Machine learning for automatic processing of modal analysis in damage detection of bridges

Machine Learning for Automatic Processing of Modal Analysis in Damage Detection of Bridges

Elia Favarelli, *Student Member, IEEE*, and Andrea Giorgetti, *Senior Member, IEEE*

***Abstract*—Autonomous structural health monitoring (SHM) of a large number of bridges became a topic of paramount importance for maintenance purposes and safety reasons. This paper proposes a set of machine learning (ML) tools to perform automatic detection of anomalies in a bridge structure from vibrational data. As a case study, we considered the *Z-24* bridge for which an extensive database of accelerometric data is available. The proposed framework starts from the stabilization diagram obtained through operational modal analysis (OMA) to perform the clustering of modal frequencies and their tracking by density-based time-domain filtering. The features extracted are then fed to an one-class classification (OCC) algorithm to perform anomaly detection. In particular, we propose two new anomaly detectors, namely one-class classifier neural network (OCCNN) and OCCNN^2, which find the normal class (the boundary of the features space in normal operating conditions) through a two-step approach: coarse and fine boundary estimate. The detection algorithms are then compared with known methods based on principal component analysis (PCA), kernel principal component analysis (KPCA), Gaussian mixture model (GMM), and autoassociative neural network (ANN). The proposed OCCNN solution presents increased accuracy and $\mathbf{F}_1$ score over conventional algorithms, without the need to set critical parameters, while OCCNN^2 provides the best performance in term of $\mathbf{F}_1$ score, accuracy, and responsiveness.**

***Index Terms*—Anomaly detection, neural network, sensor network, system identification, modal analysis, structural health monitoring, vibration measurement.**

I. Introduction

CIVIL infrastructures are the backbone of modern society, and the assessment of their conditions is of renowned importance. This aspect is even more exacerbated because existing systems, such as bridges, are fast approaching their service life. Since replacement of such structures is functionally and economically costly, maintenance and retrofitting operations must be planned wisely [1]. As far as bridges are concerned, some statistics highlight the relevance of the problem. For example, currently, in Italy there are almost 2000 bridges that require special monitoring; in France, 4000 bridges need to be restored, and 840 are considered in critical conditions; in Germany, 800 bridges are reputed critic; in the United States of America, among the 600.000 bridges, according to a conservative estimate, at least 1% of them is considered deficient.

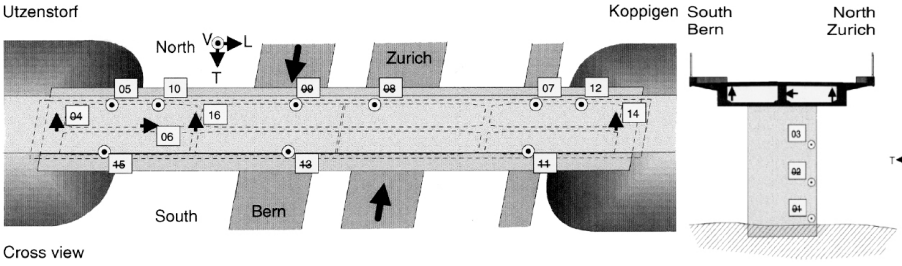
This work was supported by MIUR under the program "Dipartimenti di Eccellenza (2018-2022) - Precise-CPS," and by the CoACh project funded by the POR FESR 2014-2020 program. Part of this work has been presented at the IEEE Int. Conf. on Signal Proc. and Comm. Systems (ICSPCS), Dec. 2019.
The authors are with the Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi" (DEI), CNIT, University of Bologna, Italy (e-mail: {elia.favarelli2, andrea.giorgetti}@unibo.it).

Being able to detect damages is thus of paramount importance in this scenario, and structural health monitoring (SHM) offered numerous solutions [2]–[4].

Nowadays, the widespread need of methodologies able to determine the state of health of a structure, and the localization and quantification of damages, generated a vast literature, often reviewed in several works [5], [6]. Over the years, several techniques have been developed to extract the most significant damage-sensitive features. Such techniques can be divided into model-free and model-based. In model-free methods, the only information is gathered by measurements (e.g., acceleration, temperature, position), while in model-based approaches, information comes from measurements and prior knowledge of a model of the structure [7]. An example of model-free SHM propose operational modal analysis (OMA) based on natural vibrations, to extract damage sensitive features and then to detect anomalous changes in the features to evaluate the condition of the structure [8], [9]. Alternatively, a wide-band excitation is performed to stress the structure and extract more robust damage sensitive features [10]. The dependence of the features extracted on the environment makes the problem harder, and in some scenarios, that dependence must be mitigated, or exploited, to increase the damage detection accuracy [11]. An interesting area of research in SHM is related to the sensor network design, with particular emphasis on the type of sensors and their position [12]. For example, in [13], a sparse transducer array and optic fiber are proposed to detect and localize the damage within the structure. To alleviate some limitations of the traditional sensing networks, mobile agents able to perform sensing, and manage the energy supply are proposed [14].

Since the whole procedure is quite complex and requires fine-tuning of several parameters, specific for the structure at hand, recently, some works put forward the idea of adoption of machine learning (ML) techniques to detect changes in the damage sensitive features [15]–[17]. However, despite the abundant literature on SHM of bridges, much effort is still necessary to analyze data automatically to ensure the massive monitoring of a large number of infrastructures. Unfortunately, performing automatic SHM is not trivial because of the variety of structures and their peculiar characteristics, which, again, tend to require specific settings.

In this paper, we attempt to provide a methodology to perform automatic detection of anomalies in bridges starting from vibrational data. The proposed framework starts from the well-known stabilization diagram obtained through OMA and then performs clustering of modal frequencies and their tracking in the time domain. The modal frequencies are



Date (1998)	Scenario
4 August	Undamaged condition
9 August	Installation of pier settlement system
10 August	Lowering of pier, 20 mm
12 August	Lowering of pier, 40 mm
17 August	Lowering of pier, 80 mm
18 August	Lowering of pier, 95 mm
19 August	Lifting of pier, tilt of foundation
20 August	New reference condition
25 August	Spalling of concrete at soffit, 12 m^2
26 August	Spalling of concrete at soffit, 24 m^2
27 August	Landslide of 1 m at abutment
31 August	Failure of concrete hinge
2 September	Failure of 2 anchor heads
3 September	Failure of 4 anchor heads
7 September	Rupture of 2 out of 16 tendons
8 September	Rupture of 4 out of 16 tendons
9 September	Rupture of 6 out of 16 tendons

Figure 1. On the left, the data acquisition setup along the Z24 bridge: the selected accelerometers, their positions, and the measured acceleration direction; on the right, the chronological overview of applied damage scenarios [18].

thus used as features for a one-class classifier to perform anomaly detection. In particular, the main contributions are the following:

- We present a complete processing chain capable of monitoring the structural health of bridges autonomously using output-only system identification based on accelerometric data.
- After performing modal frequencies clustering in the stabilization diagram, we propose a new time-domain tracking algorithm based on modal frequencies density.
- We propose a new anomaly detector, namely one-class classifier neural network (OCCNN), which finds the normal class (the boundary of the features space in normal operating conditions) through a two-step approach.
- We propose a variant of OCCNN, named OCCNN^2, which combines the two-step approach of OCCNN with an autoassociative neural network (ANN).
- We applied the proposed framework on data collected from the *Z-24* bridge to assess its performance and perform comparisons with several anomaly detection techniques: principal component analysis (PCA), kernel principal component analysis (KPCA), Gaussian mixture model (GMM), and ANN.
- We prove the robustness of the proposed solution to different sensor configurations.

The performance of the proposed solution is investigated on a real dataset using the accelerometric data available for the *Z-24* bridge [19], [20]. The detection algorithms are compared with known methods based on PCA, KPCA, GMM, and ANN, in terms of F_1 score, accuracy, recall, and precision. Several tests are performed to assess the impact of the length of both the training set and the damaged set. In particular, this last test aims to understand the responsiveness of the algorithms to detect the damage. The proposed OCCNN solution presents considerable accuracy and F_1 score to conventional algorithms without the need to set specific parameters, while OCCNN^2 provides the best performance in term of F_1 score, accuracy, and responsiveness.

Throughout the paper, capital boldface letters denote matrices and tensors, lowercase bold letters denote vectors, $(\cdot)^{\mathrm{T}}$ stands for transposition, $(\cdot)^{+}$ indicates the Moore-Penrose pseudoinverse operator, $||\cdot||_2$ is the ℓ_2-norm of a vector, $||\cdot||_{\mathrm{F}}$ is the Frobenius norm of a matrix, $\Re\{\cdot\}$ and $\Im\{\cdot\}$ are the real and imaginary parts of a complex number, respectively, $\mathbb{V}\{\cdot\}$ is the variance operator, and $\mathbb{1}\{a, b\}$ is the indicator function equal to 1 when $a = b$, and zero otherwise.

The paper is organized as follows. In Section II, a brief review of the acquisition campaign, the accelerometers setup, and the data preprocessing steps are presented. The operational modal analysis adopted is sketched in Section III. The clustering and the novel density-based tracking algorithm are described in Section IV. The one-class classification (OCC) algorithms are presented in Section V, while the new anomaly detection methods based on neural networks (NNs) are detailed in Section VI. In Section VII the computational cost of the algorithms is discussed. Numerical results are given in Section VIII. Conclusions are drawn in Section IX.

