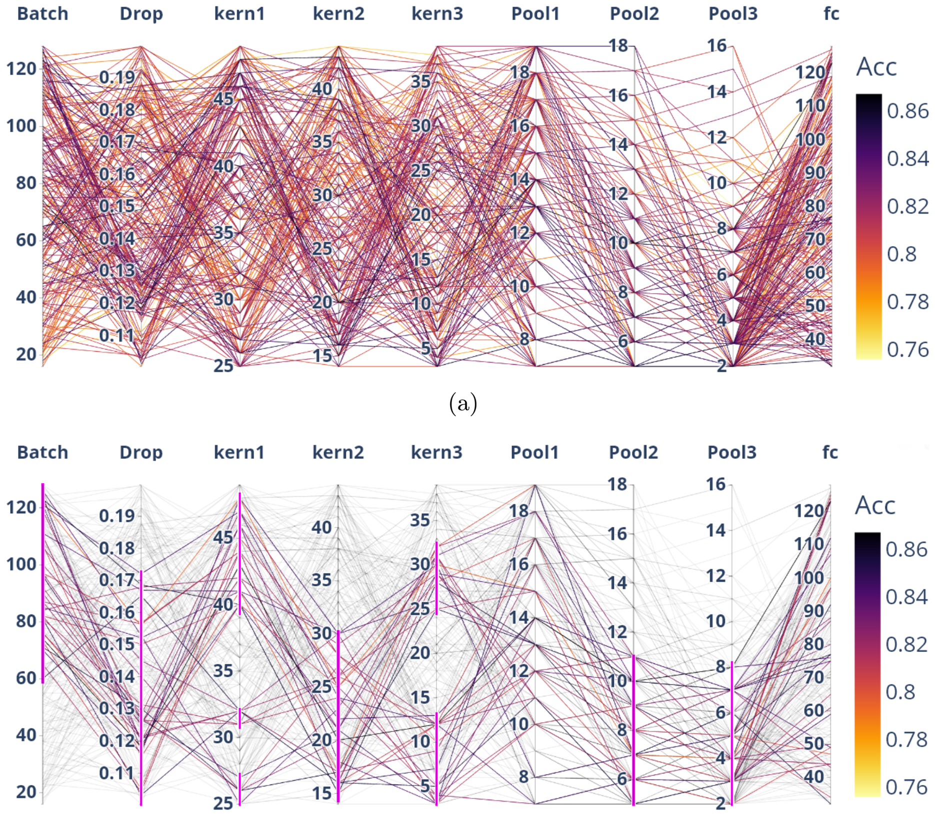
Table 5: Hyperparameter optimization: original and optimal values.

Name	Old value	Optimal value	Suitable range
Batch	128	60	[60; 128]
Drop	.15	.165	[.11; .17]
Kern-1	32	32	[25;27], [31;32], [40;48]
Kern-2	20	16	[15; 30]
Kern-3	16	5	[25; 32], [4; 13]
Pool-1	8	11	[7; 18]
POol-2	6	5	[5; 11]
Pool-3	4	3	[2; 8]
fc	64	92	[32; 128]
Acc	.83	.85	

Fig. 8a shows the iterations of the random search optimization algorithm through a parallel coordinates chart; at each iteration the relevant hyperparameters are sampled in the corresponding ranges reported in the same figure, with each parameter described by a uniform probability distribution. Overall, we obtain accuracy values between 76% and 85%. The hyperparameter ranges yielding the best results are highlighted in Fig. 8b and also summarized in Tab. 5 along with their optimal value. Adopting the optimal hyperparameters the classification accuracy over the considered damage localization task increases from the previous value of 83% up to 85%.

Fig. 9 shows the confusion matrix computed after the grid search hyperparameter optimization. We can observe the accuracy value improved for some classes in comparison to the non-optimized model. Specifically, the ground and 1^{st}-floor classes demonstrated an improvement in accuracy, whereas 2^{nd}-floor classes showed marginal or no change. Such classes count the highest number of misclassifications compared to the respective classes shown in the confusion matrix shown in Fig. 6.

Despite the partial improvement across classes, the optimization process improved the overall performance of the model, indicating the importance of hyperparameter tuning in machine learning algorithms.



