1st LATAM-SHM 2023 Conference
A. Güemes, L. Mujica, J. Sierra (Eds)

DAMAGE CLASSIFICATION UTILIZING AUTOENCODERS AND CONVOLUTIONAL NEURAL NETWORKS

FLÁVIO S. BARBOSA*, LUCAS V. RESENDE*, RAFAELLE P. FINOTTI*, ALEXANDRE A. CURY*, CÁSSIO C. MOTTA*, HERNÁN GARRIDO†, MARTÍN DOMIZIO† AND OSCAR CURADELLI†

* Programa de Pós-Graduação em Engenharia Civil (PEC)
Universidade Federal de Juiz de Fora
Faculdade de Engenharia, Campus Universitário, 36036-900 Juiz de Fora, Brazil
e-mail: flavio.barbosa@ufjf.br, http://www.ufjf.br/pec

† Facultad de Ingeniería
Universidad Nacional de Cuyo - CONICET
Centro Universitario, M5502JMA Mendoza, Argentina
e-mail: carloshernangarrido@gmail.com, https://www.uncuyo.edu.ar

Key words: Structural Health Monitoring, Autoencoder, Convolutional Neural Network, Image Processing.

Abstract. Structural Health Monitoring (SHM) is a growing field in civil engineering and has relevance for detecting changes in the state of structures, including identifying damage conditions. SHM strategies commonly employ Artificial Intelligence (AI) techniques on raw dynamic data measured from structures to perform classifications or extract features from the original data. Among the AI algorithms for SHM, autoencoder, and convolutional neural networks have been identified as promising solutions, being the focus of this article. Both algorithms are applied to identify eight damage scenarios in a beam starting from the time histories of the tested structure, pointing out the advantages and disadvantages of each algorithm. The authors tested the beam through monitoring based on image processing using a high-speed camera. By comparing the results obtained from both algorithms, the researchers were able to highlight their respective strengths and weaknesses in the context of SHM. This information can assist engineers and researchers in selecting the most appropriate algorithm based on the specific requirements of their monitoring project.

1 INTRODUCTION

Classically, monitoring the state of conservation of structures is done by analyzing the evolution of their modal characteristics [1]. This strategy was adopted in the first works that dealt with this theme and is still relevant today. However, environmental and operational factors negatively affect this strategy, causing changes in the natural frequencies of an evaluated structure to be incorrectly associated with structural damage. Therefore, several Structural Health Monitoring (SHM) techniques have been developed, seeking to consider the environmental effects in damage detection algorithms.

The most promising seems to be those based on Artificial Intelligence (AI) [2,3]. In these techniques, displacement or acceleration time histories collected from structures at different situations are presented to algorithms that can "learn" if an alteration in the mechanical behavior is detected due to operating conditions or consequences of structural damage. Among the most promising AI algorithms that have received special attention from researchers, Sparse Autoencoders (SAE) [4,5], and Convolutional Neural Networks (CNN) [6,7] stand out. These algorithms have been successfully used in several recent works that address SHM [8,9].

In this context, this article compares the performance of autoencoders and CNNs applied to identify eight damage scenarios in a beam tested by the authors, pointing out the strengths and weaknesses verified in each AI technique. The paper's experimental program includes a methodology that provided good results for the displacements measured simultaneously at 60 points on the beam, of which 40 monitored points were used to evaluate the tested AI algorithms.

2 THEORETICAL BACKGROUND

2.1 Sparse autoencoders - SAE

An autoencoding neural network [10], or just autoencoder, is a type of artificial neural network designed to reproduce its inputs as closely as possible after unsupervised training. An autoencoder is made up of two parts: an encoder, which receives the input data $\mathbf{x}$ and transforms it into a lower dimensional code $\mathbf{h}$; and a decoder, responsible for decoding $\mathbf{h}$ into an output $\mathbf{x'}$ that closely matches $\mathbf{x}$. In other words, autoencoders attempt to reconstruct their inputs as outputs while generating a vector $\mathbf{h}$ that holds much of the information contained within the training data in fewer dimensions. Therefore, autoencoders are useful tools for extracting features that represent the original data with lower dimensionality, which facilitates classification through other methods. Figure 1 shows the structure of the autoencoder applied in the present work: The input layer is composed of vectors $\mathbf{x}$ containing 2500 time steps of displacements; the hidden layer is formed by vectors $\mathbf{h}$ with 15 or 25 elements (n=15 or 25), depending on the analyzed case, used in the classification process (damage scenario identification).