II. System Configuration and Data Collection

The *Z-24* bridge was located in the Switzerland canton Bern. The bridge was a part of the road connection between Koppigen and Utzenstorf, overpassing the A1 highway between Bern and Zurich. It was a classical post-tensioned concrete two-cell box girder bridge with a main span of 30 m and two side spans of 14 m. The bridge was built as a freestanding frame with the approaches backfilled later. Both abutments consisted of triple concrete columns connected with concrete hinges to the girder. Both intermediate supports were concrete piers clamped into the girder. An extension of the bridge girder at the approaches provided a sliding slab. Al supports were rotated with respect to the longitudinal axis that yielded a skew bridge. The bridge was demolished at the end of 1998 [19]. Before complete demolition, the bridge was subjected to a long-term continuous monitoring test and several progressive damage tests (see also Fig. 1):

- A long-term continuous monitoring test took place during the year before demolition. The aim was to quantify the environmental variability of the bridge dynamics.
- Progressive damage tests took place over a month, shortly before complete demolition. The aim was to prove experimentally that realistic damage has a measurable influence on the bridge dynamics. Progressive damage tests were alternated with short-term monitoring tests while the

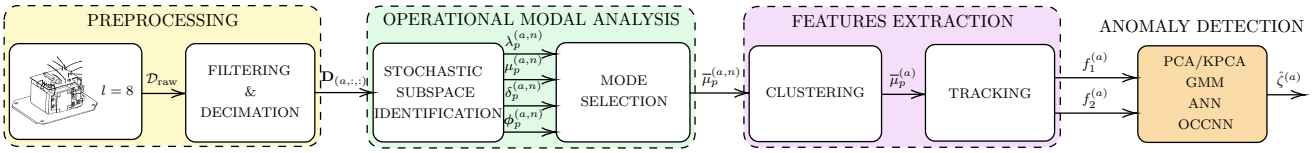


Figure 2. The block diagram for signal acquisition, processing, feature extraction, and detection.

continuous monitoring system was still running during these tests.

Table I
DATA SYMBOL, DIMENSION, UNIT OF MEASURE, AND DESCRIPTION

Variable	Dimension	Unit	Description
$\boldsymbol{\mathcal{D}}_{\text{raw}}$	$N_a \times l \times N_s$	m/s^2	Raw accelerometer data
$\boldsymbol{\mathcal{D}}_{(a,:,:)}$	$N_a \times l \times N$	m/s^2	Preprocessed accelerometer data
$\lambda_p^{(a,n)}$	1×1	Hz	Eigenvalues
$\mu_p^{(a,n)}$	1×1	Hz	Natural frequencies
$\delta_p^{(a,n)}$	1×1		Dumping ratios
$\boldsymbol{\phi}_p^{(a,n)}$	$l \times 1$		Mode shapes
$\bar{\mu}_p^{(a,n)}$	1×1	Hz	Natural freq. after mode selection
$\bar{\bar{\mu}}_p^{(a)}$	1×1	Hz	Natural frequencies after clustering
$f_1^{(a)}$	1×1	Hz	First natural freq. after tracking
$f_2^{(a)}$	1×1	Hz	Second natural freq. after tracking
$\hat{\zeta}^{(a)}$	1×1		Decision labels

A. Data collection

The database of accelerometric measurements is freely available for research purposes on the website [18]. Such a dataset was build within the framework of the European Brite EuRam research project BE-3157, "System Identification to Monitor Civil Engineering Structures" (SIMCES). One of the main objectives of the SIMCES project was to deliver a proof of feasibility for vibration-based structural health monitoring of civil engineering structures by full-scale, long-term tests and progressive failure tests of a representative structure, the *Z-24* bridge.

The accelerometers used are force balance type FBA-11 by Kinemetrics [21]; they are triaxial devices with high sensitivity and are characterized by high reliability and low current consumption. The main specifications are: full scale range ± 9.81 m/s^2, output ± 2.5 V, natural frequency critical damping 0.7, and supply voltage ± 12 V DC. The A/D converter used is the IOtech ADC 488/8SA [22], an 8 channel sample and hold with 8 differential inputs, 16 bit, and 100 kHz sampling rate.

The accelerometers position and their measurements axis are shown in Fig. 1. In this work, we considered $l = 8$ accelerometers, identified as 03, 05, 06, 07, 10, 12, 14, and 16, which are present in both long-term continuous monitoring phase and in the progressive damage one.[1] Longitudinal acceleration is collected by sensors 03 and 06, transversal acceleration is measured by sensors 14 and 16, and all the remaining sensors gather vertical accelerations.

Every hour $N_{\text{s}} = 65536$ samples are acquired from each sensor with sampling frequency $f_{\text{samp}} = 100$ Hz which corresponds to an acquisition time $T_{\text{a}} = 655.36$ s. Since the measurements are not always available, there are $N_{\text{a}} = 4107$ acquisitions collected in a period of 44 weeks. From now on, all the data collected by the selected sensors are organized in a tensor $\boldsymbol{\mathcal{D}}_{\text{raw}}$ of dimensions $N_{\text{a}} \times l \times N_{\text{s}}$.

During the long-term monitoring, also environmental parameters like temperature, wind speed, and traffic were taken before and after each accelerometer measurement. In this work, only the temperature is considered, consisting of 20 measurements acquired by different thermometers placed along with the structure.

B. Data pre-processing

The block diagram depicted in Fig. 2 represents the sequence of tasks performed for the fully automatic anomaly detection approach presented in this work.

To reduce the computational cost and memory needs of the subsequent elaborations, some pre-processing steps have been applied to the data $\boldsymbol{\mathcal{D}}_{\text{raw}}$. First, a decimation by a factor 2 is applied to each acquisition; hence the sampling frequency is scaled to $f_{\text{samp}} = 50$ Hz. Such sampling frequency is deemed sufficient because the *Z-24* fundamental frequencies fall in the $[2.5,\ 20]$ Hz frequency range [19]. After decimation, data are processed with a finite impulse response (FIR) band-pass filter of order 30 with band $[2.5,\ 20]$ Hz, to remove disturbances. The pre-processed data are then stored in a tensor $\boldsymbol{\mathcal{D}}$ of size $N_{\text{a}} \times l \times N$, where $N = 32000 \simeq N_{\text{s}}/2$. Regarding the environmetal parameters, the temperatures collected are averaged over time and among sensors to have one estimate for the whole bridge each hour. The main matrices, vectors and scalars introduced are summarised in Table I.

III. Review of Operational Modal Analysis

A widely adopted system identification technique for bridge monitoring is the output-only modal identification approach. This OMA technique relies on the assumption that the system excitation (either natural phenomena like wind or the traffic) is wideband when compared to the frequency bandwidth of the structure. Such a hypothesis allows considering the excitation as white noise in the band of interest so that all the modes of the structure are equally excited, and the output spectrum contains full information about the structure itself. Within this framework, a large set of possible algorithms can be

[1] Some accelerometers that experienced failures during the long-term monitoring have been avoided. Moreover, we select a subset of accelerometers present in both phases to ensure data consistency.

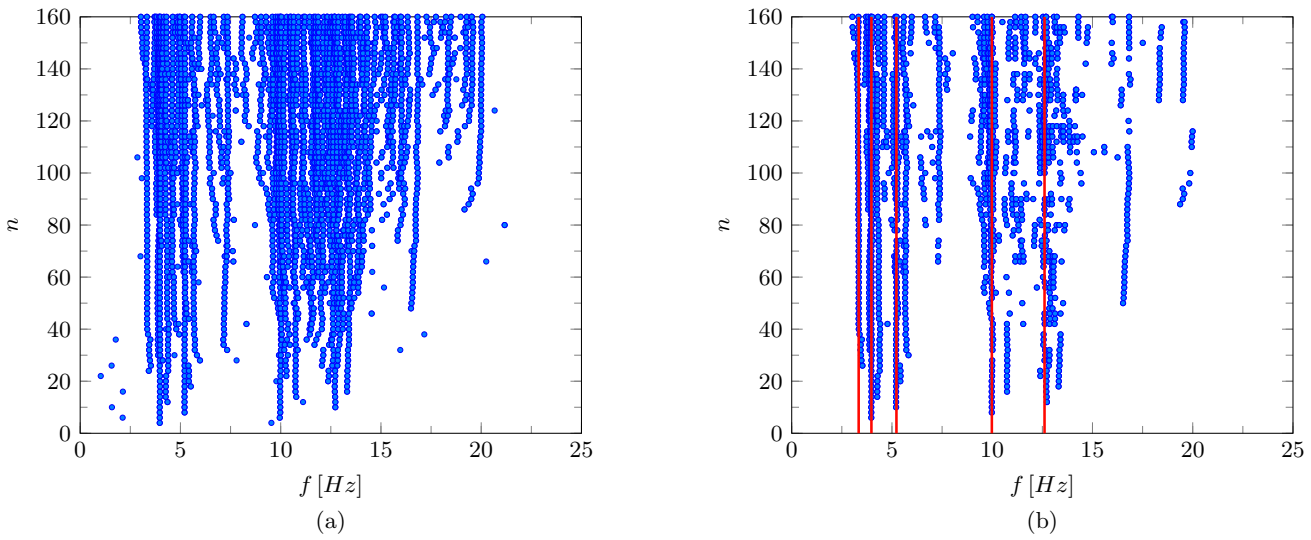


Figure 3. Example of stabilization diagram for the first hour monitoring: a) through SSI, b) after mode selection and clustering.

used to evaluate the fundamental frequencies of a structure. Peak picking is the simplest one but presents some theoretical and practical drawbacks: picking the peaks is always a subjective task; operational deflection shapes are obtained instead of mode shapes; only real modes or proportionally damped structures can be deduced by the method; damping estimates can be unreliable [23], [24]. In order to make the procedure automated and more reliable, we choose the so-called SSI, which is a time-domain, parametric, covariance-driven procedure for blind modal analysis, i.e., it can extract the modal frequencies from the accelerometric measurements without any a priori knowledge about the structure [7].

A. Stochastic subspace identification

SSI is characterized by the system order $n \in \mathbb{N}$ and the time-lag $i \geq 1$. To apply the algorithm it is enough to satisfy the following constraint, $l \cdot i \geq n$ [7]. The system order n is considered unknown, so it is kept as a parameter varied in the range $n \in [2, 160]$ (with step 2), while the time-lag is $i = 60$ [7].