(a)

(b)

Fig. 8: Hyperparameter optimization:(a) iterations of the random search optimization algorithm; (b) identified value ranges yielding an improved performance.

5 Conclusion

Within a structural health monitoring framework, this work has proposed a comprehensive methodology for structural damage localization based on convolutional neural networks and a digital twins. The digital twin paradigm has been introduced to meet the need of labeled data to train deep learning models while adopting a supervised learning approach. The digital twin design has been carried out through a physics-based numerical model useful to represent the physical asset in a virtual space. A reduced-order modeling strategy is also adopted to speed up the entire procedure and allow to move toward real-time applications. Finally, a classifier based on a convolutional neural network has been adopted to perform automatic feature extraction and to relate raw sensor

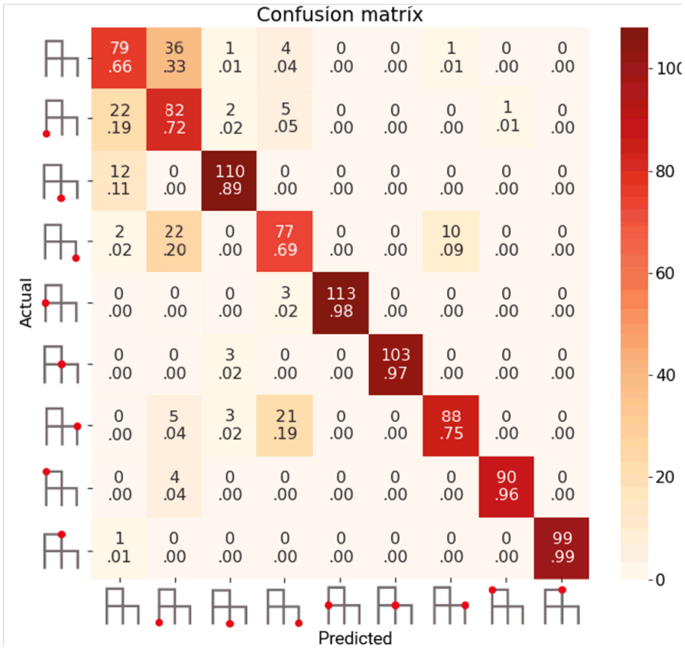


Fig. 9: Confusion matrix on test set after hyperparameter optimization

data to the corresponding structural health conditions. The latter has been fine tuned using the random search hyperparameter optimization algorithm.

The proposed strategy has been assessed on the monitoring of a two-story portal frame subjected to the action of seismic events. The damage localization task has been carried out with a remarkable accuracy and the method has shown to be insensitive to ambient-induced and measurement noise and to the varying operational conditions, characterized by seismic events of different nature.

This work represents a preliminary effort to demonstrate the capabilities of the proposed strategy. Beside the need of a further validation within a suitable experimental setting, the next studies will also take into account the eventuality of buildings simultaneously suffering multiple damaged zones. Moreover, a comparison with alternative deep learning architectures, such as the long-short term memory model and the transformer model, should be envisaged.

Acknowledgements

This work has been partially supported by: (i) the University of Pisa, in the framework of the PRA_2022_101 project “Decision Support Systems for territorial networks for managing ecosystem services”; (ii) the Tuscany Region, in the framework of the "SecureB2C" project, POR FESR 2014-2020, Law Decree 7429

31.05.2017; (iii) the Italian Ministry of Education and Research (MIUR), in the framework of the FoReLab project (Departments of Excellence). The authors are grateful to the research team at Politecnico di Milano composed of Alberto Corigliano, Andrea Manzoni, Luca Rosafalco and Stefano Mariani, for several insightful discussions about this research.