A Sparse Autoencoder (SAE) is a type of autoencoding neural network that includes a sparsity penalty term at its training function. It is particularly useful to represent large datasets with a small number of $\mathbf{h}$ components. The function that calculates the error between the input and the output is minimized in the training process (see reference [10]). In the present work, the chosen cost function for the SAE was the mean squared error with sparsity regularizers, and the training algorithm was the scaled conjugate gradient backpropagation. There are a few hyperparameters that must be optimized for SAE models to be able to represent the training data while still being able to generalize. In this work, those hyperparameters are the Sparsity Proportion, the L2 Weight Regularization, and the Sparsity Regularization. The optimization was performed empirically through a grid search process, as described in [9]. Table 1 presents general characteristics of the SAE applied in this work.

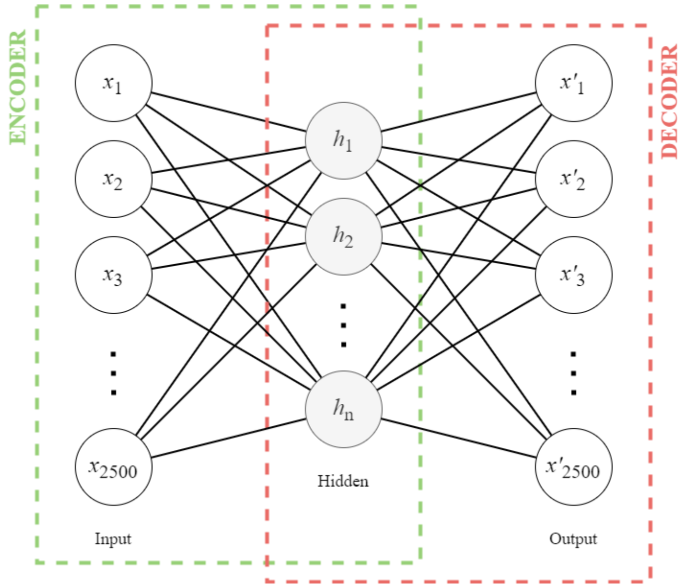

Figure 1: Basic structure of the autoencoder neural network used in this work

Table 1: Characteristics of the SAE model

Activation function - encoder	Logarithmic sigmoid
Activation function - decoder	Linear
Training algorithm	Scaled conjugate gradient
Sparsity ratio	0.4
Regularization term	0.001
Sparsity control term	4
Maximum number of epochs	1000

In this work, the vectors **h** with dimension n=15 or n=25, resulting from the application of SAE to the data coming from the structural instrumentation, are used as input data of a Multi-Layer Perceptron (MLP) network for supervised classification of each tested structural damage scenario. From this point in the text, the MPL mentioned above will be referred to as the shallow network.

Autoencoders are usually used as unsupervised classifiers. However, the objective here was to evaluate their capacity as dimensionality reducers, leaving the classification of input data to be performed by an MLP network, in a more comparable way as to what is done with the CNNs, which facilitates performance comparison between the two analyzed techniques.

2.2 Shallow neural network

The shallow neural network architecture used in this work to classify the monitoring data treated through SAE algorithm is shown in Figure 2. For each monitored structure signal (**x**), the output of the shallow network provides a vector (**y**) that is associated with each of the eight experimentally tested structural damage scenarios. For example, for **y**={1 0 0 0 0 0 0 0}, it means that the first damage scenario has been identified. Table 2 presents general characteristics of the shallow network applied in this work.

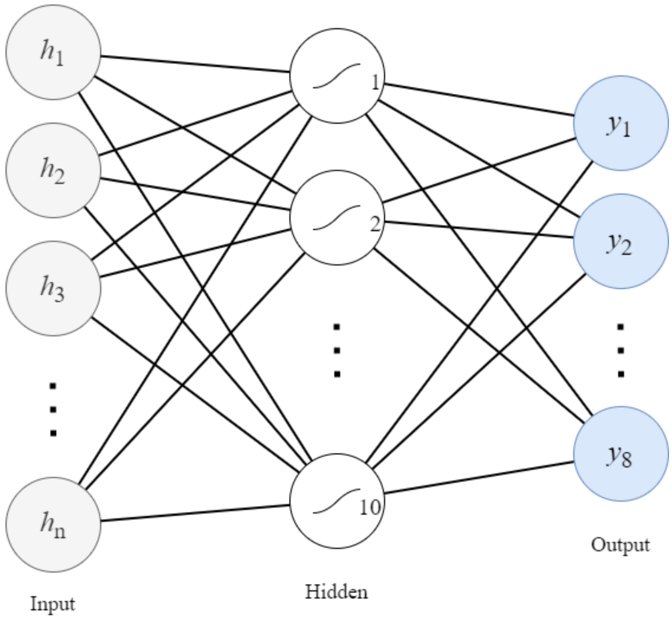

Figure 2: Structure of the shallow network

Table 2: Characteristics of the shallow network used in this work.