Let us define the block Toeplitz matrix for a given time-lag i and shift s

$$\mathbf{T}_{s|i}^{(a)} = \begin{bmatrix} \mathbf{R}_i^{(a)} & \mathbf{R}_{i-1}^{(a)} & \dots & \mathbf{R}_{s+1}^{(a)} & \mathbf{R}_s^{(a)} \\ \mathbf{R}_{i+1}^{(a)} & \mathbf{R}_i^{(a)} & \dots & \mathbf{R}_{s+2}^{(a)} & \mathbf{R}_{s+1}^{(a)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{R}_{2i-1}^{(a)} & \mathbf{R}_{2i-2}^{(a)} & \dots & \mathbf{R}_{i+1}^{(a)} & \mathbf{R}_i^{(a)} \end{bmatrix} \tag{1}$$

of dimensions $li \times li$ where

$$\mathbf{R}_i^{(a)} = \frac{1}{N-i} \mathbf{D}_{(a,1:N-i,:)} \mathbf{D}_{(a,i:N,:)}^{\mathrm{T}} \tag{2}$$

is a correlation matrix of dimension $l \times l$, and the data matrix $\mathbf{D}_{(a,b:c,:)}$ is obtained by the tensor $\boldsymbol{\mathcal{D}}$ selecting a particular acquisition a from all the sensors in the time interval $[b, c]$. For the sake of presentation we drop the acquisition index a, considering that all the subsequent operations are performed for each acquisition. Now, by the singular values decomposition (SVD) we can factorize the block Toeplitz matrix (1), with $s = 1$, as

$$\mathbf{T}_{1|i} = \mathbf{U}^{(n)} \boldsymbol{\Sigma}^{(n)} \mathbf{V}^{(n)\mathrm{T}} \tag{3}$$

where $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)\mathrm{T}}$ are matrices of dimensions respectively $li \times n$ and $n \times li$ and $\boldsymbol{\Sigma}$ is a diagonal matrix, of dimension $n \times n$, that contain the singular values arranged in descending order. For the sake of presentation, we drop the model order index n, considering that all the subsequent observations can be applied for each model order. The SVD allows selecting the correct number of singular values that must be considered, and therefore to split the matrix $\mathbf{T}_{1|i}$ in two parts

$$\mathbf{T}_{1|i} = \mathbf{O}_i \boldsymbol{\Gamma}_i \tag{4}$$

$$\mathbf{O}_i = \mathbf{U} \boldsymbol{\Sigma}^{1/2} \mathbf{S} \tag{5}$$

$$\boldsymbol{\Gamma}_i = \mathbf{S}^{-1} \boldsymbol{\Sigma}^{1/2} \mathbf{V}^{\mathrm{T}} \tag{6}$$

where

$$\mathbf{O}_i = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{i-1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Gamma}_i = \begin{bmatrix} \mathbf{A}^{i-1}\mathbf{G} & \dots & \mathbf{AG} & \mathbf{G} \end{bmatrix} \tag{7}$$

represent, respectively, the observability matrix and the reversed controllability matrix. In (5) and (6) the matrix $\mathbf{S}$ plays the role of a similarity transformation applied to the state-space model, therefore it can be set equal to the identity matrix $\mathbf{I}$. The matrices $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{G}$, in equation (7) represent the state matrix, the output influence matrix, and the next state-output covariance matrix, respectively. Note that $\mathbf{C}$ and $\mathbf{G}$ can be easily extracted from the matrices $\mathbf{O}_i$ and $\boldsymbol{\Gamma}_i$, while $\mathbf{A}$ can be calculated by (1) as

$$\mathbf{A} = \mathbf{O}_i^{+} \mathbf{T}_{2|i+1} \boldsymbol{\Gamma}_i^{+}. \tag{8}$$

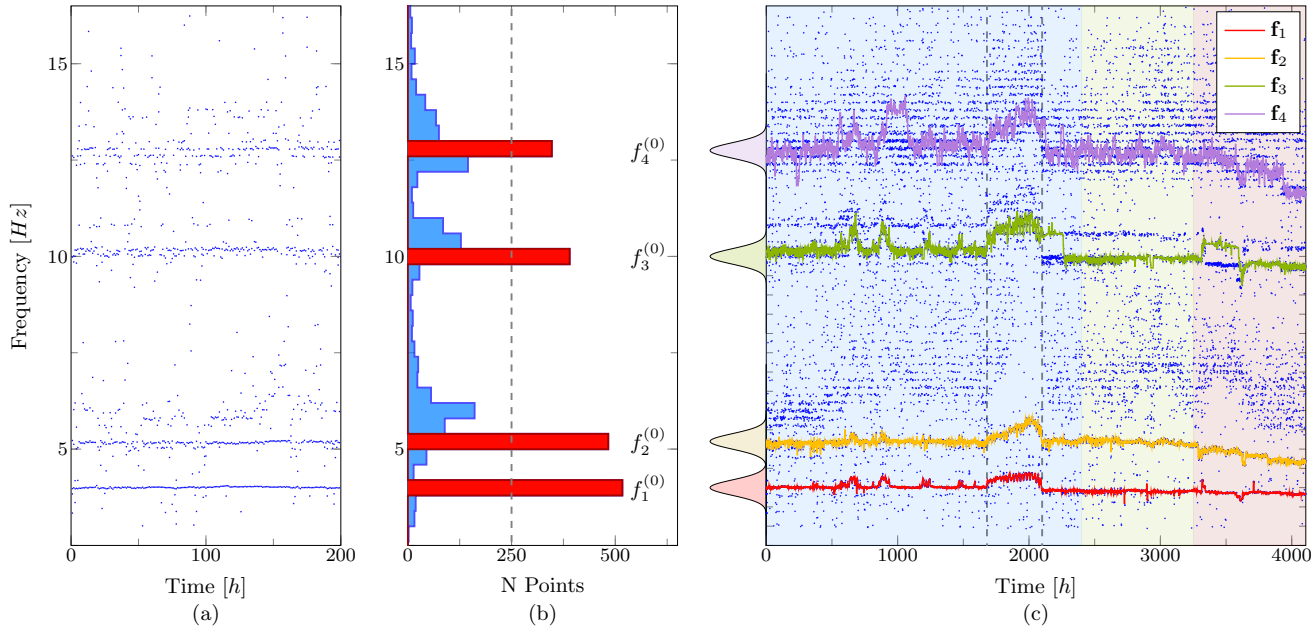


Figure 4. a) Residual modes after mode selection over time with one acquisition per hour (first 200 acquisitions). b) Natural frequencies selected after the initial phase of the tracking algorithm. c) First two natural frequencies estimation after the density-based tracking algorithm. Blue points represent the residual modes, red and yellow tracks represent, respectively, the first and second fundamental frequency estimated after the tracking algorithm; vertical dashed lines highlight the period when the average measured temperatures are below 0 °C [11], in particular, it has been demonstrated in [25] that when the temperature goes below 0 °C the natural frequencies of the bridge increase. Blue and green backgrounds highlight the acquisitions made during the normal condition of the bridge, used respectively as training and test sets, while red background stands for damaged condition acquisitions used in the test phase.

Applying the eigenvalues decomposition to $\mathbf{A}$ we get

$$\mathbf{A} = \mathbf{\Psi \Omega \Psi}^{\mathrm{T}} \tag{9}$$

where $\mathbf{\Psi}$ is an orthonormal matrix whose columns are the eigenvectors, while $\mathbf{\Omega} = \mathrm{diag}(\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n)$ is a diagonal matrix that contains the n eigenvalues of the state matrix. Therefore, reintroducing the previously omitted indices, it is possible to estimate the continuous-time damage sensitive parameters of the pth mode as follows:

- eigenvalues $\lambda_p^{(a,n)} = f_s \ln(\widetilde{\lambda}_p^{(a,n)})$;
- natural frequencies $\mu_p^{(a,n)} = |\lambda_p^{(a,n)}|/(2\pi)$;
- dumping ratios $\delta_p^{(a,n)} = -\Re\{\lambda_p^{(a,n)}\}/|\lambda_p^{(a,n)}|$;
- mode shapes $\boldsymbol{\phi}_p^{(a,n)} = \mathbf{C}^{(a,n)}\boldsymbol{\psi}_p^{(a,n)}$;

where $\boldsymbol{\phi}_p^{(a,n)}$ is a $l \times 1$ vector, and $\boldsymbol{\psi}_p^{(a,n)}$ is the pth column vector of $\mathbf{\Psi}^{(a,n)}$ defined in (9). The natural frequencies extracted through this approach for the first acquisition ($a = 1$) varying the order n is depicted in the stabilization diagram reported in Fig. 3a.

B. Mode selection

Through SSI, a large number of spurious modes are generated. To sift them out, there are several approaches to distinguish between the real modes and the spurious ones [26]. In this work we used four criteria, which must be satisfied to select a physic mode and reject the spurious ones. The four metrics extracted are: modal assurance criterion (MAC), mean phase deviation (MPD), dumping ratio check, and complex conjugate poles check. Among several criteria, we choose to use these four metrics as a good compromise between computational cost and accuracy of the spurious modes detection. The metrics selected are widely used and discussed in literature [26]–[29].

1) MAC: It is a dimensionless squared correlation coefficient between mode shapes, defined as [27]

$$\mathrm{MAC}(\boldsymbol{\phi}_p^{(a,n)}, \boldsymbol{\phi}_q^{(a,j)}) = \frac{|\boldsymbol{\phi}_p^{(a,n)\mathrm{T}}\boldsymbol{\phi}_q^{(a,j)}|^2}{||\boldsymbol{\phi}_p^{(a,n)}||_2^2||\boldsymbol{\phi}_q^{(a,j)}||_2^2} \tag{10}$$

with values between 0 and 1. A MAC larger than 0.9 indicates a consistent correspondence between the modes and so, very likely physical modes, whereas small values indicate poor resemblance of the two shapes and so a spurious mode. A validation criteria based on MAC is the following [20]

$$d_{\mathrm{m}}(a,n,j,p,q) = \frac{|\lambda_p^{(a,n)} - \lambda_q^{(a,j)}|}{\max(|\lambda_p^{(a,n)}|, |\lambda_q^{(a,j)}|)} + 1 - \mathrm{MAC}(\boldsymbol{\phi}_p^{(a,n)}, \boldsymbol{\phi}_q^{(a,j)}) \tag{11}$$

where the first term is a measure of the distance between the pth and qth eigenvalues. With (11) a mode is considered physical when $d_{\mathrm{m}}(a,n,j,p,q) < 0.1$.