References

1. Ye, X., Jin, T., Yun, C.: A review on deep learning-based structural health monitoring of civil infrastructures. Smart Struct. Syst **24**(5), 567–585 (2019)
2. Torzoni, M., Rosafalco, L., Manzoni, A., Mariani, S., Corigliano, A.: SHM under varying environmental conditions: an approach based on model order reduction and deep learning. Comput Struct **266**, 106790 (2022). https://doi.org/https://doi.org/10.1016/j.compstruc.2022.106790
3. Toh, G., Park, J.: Review of vibration-based structural health monitoring using deep learning. Applied Sciences **10**(5), 1680 (2020)
4. Torzoni, M., Manzoni, A., Mariani, S.: Structural health monitoring of civil structures: a diagnostic framework powered by deep metric learning. Comput Struct **271**, 106858 (2022). https://doi.org/10.1016/j.compstruc.2022.106858
5. Torzoni, M., Manzoni, A., Mariani, S.: A deep neural network, multi-fidelity surrogate model approach for bayesian model updating in shm. In: European Workshop on Structural Health Monitoring, pp. 1076–1086. Springer International Publishing (2023). https://doi.org/10.1007/978-3-031-07258-1_108
6. Wang, X., et al.: Probabilistic machine learning and bayesian inference for vibration-based structural damage identification (2022)
7. Cimino., M., Galatolo., F., Parola., M., Perilli., N., Squeglia., N.: Deep learning of structural changes in historical buildings: the case study of the pisa tower. In: Proceedings of the 14th International Joint Conference on Computational Intelligence. pp. 396–403. INSTICC, SciTePress (2022)
8. Aydemir, H., Zengin, U., Durak, U.: The digital twin paradigm for aircraft review and outlook. In: AIAA Scitech 2020 Forum. p. 0553 (2020)
9. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the digital twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology **29**, 36–52 (2020)
10. Rosafalco, L., Torzoni, M., Manzoni, A., Mariani, S., Corigliano, A.: Online structural health monitoring by model order reduction and deep learning algorithms. Computers & Structures **255**, 106604 (2021)
11. Quarteroni, A., Manzoni, A., Negri, F.: Reduced basis methods for partial differential equations: an introduction, vol. 92. Springer (2015)
12. Parola., M., Galatolo., F., Torzoni., M., Cimino., M., Vaglini., G.: Structural damage localization via deep learning and iot enabled digital twin. In: Proceedings of the 3rd International Conference on Deep Learning Theory and Applications - DeLTA,. pp. 199–206. INSTICC, SciTePress (2022). https://doi.org/10.5220/0011320600003277
13. Torzoni, M., Rosafalco, L., Manzoni, A.: A combined model-order reduction and deep learning approach for structural health monitoring under varying operational and environmental conditions. Engineering Proceedings **2**(1), 94 (2020)
14. Galatolo, F.A., Cimino, M.G.C., Vaglini, G.: Using stigmergy to incorporate the time into artificial neural networks. In: International Conference on Mining Intelligence and Knowledge Exploration. pp. 248–258. Springer (2018)

15. Li, D., Zhang, J., Zhang, Q., Wei, X.: Classification of ecg signals based on 1d convolution neural network. In: 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom). pp. 1–6. IEEE (2017)
16. Yiğitler, H., Badihi, B., Jäntti, R.: Overview of time synchronization for iot deployments: Clock discipline algorithms and protocols. Sensors **20**(20), 5928 (2020)
17. Paolucci, R., Gatti, F., Infantino, M., Smerzini, C., Özcebe, A.G., Stupazzini, M.: Broadband ground motions from 3d physics-based numerical simulations using artificial neural networksbroadband ground motions from 3d pbss using anns. Bulletin of the Seismological Society of America **108**(3A), 1272–1286 (2018)
18. Sabetta, F., Pugliese, A.: Estimation of response spectra and simulation of non-stationary earthquake ground motions. Bulletin of the Seismological Society of America **86**(2), 337–352 (1996)
19. Aparicio, J., Jiménez, A., Ureña, J., Alvarez, F.J.: Realistic modeling of underwater ambient noise and its influence on spread-spectrum signals. In: OCEANS 2015-Genova. pp. 1–6. IEEE (2015)
20. Bisong, E.: Building machine learning and deep learning models on Google cloud platform. Springer (2019)
21. Parola, M.: Damage localization task source code and data, `https:\/\/github.com\/MarcoParola\/structural_health_monitoring`
22. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of machine learning research **13**(2) (2012)
23. Yu, T., Zhu, H.: Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689 (2020)