Activation function – hidden layer	Logarithmic sigmoid
Activation function – output layer	Softmax
Training algorithm	Scaled conjugate gradient
Number of neurons in hidden layer	10
Maximum number of epochs	1000

2.3 Convolutional neural network - CNN

CNNs [11] are biologically inspired neural networks widely used and have become the standard in many systems for recognizing objects and events in an image or video. Although almost 30 years have passed since the first CNN was proposed, its modern architectures still share common features with the first, such as convolutional and clustering layers. Furthermore, the most used training method is still the backpropagation technique [12].

The popularity and wide range of application domains of deep CNNs can be attributed to the following advantages:

- CNN models can merge the processes of parameter extraction and feature classification into a single body of learning;
- Since CNN neurons are sparsely connected, they can process large inputs with high

computational efficiency compared to conventional fully connected Multi-Layer Perceptron (MLP) networks;

- CNNs are invariant to minor transformations on the input data, including translation, scaling, skewing, and skew;
- CNN can adapt to different input sizes

Classically applied to image data (two-dimensional), the use of CNN in problems where the inputs have only one dimension (time series) can be interpreted as applying the technique to an image with a width of one pixel.

In this paper, the architectures of the CNNs used are shown in Figure 3. The input layer has 2500 neurons with the respective 2500 time steps of each monitored response. The next layer is a convolution layer, where 10 convolution filters, with 50 positions, are used. Then, batch normalization and ReLU[1] activation function are applied to the convolution layer. The padding technique is used here to keep the ReLU's outputs the same size as the inputs. The next layer is a pooling layer: clusters with 100 and 125 positions were evaluated and the highest values (maxpool) were adopted as the result of each grouping. It leads to 25 or 20 (respectively associated to p=100 or p=250 positions of each cluster) in the pooling layer. Softmax activation function is finally applied to the outputs of pooling layer (fully connected) leading to classification vector (**y**) associated with damage scenarios (identical to the output of the shallow network).

The CNNs are trained in a supervised way via the backpropagation algorithm [12]. During each iteration, the gradient magnitude (or sensitivity) of each network parameter, such as the weights of convolution layers and fully connected layers, is calculated. The parameter sensitivities are then used to iteratively update the CNN parameters until a given stopping criterion is reached. There are several optimization methods that can be used to calculate the CNN parameters, similarly to what happens with the SAE. In this work, the Adam Stochastic Optimization algorithm was applied [13].

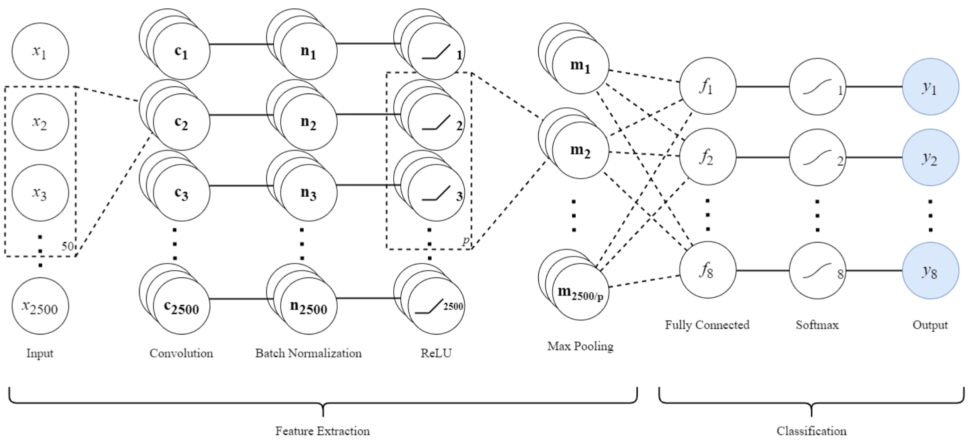

Figure 3: Architecture of the CNN adopted in this paper

[1] A rectified linear unit - ReLU is an activation function that introduces the property of non-linearity to a deep learning model and solves the vanishing gradients issue.