2) MPD: It measures the deviation of a mode shape components from the mean phase (MP) of all its components. A widely used technique to evaluate the MP is through the SVD [20]

$$\mathbf{P\Lambda Q}^T = [\Re\{\boldsymbol{\phi}_p^{(a,n)}\} \;\; \Im\{\boldsymbol{\phi}_p^{(a,n)}\}] \tag{12}$$

where $\mathbf{P}$ is $l \times 2$, $\mathbf{\Lambda}$ is 2×2, and $\mathbf{Q}$ is 2×2. From this decomposition the MPD can be evaluated as follows

$$\mathrm{MPD}(\boldsymbol{\phi}_p^{(a,n)}) = \frac{\sum_{r=1}^{l} |\phi_{r,p}^{(a,n)}| \arccos \left| \frac{\Re\{\phi_{r,p}^{(a,n)}\} q_{22} - \Im\{\phi_{r,p}^{(a,n)}\} q_{12}}{\sqrt{q_{12}^2 + q_{22}^2} |\phi_{r,p}^{(a,n)}|} \right|}{\sum_{r=1}^{l} \phi_{r,p}^{(a,n)}}. \quad (13)$$

When the ratio $\mathrm{MPD}(\boldsymbol{\phi}_p^{(a,n)})/(\pi/2) > 0.75$ the deviation of the mode shape components is stronger than MP and probably the mode is spurious, hence it is rejected.

3) Damping ratio and complex conjugate poles: For each mode the dumping ratio $\delta_p^{(a,n)}$ in real structures must be positive and lower than 0.2 (otherwise the structure is unstable) hence only modes with $0 < \delta_p^{(a,n)} < 0.2$ are considered. Moreover if $\Re\{\lambda_p^{(a,n)}\} > 0$ the mode represent an unstable structure hence it is considered spurious.

In Fig. 3b, the stabilization diagram after mode selection is shown, the remaining modes are represented with $\bar{\mu}_p^{(a,n)}$ (see Fig. 2).

IV. Features Extraction

After mode selection, we propose clustering and a novel tracking algorithm to extract the time-series of the first two fundamental frequencies.

A. Clustering

The stabilization diagram obtained after mode selection needs to be further processed to select the natural frequencies. For this purpose, K-means can be used to cluster the data, associating each cluster to a prospective natural frequency corresponding to its centroid [15], [30]. For each acquisition the algorithm starts with an initial number of centroids (e.g., $K = 4$) initialized at random frequencies within the range of interest, and then their position updated until convergence. Since the number of natural frequencies is unknown, the approach is repeated for different K values, varied between 2 and 6; a large K tends to produce many spurious modes while small values may discard real fundamental frequencies. Finally, the centroids configuration that leads to the solution with the lowest error (evaluated as the sum of the Euclidean distances between the centroids and the associated points) is selected. The output of K-means is the number of natural frequencies and their estimation (red lines in Fig. 3(b)). Repeating this procedure for each acquisition the blue points depicted in Fig. 4a are obtained.

At the end of clustering, the estimated frequencies are no more dependent on the model order n, consequently the notation $\bar{\mu}_p^{(a)}$ is adopted (see Fig. 2).

B. Density-based mode tracking

The tracking phase is the final step in the natural frequencies extraction chain. There are several algorithms able to find the frequency traces starting from the estimation made through the clustering phase [28]. In this paper, a novel technique that needs neither the frequency starting points nor their number is proposed, contrary to several approaches that infer the number and the starting position of the fundamental frequencies through simulation of the physical structure. The proposed tracking algorithm consists of two steps: a starting phase and an online phase.

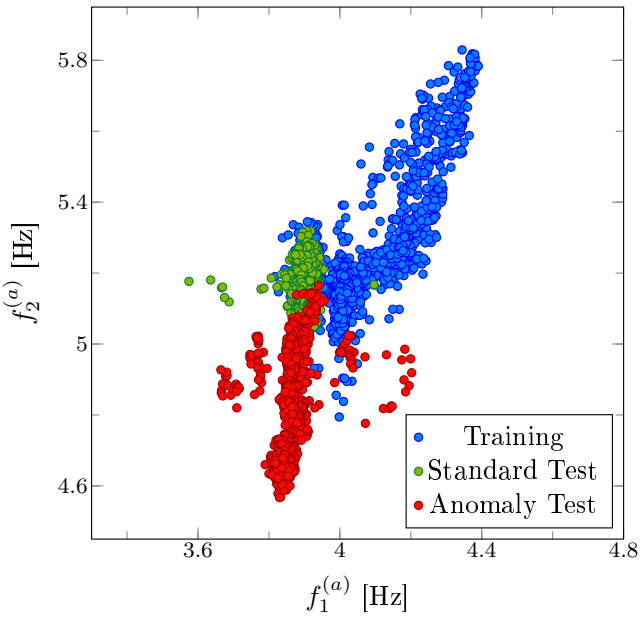


Figure 5. First two natural frequencies plotted in a bi-dimensional feature space.

1) Initial phase: Without any assumption about the structure, the idea is to analyze the data, $\bar{\mu}_p^{(a)}$, to find some clusters of points that could be the starting ones. To perform this task, the first $N_\mathrm{t} = 200$ acquisitions, i.e., $\bar{\mu}_p^{(a)}$ with $a = 1, 2, \ldots, N_\mathrm{t}$, are considered (see Fig. 4a). From this initial data, the number of points that fall in frequency bins of bandwidth $B_\mathrm{f} = 0.4\,\mathrm{Hz}$ are counted. The histogram obtained is depicted in Fig. 4b. Selecting the largest values of the histogram the number of starting points and the corresponding frequencies, $f_s^{(0)}$, are estimated. In particular, the first estimated frequency is evaluated as the average values of the frequencies that fall in the respective bins. For example, according to Fig. 4b the values of the starting points, $f_s^{(0)}$, in this case $s = 1, \ldots, 4$, are estimated and correspond to 4.0, 5.2, 10.1 and 12.8 Hz.

2) Online phase: In this phase, for each starting point a Gaussian kernel of the form

$$\mathcal{G}\left(\theta; f_s^{(a)}, \sigma_\mathrm{f}\right) = e^{-\frac{1}{2\sigma_\mathrm{f}^2}\left(\theta - f_s^{(a)}\right)^2} \quad (14)$$

is used to track the frequencies evolution over time. The parameter σ_f controls the kernel width and has been chosen equal to $\sigma_\mathrm{f} = 0.01\,\mathrm{Hz}$; larger values of σ_f make the system more reactive to fast frequency changes during the tracking but more sensitive to outliers due to the noisy measurements. The tracking algorithm is initialized with $f_s^{(0)}$ and iteratively updated through the following rule

$$f_s^{(a)} = (1 - \epsilon) f_s^{(a-1)} + \epsilon \widetilde{f}_s^{(a)} \quad (15)$$

where the parameter $\epsilon \in [0, 1]$ controls the impact of the new observation. Large values of ϵ reduce smoothness but

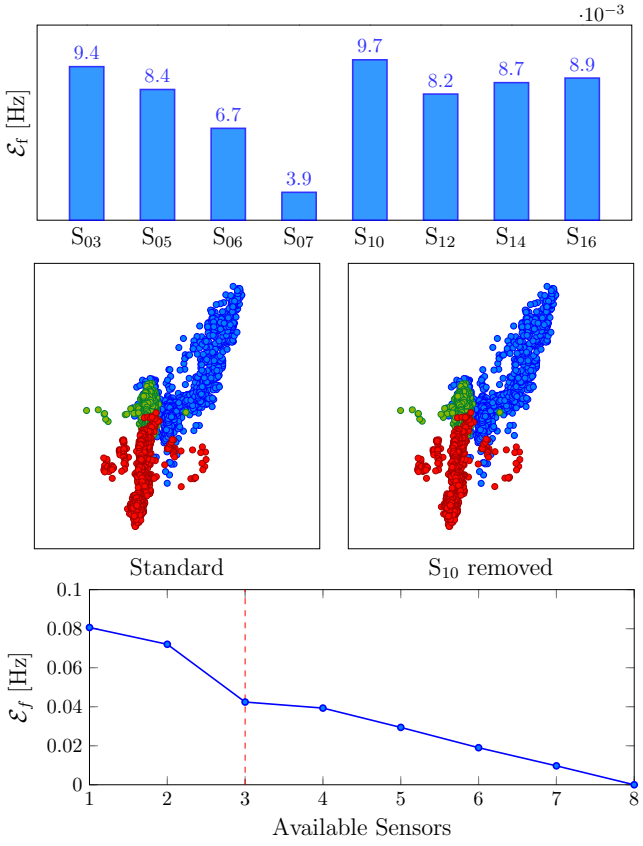


Figure 6. On the top, the RMSE of the estimated frequencies when single sensors are removed from the estimation process. In the middle, the effect of sensor S_{10} removal on the feature space points. On the bottom, the effect of the number of sensors available; 3 represents the minimum number of sensors that ensure enough accurate frequency estimation.

allows capturing sudden changes of modal frequencies. For the specific data set $\epsilon = 0.7$ is selected. The innovation term $\widetilde{f}_s^{(a)}$ in (15) is evaluated through the Gaussian kernel as

$$\widetilde{f}_s^{(a)} = \frac{\sum_p \bar{\mu}_p^{(a)} \mathcal{G}\left(\bar{\mu}_p^{(a)}; f_s^{(a-1)}, \sigma_\mathrm{f}\right)}{\sum_p \mathcal{G}\left(\bar{\mu}_p^{(a)}; f_s^{(a-1)}, \sigma_\mathrm{f}\right)}. \tag{16}$$

The four tracks $\mathbf{f}_s = \{f_s^{(a)}\}_{a=1}^{N_\mathrm{a}}$ are shown in Fig. 4.