3 EXPERIMENTAL PROGRAM

The methodology is demonstrated through an experimental example which includes some laboratory-conditions and some field-conditions. Figure 4 displays a photograph of the experimental setup. The healthy specimen under test was new, and the damage was artificially induced. This ensures the damage scenarios can be accurately identified to provide a good reference for assessing the performance of the methodology. Natural daylight was used to illuminate the scene, and the background was not specially prepared; resembling imperfect situations that are common in field test programs.

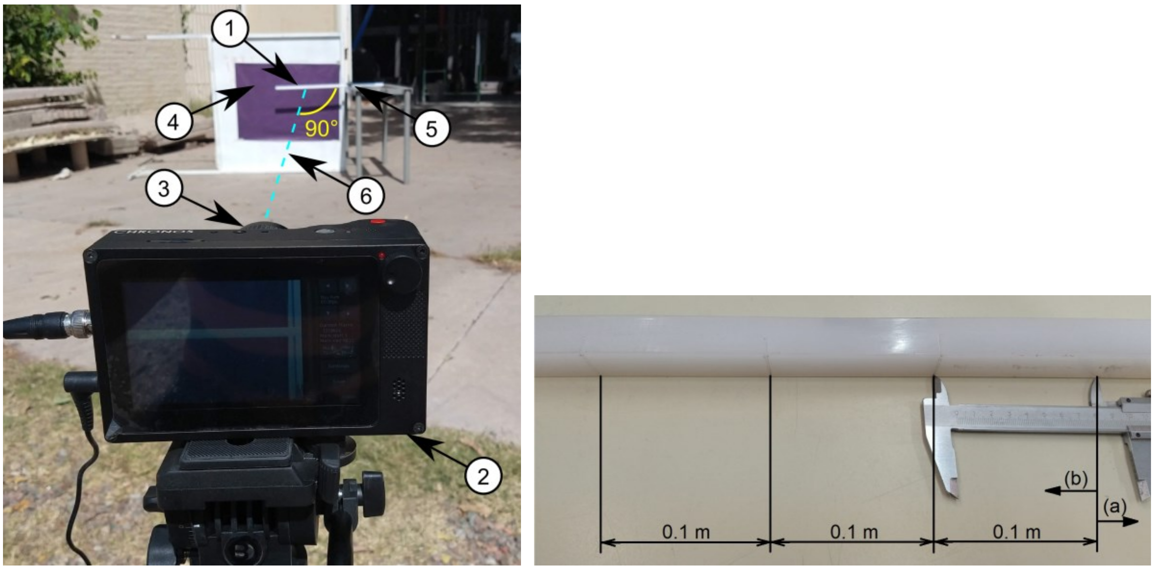

Figure 4: Experimental setup. On the left: (1) beam (specimen), (2) high-speed camera, (3) lens, (4) background, (5) fixed support, (6) line of sight. On the right: Zoom of the specimen in the damaged region. References: (a) clamped zone, (b) free span

The beam specimen under test, (1) in Figure 4, consisted of a 1 m long straight beam made of polypropylene with a 25.4x25.4 mm squared cross-section. The beam was free in the left end and fixed at the right end. The resulting free span was 0.8 m long and the first natural frequency in the healthy state was 7.69 Hz.

The beam was artificially damaged by producing vertical and rectangular Mode I (opening) cracks on its upper face using a steel saw. The actual widths of the cracks were between 0.5 and 1 mm. As a combination of crack locations and depths, eight successive damage scenarios were defined, all of which are summarized in Table 3. Cracks were produced at 100, 200 and 300 mm from the fixed end of the beam.

Two videos were recorded for each damage scenario (16 runs in total) with a Chronos 2.1-HD high-speed camera, (2) in Fig. 4, at a frame rate of 1000 fps, with an exposure time of 100% (1/1000 s), and a resolution of 1280 columns by 1024 rows. In this work, only 1200 columns where in the region of interest for measurement. Columns were grouped into 60 horizontal windows (of 20 columns each), one for each space sample point. The optical setup resulted in a pixel scale of ≈ 0.6 mm/px.

The video records were taken during free vibration, after imposing the following initial conditions: a displacement of 0.05 m at the beam free end, with a velocity of 0 m/s in the whole length.