From now on, only the first two frequencies are considered, as deemed sufficient for anomaly detection in the Z-24 bridge [31]. The selected frequencies are stored in the following matrix

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}^\mathrm{T} = \begin{bmatrix} f_1^{(1)} & f_1^{(2)} & \dots & f_1^{(N_\mathrm{a})} \\ f_2^{(1)} & f_2^{(2)} & \dots & f_2^{(N_\mathrm{a})} \end{bmatrix}^\mathrm{T}. \tag{17}$$

C. Robustness

To prove that our framework is enough robust and applicable to different setups, we artificially remove accelerometric data from the input of SSI to evaluate the performance degradation of the whole monitoring system. More precisely, sensors' relevance is determined by removing sensors one by one and evaluating the error resulting in the feature space points with respect to the standard condition. The error is calculated as the RMSE

$$\mathcal{E}_\mathrm{f} = \frac{||\mathbf{F} - \hat{\mathbf{F}}||_\mathrm{F}}{\sqrt{N_\mathrm{a}}}$$

where $\mathbf{F}$ is matrix of natural frequencies in (17), and $\hat{\mathbf{F}}$ is the same matrix estimated when the sensor is removed. As can be seen in Fig. 6, sensor S_{10} generates the most significant error in the fundamental frequencies extraction, when removed. Moreover, removing sensors progressively, it is interesting to note that if at least 3 of them are present, the error in fundamental frequencies estimation is tolerable as the resulting classification accuracy is greater than 90% for OCCNN and OCCNN^2. Therefore, our results show that the features extraction process is rather robust with respect to the number of sensors and their position on the bridge.

V. Survey of Anomaly Detection Techniques

In this section we briefly review PCA, KPCA, GMM and ANN, which are often adopted for OCC [32]–[39].

As described in [19], the damage is introduced at the acquisition $a = N_\mathrm{d} = 3253$, corresponding to the installation of the lowering system. Therefore, through this section, the matrix $\bar{\mathbf{X}} = \mathbf{F}_{1:2N_\mathrm{d}-N_\mathrm{a}-1,:}$ contains the training points (blue background in Fig. 4c), $\bar{\mathbf{Y}} = \mathbf{F}_{2N_\mathrm{d}-N_\mathrm{a}:N_\mathrm{d}-1,:}$ contains the test points in standard condition (green background in Fig. 4c), and $\bar{\mathbf{U}} = \mathbf{F}_{N_\mathrm{d}:N_\mathrm{a},:}$ contains the test points in damaged condition (red background in Fig. 4c). The three subsets of acquisitions that correspond to training, test, and damaged points are, respectively, $\mathcal{I}_\mathrm{x} = \{1, ..., 2N_\mathrm{d} - N_\mathrm{a} - 1\}$, $\mathcal{I}_\mathrm{y} = \{2N_\mathrm{d} - N_\mathrm{a}, ..., N_\mathrm{d} - 1\}$, and $\mathcal{I}_\mathrm{u} = \{N_\mathrm{d}, ..., N_\mathrm{a}\}$.

Let us define the offset $\hat{\mathbf{x}}$ as the column vector containing the row-wise mean of the matrix $\bar{\mathbf{X}}$, and the rescaling factor $x_\mathrm{m} = \max_{a,s} |\bar{x}_{a,s} - \hat{x}_a|$. Before proceeding with the anomaly detection, the matrices $\bar{\mathbf{X}}$, $\bar{\mathbf{Y}}$ and $\bar{\mathbf{U}}$ are centered and normalized subtracting the offset $\hat{\mathbf{x}}$ row-wise and dividing each entry by the rescaling factor x_m. The resulting data matrices are $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{U}$, of size $N_\mathrm{x} \times D$, $N_\mathrm{y} \times D$, and $N_\mathrm{u} \times D$, respectively, with $D = 2$ features.

A. Principal component analysis

This technique remaps the training data from the feature space $\mathbb{R}^D$ in a subspace $\mathbb{R}^P$ (where $P < D$ is the number of components selected) that minimizes the error (defined as Euclidean distance) between the data in the feature space and their projection in the selected subspace [40]. More in detail, to find the best subspace to project the training data, we need to evaluate the $D \times D$ sample covariance matrix

$$\mathbf{\Sigma}_\mathrm{x} = \frac{\mathbf{X}^\mathrm{T}\mathbf{X}}{N_\mathrm{x} - 1}. \tag{18}$$

By eigenvalue decomposition $\mathbf{\Sigma}_\mathrm{x}$ can be factorized as $\mathbf{\Sigma}_\mathrm{x} = \mathbf{V}_\mathrm{x}\mathbf{\Lambda}_\mathrm{x}\mathbf{V}_\mathrm{x}^\mathrm{T}$, where $\mathbf{V}_\mathrm{x}$ is an orthonormal matrix whose columns are the eigenvectors, while $\mathbf{\Lambda}_\mathrm{x}$ is a diagonal matrix that contains the D eigenvalues. The eigenvalues magnitude represents the importance of the direction pointed by the relative eigenvector. In our setting we select the largest component, hence $P = 1$, therefore the best linear subspace of dimension

one is $\mathbf{v}_\text{P}$, which coincides with the eigenvector related to the largest eigenvalue of $\mathbf{\Sigma}_\text{x}$. The projection into the subspace is obtained multiplying the data by $\mathbf{v}_\text{P}$, i.e., $\mathbf{x}_\text{P} = \mathbf{X}\mathbf{v}_\text{P}$, $\mathbf{y}_\text{P} = \mathbf{Y}\mathbf{v}_\text{P}$, and $\mathbf{u}_\text{P} = \mathbf{U}\mathbf{v}_\text{P}$.

To evaluate the error between the projected points and the starting ones it is necessary to reconstruct the data in the original feature space, i.e., $\widetilde{\mathbf{X}} = \mathbf{x}_\text{P}\mathbf{v}_\text{P}^\text{T}$, $\widetilde{\mathbf{Y}} = \mathbf{y}_\text{P}\mathbf{v}_\text{P}^\text{T}$, and $\widetilde{\mathbf{U}} = \mathbf{u}_\text{P}\mathbf{v}_\text{P}^\text{T}$. After the reconstruction, it is possible to evaluate the error as the Euclidean distance between the original data and reconstructed data.

B. Kernel principal component analysis

In our feature space, the linear boundaries found by PCA represent a severe limitation [41]. KPCA firstly maps the data with a non-linear function, named kernel, then applies the standard PCA to find a linear boundary in the new feature space. Such boundary becomes non-linear in the original feature space. A crucial point in KPCA is the selection of the kernel that leads to linearly separable data in the new feature space. In [42], when the data distribution is unknown, the radial basis function (RBF) kernel is proposed as good candidate to accomplish this task. Given a generic point $\mathbf{z}$ that correspond to a $1 \times D$ vector, we can apply the RBF as

$$K_n^{(\mathbf{z})} = e^{-\gamma||\mathbf{z}-\mathbf{x}_n||^2} \quad \text{with} \quad n = 1, 2, \ldots, N_\text{x} \tag{19}$$

where γ is a kernel parameter (which controls the width of the Gaussian function) that must be set properly, $\mathbf{x}_n$ is the nth row of $\mathbf{X}$, and $K_n^{(\mathbf{z})}$ is the nth component of the point $\mathbf{z}$ in the kernel space. Overall, the vector $\mathbf{z}$ is mapped in the vector $\mathbf{k}^{(\mathbf{z})} = [K_1^{(\mathbf{z})}, K_2^{(\mathbf{z})}, \ldots, K_{N_\text{x}}^{(\mathbf{z})}]$. Remapping all the data in the kernel space, we obtain the subsequent matrices $\mathbf{K}_\text{x}$ of size $N_\text{x} \times N_\text{x}$ for training, $\mathbf{K}_\text{y}$ of size $N_\text{y} \times N_\text{x}$ for validation, and $\mathbf{K}_\text{u}$ of size $N_\text{u} \times N_\text{x}$ for test, respectively.

Applying now the PCA to the new data sets, it is possible to find non-linear boundaries in the original feature space.

C. Gaussian mixture model

Another important algorithm that can be used to solve OCC problems is the GMM [43]. This approach assumes that data can be represented by a mixture of $\mathcal{M}$ multivariate Gaussian distributions. The outputs of the algorithm are the covariance matrices, $\mathbf{\Sigma}_m$, and the mean values, $\boldsymbol{\mu}_m$, of the Gaussian functions, with $m = 1, 2, \ldots, \mathcal{M}$. The GMM algorithm finds the set of parameters $\mathbf{\Sigma}_m$ and $\boldsymbol{\mu}_m$ of a Gaussian mixture that better fit the data distribution through iterative algorithms like stochastic gradient descent or Newton-Raphson [15], [16].

D. Autoassociative neural network

The last algorithm considered is ANN [17]. This particular structure of the neural network is composed of the following sections:

- Mapping layers, which consist of one or more hidden layers, with the number of neurons in each layer decreasing progressively till the last one named bottleneck. In the bottleneck the number of neurons is usually lower than the number of input features;

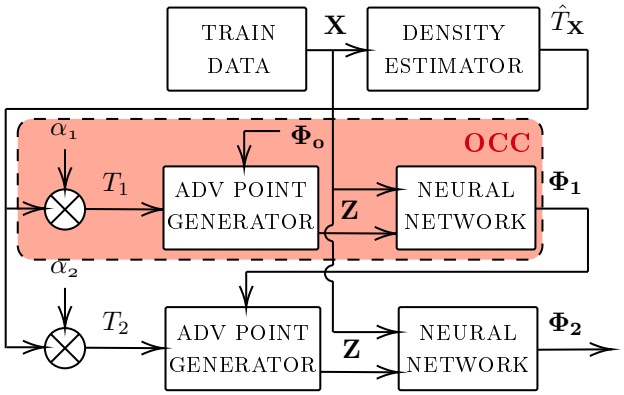


Figure 7. Block diagram of the OCCNN algorithm.

- Demapping layers, composed of one or more hidden layers where the number of neurons increases progressively.

The input and output layers have the same dimension of the feature space, and the labels during the training phase must be set equals to the input data point. With this structure, the data are mapped in lower-dimensional feature space (with dimension equal to the number of neurons present in the bottleneck layer), and then reconstructed through the demapping layers minimizing the error with respect to the input data. After the training phase, if the new data point belongs to the normal class, the reconstruction error is expected to be low; otherwise, if the data refers to an anomaly, the output of the demapping layers is likely to be very different from the input.

VI. One-Class Classifier Neural Network

In this section, a novel technique for anomaly detection based on a NN is presented. In the following, we describe each step of the algorithm whose block diagram is depicted in Fig 7.

A. Density estimator

The first fundamental step in the classification chain is the training points density estimation. As proved in [44], a minimum-variance unbiased estimator of the density, λ_x, exists when the data are distributed according to a Poisson point process. Such estimator assumes the form

$$\widehat{\lambda}_\text{x} = \frac{\left(\sum_{n=1}^{N_\text{DE}} k_n\right) - 1}{\pi \sum_{n=1}^{N_\text{DE}} r_n^2} \tag{20}$$

where $\widehat{\lambda}_\text{x}$ is the estimated density of the dataset $\mathbf{X}$, r_n is the Euclidean distance between the the nth point and its k_nth neighbour, and N_DE is the number of points considered for the estimation. The variance of the estimator is

$$\mathbb{V}\{\widehat{\lambda}_\text{x}\} = \frac{\lambda_\text{x}}{\left(\sum_{n=1}^{N_\text{DE}} k_n\right) - 2} \tag{21}$$

and it is clear that it decreases as the number of points N_DE and the value of k_n (hence considering points farther away) decrease. However, when the spatial distribution of data points

deviates from Poisson some countermeasures are taken, when possible, especially to account for the finite boundaries of the normal class: i) the set of points from which evaluate the distances r_n is a subset of the training data $\mathbf{X}$, i.e., $N_{\text{DE}} < N_{\text{x}}$; ii) low values of k_n are selected. More precisely, a subset of $N_{\text{DE}} = 200$ points is chosen from the training set (when $\mathbf{X}$ has less points, all the training set is used for the density estimation), and for all these points the second nearest point is considered, hence $k_n = 2$.

B. Adversarial points generator

This block aims at generating uniformly distributed random points with density $\lambda_i = \alpha_i \widehat{\lambda}_{\text{x}}$ in a particular portion of the feature space defined by $\mathbf{\Phi}_{i-1}$ where i is the iteration index. This is fundamental to create an adversarial class of points $\mathbf{Z}$ from which the NN can learn the boundaries between the training data $\mathbf{X}$ and the adversarial ones. The adversarial points are generated in the portion of space where the training points are absent hence where the output layer activation function is greater than 0.5 (we suppose to associate label 1 to the training set points and 0 to the adversarial ones). Therefore, this block gets λ_i, and the NN weights matrix $\mathbf{\Phi}_{i-1}$ that represents the network state after the previous training iteration and define the actual estimated boundaries. At the first iteration, $i = 1$, $\mathbf{\Phi}_0$ is the null matrix; this means that the adversarial points are generated in all the feature space (including the area covered by the training points).

C. Neural network

A two hidden layers feed-forward NN with L neurons in each layer is trained to find the boundaries between the normal class data and the adversarial ones [16]. All the layers are fully connected, and the activation functions are rectified linear unit (ReLU) for the hidden layers and softmax for the output layer. The network is trained with all the training set points $\mathbf{X}$ for N_{e} epochs (iteration of the stochastic gradient descent algorithm) with a learning rate equal to ρ. At the end of the training phase, the network provides the weight matrix $\mathbf{\Phi}_i$ that defines the boundaries estimated at the ith iteration of the OCCNN algorithm used in the next iteration by the adversarial points generator block to draw points outside the boundaries identified. After the second iteration, the boundaries estimated by the NN are the ones used for anomaly detection.

A detailed description of the hyper-parameters setup is discussed in [35].

D. OCCNN^2

The OCCNN performance is rather sensitive to the density estimation; therefore, depending on the data set, its ability to detect anomalies may be dominated by this first step [35]. Moreover, the Pollard's estimator (20) may exhibit accuracy degradation when the data set points distribution deviates from Poisson. Based on these considerations, in this work we propose OCCNN^2 an anomaly detector that share the same strategy of OCCNN but where the first NN of the classification chain (the red box in Fig. 7) is replaced with another OCC to have a rough estimate of the normal class boundaries. As shown in Section VIII, a good choice for this step is to use an ANN.

Table II
COMPUTATIONAL COMPLEXITY AND PROCESSING TIME OF THE ALGORITHMS

Algorithm	Complexity	Time [s]
SSI	$\mathcal{O}(N_{\text{a}} n_{\text{s}} (k(li)^2 + k'(li)^3 + 2n_{\text{s}}^3))$	12 334.29
PCA	$\mathcal{O}(N_{\text{x}} s^2 + s^3)$	3.55
KPCA	$\mathcal{O}(N_{\text{x}}^2 + 2N_{\text{x}}^3)$	8.71
GMM	$\mathcal{O}(N_{\text{x}} \mathcal{M}^2)$	2.50
ANN	$\mathcal{O}(N_{\text{x}} N_{\text{e}} (L_1 L_2 + L_2 L_3 + L_3 L_4 + L_4 L_5))$	26.26
OCCNN	$\mathcal{O}((N_{\text{x1}} + N_{\text{x2}}) N_{\text{e}} (L_1 L_2 + L_2 L_3 + L_3 L_4))$	135.65
OCCNN^2	$\mathcal{O}(N_{\text{x}} N_e (L_1 L_2 + L_2 L_3 + L_3 L_4 + L_4 L_5) + N_{x2} N_{\text{e}} (L_6 L_7 + L_7 L_8 + L_8 L_9))$	54.86

VII. ALGORITHMIC COMPLEXITY AND PROCESSING TIME

In this section, we discuss the computational complexity of the algorithms adopted and the corresponding processing time. The most computationally expensive blocks are SSI and the anomaly detectors during the training phase: PCA, KPCA, GMM, ANN, OCCNN, OCCNN^2. Their complexity is evaluated in the following.

- SSI requires the SVD of matrix $\mathbf{T}_{1|i}$ (3), of size $li \times li$, and eigenvalues decomposition of matrix $\mathbf{A}$ (9), of size $n \times n$. SVD complexity is $\mathcal{O}(k(li)^2 + k'(li)^3)$ (where k and k' are constants which are 4 and 22, respectively, for the R-SVD algorithm) [45]; the complexity of eigenvalues decomposition is $\mathcal{O}(n^3)$. All these procedures are applied for each system order n considered in the range $n \in [2, 160]$ (with step 2, for a total of $n_{\text{s}} = 80$), and repeated for all the N_{a} acquisitions. Thus, the overall complexity is $\mathcal{O}(N_{\text{a}} \sum_n ((k(li)^2 + k'(li)^3) + n^3))$, and can be rewritten as $\mathcal{O}(N_{\text{a}} n_{\text{s}} (k(li)^2 + k'(li)^3 + 2n_{\text{s}}^3))$.
- PCA needs the evaluation of the covariance matrix of the training points with complexity $\mathcal{O}(N_{\text{x}} s^2)$, and its eigenvalues decomposition, $\mathcal{O}(s^3)$, where s is the number of features and N_{x} the number of training points. The total complexity of PCA results to be $\mathcal{O}(N_{\text{x}} s^2 + s^3)$.
- KPCA requires the evaluation of the kernel with complexity $\mathcal{O}(N_{\text{x}}^2)$, its covariance matrix evaluation, $\mathcal{O}(N_{\text{x}}^3)$, and its eigenvalues decomposition, $\mathcal{O}(N_{\text{x}}^3)$. The overall complexity of KPCA is $\mathcal{O}(N_{\text{x}}^2 + 2N_{\text{x}}^3)$.
- GMM based on expectation-maximization (EM) has complexity $\mathcal{O}(N_{\text{x}} \mathcal{M}^2)$ where $\mathcal{M}$ is the order of the model, according to [46].
- ANN is a particular kind of NN with 5 layers of L_1, L_2, L_3, L_4 and L_5 neurons each, respectively. The computational cost for training the ANN using the backpropagation algorithm is $\mathcal{O}(N_{\text{x}} N_{\text{e}} (L_1 L_2 + L_2 L_3 + L_3 L_4 + L_4 L_5))$, where N_{e} is the number of epochs.
- OCCNN requires a two-step training with a different number of training points in each phase, N_{x1} and N_{x2} respectively, with $N_{\text{x1}} > N_{\text{x2}}$ as explained in Section VI.

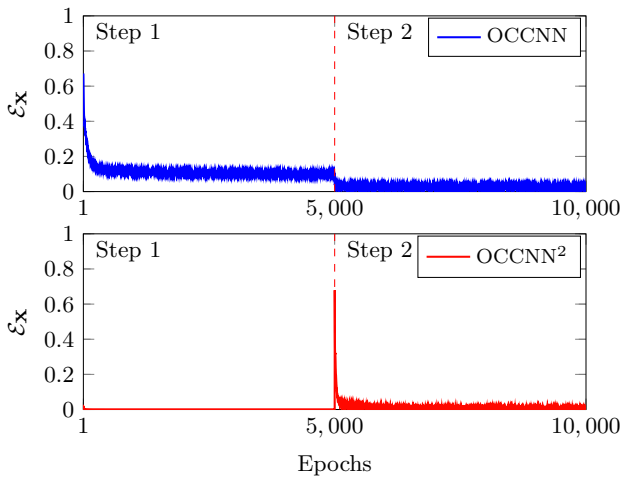


Figure 8. Error function evolution over the epochs during training.

Considering a NN of 4 layers with L_1, L_2, L_3, and L_4 neurons, respectively, the complexity is $\mathcal{O}((N_{x1} + N_{x2})N_e(L_1L_2 + L_2L_3 + L_3L_4))$.

- OCCNN^2 exploits the combination of ANN and OCCNN. Therefore, the overall complexity is $\mathcal{O}(N_xN_e(L_1L_2 + L_2L_3 + L_3L_4 + L_4L_5) + N_{x2}N_e(L_6L_7 + L_7L_8 + L_8L_9))$ where L_1, L_2, L_3, L_4 and L_5 are the neurons in the ANN layers, and L_6, L_7, L_8, and L_9 are the ones in the OCCNN layers, respectively.

In Table II a summary of the complexity of all the algorithms is reported, along with the processing times experienced on a computer with 2,4 GHz Intel Core i5 processor and 2133 MHz LPDDR3 RAM.