Table 3: Damage scenarios

Scenario name	Depth (mm) at		
	100 mm	200 mm	300 mm
Scenario #0 (no damage)	0	0	0
Scenario #1	5	0	0
Scenario #2	10	0	0
Scenario #3	13	0	0
Scenario #4	13	5	0
Scenario #5	13	5	5
Scenario #6	13	10	5
Scenario #7	13	10	11

4 SIGNAL PROCESSING

For extracting displacements from video, in the particular case of a straight beam, an axial edge is a good natural feature if the background is uniform and of a different color/intensity than the beam [14]. An additional requirement is that the face corresponding to the selected edge is always hidden from the camera perspective. In this work, the method described in [14] for a single image, which uses the beam edge as natural feature, was applied to each video frame to obtain displacement records. The main steps of the method are detailed below for completeness of the present manuscript.

1. For each video frame:
    a. The image is converted to grey-scale by combining the three 8-bit color channels into a single double-precision matrix of intensity values.
    b. The horizontal edge is detected using the smooth gaussian gradient calculated along the vertical direction with a 15 px long vertical gaussian window of ±3 standard deviations.
    c. image gradient is binarized using a threshold and, then, the largest connected component is identified using the flood-fill algorithm. This component is a region of the image that will be used as a mask.
    d. For each horizontal window:
        i. For each column:
            A. Where the mask is active: Sub-pixel resolution gradient peaks are estimated and located by using cubic splines.
        ii. A linear function is fitted using the peak locations as sample points and their amplitudes as weights in a weighted least squares linear fitting.[2]

[2] Weighted fitting is advantageous because image columns where, accidentally, the beam and the background are of similar intensities are penalized. On the contrary, zones of high contrast are emphasized.

iii. The vertical positions are estimated as the y-intercepts of each fitted straight line (assuming the y-axis is at the middle of each horizontal window).

2. Time history records of vertical positions are low-pass filtered in time, using a first order Butterworth filter with a cut-off frequency of approximately two times the first natural frequency of the beam.
3. Transverse displacements are finally estimated as the differences between filtered positions and the mean filtered positions over the whole video duration.

To illustrate the data obtained with this signal processing method, Figure 5 is presented.

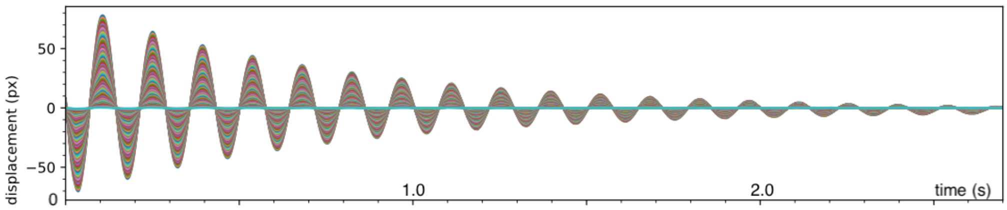

Figure 5: Typical time history of displacements obtained from video records

5 RESULTS

Two configurations of SAE (15 and 25 neurons in the hidden layer) and two configurations of CNN (p=250 and p=100 elements to construct the pooling layer) were tested in this paper. The ability of each neural network to distinguish data from each structural damage scenario was evaluated. Considering the eight damage scenarios, with displacements measured at 40 points in 2 test series, there are 640 signals (80 signals for each damage scenario). In each analysis, the respective confusion matrix obtained for a 5-fold cross-validation was calculated. The results are presented in Figure 6 and 7, for SAE and CNN, respectively.

Figure 6: Confusion matrices obtained by SAE. On the left for 15 hidden layer neurons. On the right for 25 hidden layer neurons

Figure 7: Confusion matrices obtained by CNN. On the left for 250 elements to construct the pooling layer. On the right 100 elements to construct the pooling layer

Upon analyzing these figures, it can be observed that the two algorithms utilized in the study yielded a 100% accuracy rate in one of the analyses, while all other results demonstrated hit rates greater than 70%. It is important to note that, in terms of processing time, the CNN algorithm achieved 100% hit efficiency in approximately half of the time required by the SAE algorithm, using the parameters that were adopted.

It is pertinent to acknowledge that the selection of various parameters for each algorithm directly impacts the results obtained. Nevertheless, it can be concluded that both techniques are highly efficient in addressing problems related to Structural Health Monitoring (SHM), once the algorithms are properly calibrated.

6 CONCLUSIONS

This article presents comparisons between the performance of AI algorithms - SAE and CNN - in identifying eight controlled damage scenarios. The analyzed structure was a beam monitored through image processing. The results obtained showed that, for the tests carried out, SAEs required a longer processing time than CNNs. It was observed that both analyzed techniques presented exceptional performance, correctly classifying the signals from all evaluated damage scenarios, since the parameters inherent to their respective operations are correctly calibrated. The optimization of these parameters was not the target of this article. Although the article used a cantilever beam as an example, which is a bit far from applications in real structures, it is observed that the presented methodology can be extended to more complex problems. In other words, the article's example may not directly apply to real-world structures, but the methodology described can be adapted and used for more complex problems involving structural health monitoring. A positive outcome of this work is that both methods can take the displacements regardless of the position in the beam they have been acquired from. Therefore, both methods are robust with respect to camera relative position, which is not common in camera-based damage identification methods.

ACKNOWLEDGEMENTS

This study was financed by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Brazil) - grants CNPq/ FNDCT/ MCTI 407256/2022-9, 308008/2021-9 and 304329/2019-3; Fundação de Amparo à Pesquisa do Estado de Minas Gerais - FAPEMIG - grants TEC PPM 00106-17, TEC PPM-00001-18 and BPD 00080-22; Consejo Nacional de Investigaciones Científicas y Técnicas de Argentina (CONICET); Universidad Nacional de Cuyo; and Universidade Federal de Juiz de Fora.

REFERENCES

[1] Cardoso, R., Cury, A., Barbosa, F. A clustering-based strategy for automated structural modal identification. *Structural Health Monitoring* (2018) **17**(2):201–217. https://doi.org/10.1177/1475921716689239

[2] Nunes, L.A., Amaral, R.P.F., Barbosa, F.S., Cury, A.A. A hybrid learning strategy for structural damage detection. *Structural Health Monitoring* (2021) **20**(4):2143–2160. https://doi.org/10.1177/1475921720966943

[3] Finotti, R.P., Gentile, C., Barbosa, F.S., Cury, A.A. A novel natural frequency-based technique to detect structural changes using computational intelligence. *Procedia engineering* (2017) **199**:3314–3319. https://doi.org/10.1016/j.proeng.2017.09.438

[4] Finotti, R., Barbosa, F., Cury, A., Pimentel, R. Numerical and Experimental Evaluation of Structural Changes Using Sparse Auto-Encoders and SVM Applied to Dynamic Responses. *Applied Sciences* (2021) **11**(24):11965. https://doi.org/10.3390/app112411965

[5] Resende, L., Finotti, R., Barbosa, F., Cury, A. Structural damage detection with autoencoding neural networks Health monitoring and numerical modeling of structures. In: Ibero-Latin-American Congress on Computational Methods in Engineering. *XLIII CILAMCE Proceedings*, Foz do Iguaçu (2022).

[6] Resende, L., Finotti, R., Barbosa, F., Garrido, H., Cury, A., Domizio, M. Damage identification using convolutional neural networks from instantaneous displacement measurements via image processing. *Structural Health Monitoring* (2023): 14759217231193102. https://doi.org/10.1177/14759217231193102

[7] Chen, Z., Wang, Y., Wu, J., Deng, C., Hu, K. Sensor data-driven structural damage detection based on deep convolutional neural networks and continuous wavelet transform. *Applied Intelligence* (2021) **51**:5598–5609. https://doi.org/10.1007/s10489-020-02092-6

[8] Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., Inman, D. 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. https://doi.org/10.1016/j.ymssp.2020.107077

[9] Finotti, R., Gentile, C., Barbosa, F., Cury, A. Structural novelty detection based on sparse autoencoders and control charts. *Structural Engineering and Mechanics* (2022) **81**(5):647–664. https://doi.org/10.12989/sem.2022.81.5.647

[10] Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. MIT Press, Cambridge (2016).

[11] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D. 1D Convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing* (2021) **151**:107398. https://doi.org/10.1016/j.ymssp.2020.107398

[12] Bishop, M. *Pattern Recognition and Machine Learning.* Springer-Verlag (2006).

[13] Kingma, D., Ba, J. Adam: A Method for Stochastic Optimization. In: 3rd International Conference for Learning Representations, San Diego (2015).
[14] Garrido, H., Domizio, M., Curadelli, O., Ambrosini, D. Synthesis of healthy-structure model responses for damage quantification. *Structural Health Monitoring* (2023) **22**(1):689–713. https://doi.org/10.1177/14759217221088493