VIII. Numerical Results

In this section, the proposed algorithms are applied to the *Z-24* bridge data set to detect anomaly based on the fundamental frequencies estimation [7], [8], [31]. The performance is evaluated through the following metrics considering only the test set:

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

$$\text{Precision} = \frac{T_P}{T_P + F_P}$$

$$\text{Recall} = \frac{T_P}{T_P + F_N}$$

$$\text{F}_1 \text{ score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

where T_P, T_N, F_P, and F_N, represent respectively true positive, true negative, false positive, and false negative predictions. Such indicators are obtained comparing the actual labels $[\zeta^{(1)}, \ldots, \zeta^{(N_a)}]$, with those predicted by the OCC $[\widehat{\zeta}^{(1)}, \ldots, \widehat{\zeta}^{(N_a)}]$. In this application, labels are 0 for normal condition and 1 for anomaly condition, respectively. Therefore,

$$T_P = \sum_{a \in \mathcal{I}_u} \mathbb{1}\{\zeta^{(a)}, \widehat{\zeta}^{(a)}\} \qquad T_N = \sum_{a \in \mathcal{I}_y} \mathbb{1}\{\zeta^{(a)}, \widehat{\zeta}^{(a)}\}$$

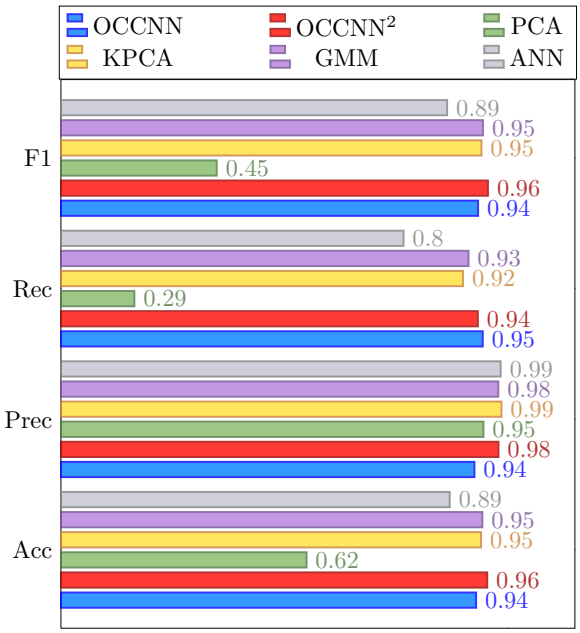


Figure 9. Comparison of the classification algorithms in terms of F_1 score, recall, precision, and accuracy.

$F_N = N_u - T_N$, and $F_P = N_y - T_P$. In the case of unbalanced classes in the test set, the F_1 score represents a more reliable metric to evaluate the performance with respect to the accuracy.

In these numerical results, the dataset is divided into a training set and a test set. For this work, we decided not to use a validation phase because it is not needed in OCCNN and OCCNN^2 as there are no hyperparameters to set. For the other algorithms, we set their parameters to ensure the maximum accuracy on the test set. With this methodology, we slightly overestimate the accuracy of PCA, KPCA, and GMM. Despite this setup favor PCA, KPCA and GMM, in the following, it is shown that the proposed solution, OCCNN^2, overcome their performance.

The feature space has dimension $D = 2$, and unless otherwise specified the three data sets used for training, test in normal condition, and damaged condition, have cardinality $N_x = 2399$, $N_y = 854$, and $N_u = 854$, respectively. For PCA, the number of components selected is $P = 1$. For KPCA, after several tests the values of P and γ that ensure the minimum reconstruction error are $P = 3$ and $\gamma = 8$. For GMM the order of the model that maximize performance is $\mathcal{M} = 10$. Regarding the ANN we adopted a fully connected network with 5 layers of, respectively, 100, 50, 1, 50 and 100 neurons, with ReLU activation functions. Note that, the same ANN is also used for the first step of the OCCNN^2 algorithm. The parameters of the OCCNN and OCCNN^2 are set accordingly to [35], resulting in $\alpha_1 = 0.3$, $\alpha_2 = 0.8$, and are largely independent on the spatial distribution of the feature points; this allows to presume that such OCCs can work on different structures and bridges. The NN has 2 hidden layers with $L = 50$ neurons each. All the NNs are trained for a number of epochs $N_e = 5000$ with a learning rate $\rho = 0.05$. The error

Table III
ALGORITHM, PARAMETER, VALUE, AND DESCRIPTION

Algorithm	Parameter	Value	Description
SSI	n	$[2, 160]$	System order
Clustering	K	$[2, 6]$	Number of clusters
Tracking	N_t	200	Window size
Tracking	B_f	0.4 Hz	Frequency bin bandwidth
Tracking	ϕ_f	0.01 Hz	Kernel width
Tracking	ϵ	0.7	Innovation
PCA	P	1	Number of selected components
KPCA	γ	8	Gaussian kernel width
KPCA	P	3	Number of selected components
GMM	$\mathcal{M}$	10	Gaussian mixture model order
ANN	layers	5	Number of hidden layers
ANN	neurons	$\{100, 50, 1, 50, 100\}$	Number of neurons in the hidden layers
NN	layers	2	Number of hidden layers
NN	neurons	$\{50, 50\}$	Number of neurons in the hidden layers
Training	N_e	5000	Number of training epochs
Training	ρ	0.05	Learning rate
OCCNN	α_1	0.3	Density factor first iteration
OCCNN	α_2	0.8	Density factor second iteration
OCCNN^2	α_2	0.8	Density factor second iteration

function adopted for a training set $\mathbf{X}$ is

$$\mathcal{E}_\mathbf{X} = -\sum_{n=1}^{N_\mathrm{x}} \sum_{c=1}^{C} t_{n,c} \ln \widetilde{t}_{n,c} \tag{22}$$

where N_x is the number of points in the training set, C is the number of classes ($C = 2$), $t_{n,c} = 1$ if the nth acquisition belongs to the cth class and zero otherwise, and $\widetilde{t}_{n,c}$ is the activation function value for point n of the cth output neuron [16]. As can be seen in Fig. 8 for each step of the algorithms the networks are trained for $N_\mathrm{e} = 5000$ epochs, quite enough for the error function to reach the minimum. A target false alarm (FA) rate on the training set is fixed equal to 0.01, from which a threshold is selected for each algorithm to guarantee such constraint. In Table III the principal hyperparameters are summarized and described.

A. Performance comparison

The performance comparison of the algorithms is reported in Fig. 9. As we can see, considering the F_1 score OCCNN outperforms PCA and ANN but is overtaken by KPCA and GMM. This happens because the non-uniform data distribution of the training set influences Pollard's estimator which overestimates the density. As a result, there are too many adversarial points generated, which leads to an underestimation of the normal class boundary. The OCCNN^2 overcomes this limitation and shows the best F_1 score and accuracy. This

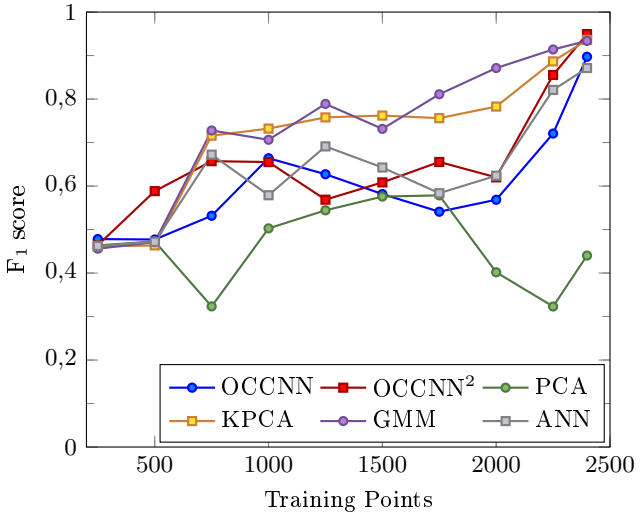


Figure 10. F_1 score varying the number of points, N_x, used for training, with $N_\mathrm{y} = 854$ and $N_\mathrm{u} = 854$.

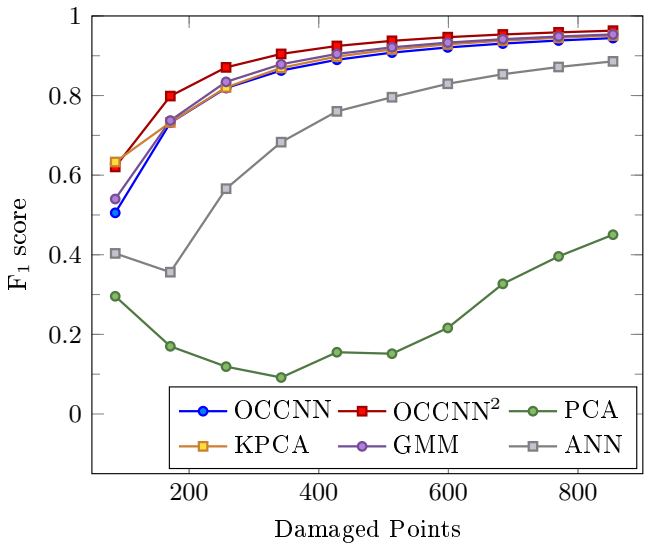


Figure 11. F_1 score varying the number of points during damage, N_u, used to test the algorithms, with $N_\mathrm{x} = 2399$ and $N_\mathrm{y} = 854$.

represents a remarkable result considering the low number of hyper-parameters to be tuned.

B. Impact of the training set and responsiveness

In Fig. 10 and Fig. 11, the F_1 score dependence on the number of points used to train the algorithms, N_x, and the number of anomaly points, N_u, used to test the algorithms, are reported.

The number of training points, N_x, in Fig. 10 varies from 250 to 2399 with steps of 250 points (except the last step equal to 149). This plot shows how many points are necessary for the correct estimation of the normal class boundaries. In these results, all the numbers of points used for test are kept constant, i.e., $N_\mathrm{y} = N_\mathrm{u} = 854$ to maintain the same boundaries between the classes. As can be seen in the figure,

all the algorithms require an high number of data points corresponding to long-lasting monitoring (almost one year [7]) to provide the highest F_1 score. The algorithms that show a sharper increase in their performance, hence requiring less training points, are KPCA and GMM.

In Fig. 11, the number of training points is kept constant, $N_x = 2399$, and the number of anomaly test points, N_u, varies from 86 to 854 with 10 points steps. The plot shows that OCCNN^2 exhibit the best responsiveness, i.e., it requires the lowest number of damaged points to achieve higher values of the F_1 score.

IX. Conclusion

In this paper, we presented a complete processing chain capable of monitoring the structural health of bridges by means of output-only system identification based on accelerometric data. The proposed strategy is capable of automatically process the OMA results to detect the presence of damages in a bridge. In particular, we found that combining clustering of the stabilization diagram with a proper time-domain tracking algorithm for the extracted natural frequencies, we obtain a feature space capable of revealing the presence of anomalies when processed by ML tools. We then proposed OCCNN and OCCNN^2 algorithms, two novel one-class classifiers for anomaly detection, which exhibit very good performance. We applied the proposed framework on data collected from the *Z-24* bridge to assess its effectiveness and perform comparisons with several anomaly detection techniques: PCA, KPCA, GMM, and ANN. The OCCNN solution presents considerable accuracy and an F_1 score higher than conventional algorithms without the need to set critical parameters, while OCCNN^2, with similar complexity, provides the best performance in term of F_1 score, accuracy, and responsiveness.

Acknowledgement

The authors wish to thank Andrea Benedetti, Alessandro Marzani, and Mirco Tarozzi for the helpful suggestions.

References

[1] R. Ferrari, D. Froio, E. Chatzi, F. Pioldi, and E. Rizzi, "Experimental and numerical investigations for the structural characterization of a historic RC arch bridge," in *Proc. Int. Conf. on Comp. Methods in Structural Dynamics and Earthquake Eng.*, vol. 1, Athens, Greece, May 2015, pp. 2337–2353.

[2] A. Benedetti, M. Tarozzi, G. Pignagnoli, and C. Martinelli, "Dynamic investigation and short-monitoring of an historic multi-span masonry arch bridge," in *Proc. Int. Conf. on Arch Bridges*, vol. 11, Porto, Portugal, Oct. 2019, pp. 831–839.

[3] A. Benedetti, G. Pignagnoli, and M. Tarozzi, "Damage identification of cracked reinforced concrete beams through frequency shift," *Materials and Structures*, vol. 51, pp. 1–15, Oct. 2018.

[4] A. Benedetti, C. Colla, G. Pignagnoli, and M. Tarozzi, "Static and dynamic investigation of the taro masonry bridge in parma, italy," in *Proc. Int. Conf. on Struct. Analysis of Hist. Const.*, vol. 18, Cusco, Perù, Sep. 2019, pp. 2264–2272.

[5] G. D. Roeck, "The state-of-the-art of damage detection by vibration monitoring: the SIMCES experience," *J. of Structural Control*, vol. 10, no. 2, pp. 127–134, May 2003.

[6] K. Worden, C. Farrar, J. Haywood, and M. Todd, "A review of nonlinear dynamics applications to structural health monitoring," *Structural Control and Health Monitoring*, vol. 15, no. 4, pp. 540–567, Jul. 2008.

[7] G. Fabbrocino and C. Rainieri, *Operational modal analysis of civil engineering structures*. New York: Springer-verlag, May 2014.

[8] M. Silva, A. Santos, E. Figueiredo, R. Santos, C. Sales, and J. Costa, "A novel unsupervised approach based on a genetic algorithm for structural damage detection in bridges," *Eng. Applications of Artificial Intelligence*, vol. 52, pp. 168–180, Jun. 2016.

[9] A. Santos, R. Santos, M. Silva, E. Figueiredo, C. Sales, and J. C. W. A. Costa, "A global expectation–maximization approach based on memetic algorithm for vibration-based structural damage detection," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 4, pp. 661 – 670, 2017.

[10] J. Nichols, S. Trickey, M. Todd, and L. Virgin, "Structural health monitoring through chaotic interrogation," *Meccanica*, vol. 38, no. 2, p. 239–250, Mar. 2003.

[11] R. M. Azzara, G. D. Roeck, M. Girardi, C. Padovani, D. Pellegrini, and E. Reynders, "The influence of environmental parameters on the dynamic behaviour of the San Frediano bell tower in Lucca," *Engineering Structures*, vol. 156, pp. 175 – 187, 2018.

[12] C. Farrar, G. Park, D. Allen, and M. Todd, "Sensor network paradigms for structural health monitoring," *Structural Control and Health Monitoring*, vol. 13, no. 1, pp. 210–225, Dec. 2006.

[13] E. Flynn, M. Todd, P. Wilcox, B. Drinkwater, and A. Croxford, "Maximum-likelihood estimation of damage location in guided-wave structural health monitoring," *Proc. of the Royal Society A: Math., Phys. and Eng. Sciences*, vol. 467, no. 2133, pp. 2575–2596, Apr. 2011.

[14] S. Taylor, K. Farinholt, E. Flynn, E. Figueiredo, D. L. Mascarenas, E. A. Moro, G. Park, M. Todd, and C. Farrar, "A mobile-agent-based wireless sensing network for structural monitoring applications," *Measurement Science and Technology*, vol. 20, no. 4, pp. 1–14, Jan. 2009.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, Aug. 2006.

[16] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine Learning Refined*. Cambridge University Press, 2016.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[18] Z-24 bridge data. [Online]. Available: https://bwk.kuleuven.be/bwm/z24

[19] E. Reynders and G. D. Roeck, "Continuous vibration monitoring and progressive damage testing on the z 24 bridge," *Encyclopedia of structural health monitoring*, vol. 10, pp. 127–134, May 2009.

[20] E. Reynders, J. Houbrechts, and G. D. Roeck", "Fully automated (operational) modal analysis," *Mechanical Systems and Signal Processing*, vol. 29, pp. 228–250, May 2012.

[21] (2001) FBA11 datasheet. [Online]. Available: ftp://aplica.at/Manuals/Sensors/FBA11.pdf

[22] (1998) ADC488 datasheet. [Online]. Available: https://www.artisantg.com/info/IOtech_ADC488_Manual.pdf

[23] J. S. Bendat and A. G. Piersol, "Engineering applications of correlation and spectral analysis," *New York, Wiley-Interscience*, 1980.

[24] R. Wei-Xin and Z. Zhou-Hong, "Output-only modal parameter identification of civil engineering structures," *Structural Engineering and Mechanics*, vol. 17, Mar. 2004.

[25] E. Reynders, G. Wursten, and G. D. Roeck, "Output-only structural health monitoring in changing environmental conditions by means of nonlinear system identification," *Structural Health Monitoring*, vol. 13, no. 1, pp. 82–93, 2014.

[26] C. Wu, H. Liu, X. Qin, and J. Wang, "Stabilization diagrams to distinguish physical modes and spurious modes for structural parameter identification," *Journal of Vibroeng.*, vol. 19, no. 4, pp. 2777–2794, Jun. 2017.

[27] D. Brigante, C. Rainieri, and G. Fabbrocino, "The role of the modal assurance criterion in the interpretation and validation of models for seismic analysis of architectural complexes," in *Proc. Int. Conf. on Structural Dynamics (Eurodin)*, vol. 199, Rome, Italy, Sep. 2017, pp. 3404–3409.

[28] A. Cabboi, F. Magalhães, C. Gentile, and Á. Cunha, "Automated modal identification and tracking: Application to an iron arch bridge," *Structural Control and Health Monitoring*, vol. 24, no. 1, p. e1854, Feb. 2017.

[29] M. Pastor, M. Binda, and T. Harčarik, "Modal assurance criterion," *Procedia Engineering*, vol. 48, pp. 543–548, 2012.

[30] E. P. Carden and J. M. W. Brownjohn, "Fuzzy clustering of stability diagrams for vibration-based structural health monitoring," *Computer-Aided Civil and Infrastructure Eng.*, vol. 23, no. 5, pp. 360–372, May 2008.

[31] M. Silva, A. Santos, R. Santos, E. Figueiredo, C. Sales, and J. C. Costa, "Agglomerative concentric hypersphere clustering applied to structural damage detection," *Mechanical Systems and Signal Processing*, vol. 92, pp. 196–212, Feb. 2017.

[32] A. Santos, M. Silva, C. Sales, J. Costa, and E. Figueiredo, "Applicability of linear and nonlinear principal component analysis for damage detection," in *Proc. IEEE Int. Instr. and Meas. Tech. Conf. (I2MTC)*, Pisa, Italy, May 2015, pp. 869–874.

[33] E. Favarelli, E. Testi, L. Pucci, and A. Giorgetti, "Anomaly detection using wifi signal of opportunity," in *Proc. IEEE Int. Conf. on Signal Proc. and Comm. Sys. (ICSPCS)*, Surfers Paradise, Gold Coast, Australia, Dec. 2019, pp. 1–7.

[34] E. Testi, E. Favarelli, and A. Giorgetti, "Machine learning for user traffic classification in wireless systems," in *Proc. Europ. Signal Proc. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 2040–2044.

[35] E. Favarelli, E. Testi, and A. Giorgetti, "One class classifier neural network for anomaly detection in low dimensional feature spaces," in *Proc. IEEE Int. Conf. on Signal Proc. and Comm. Sys. (ICSPCS)*, Surfers Paradise, Gold Coast, Australia, Dec. 2019, pp. 1–7.

[36] E. Testi, E. Favarelli, L. Pucci, and A. Giorgetti, "Machine learning for wireless network topology inference," in *Proc. IEEE Int. Conf. on Signal Proc. and Comm. Sys. (ICSPCS)*, Surfers Paradise, Gold Coast, Australia, Dec. 2019, pp. 1–7.

[37] P. Perera and V. M. Patel, "Learning deep features for one-class classification," *IEEE Trans. Image Process*, vol. 28, no. 11, pp. 5450–5463, Nov. 2019.

[38] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *CoRR*, vol. abs/1802.06360, Aug. 2018.

[39] P. Schlachter and B. Yang, "Active learning for one-class classification using two one-class classifiers," in *Proc. Europ. Signal Proc. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 1197–1201.

[40] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interd. Reviews: Comp. Stat.*, vol. 2, no. 4, pp. 433–459, 2010.

[41] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. Int. conf. on artificial neural networks*, vol. 1327, no. 6. Lausanne, Switzerland: Springer, Oct. 1997, pp. 583–588.

[42] B. Schölkopf, A. Smola, E. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comp.*, vol. 10, pp. 1299–1319, Jul. 1998.

[43] A. Santos, E. Figueiredo, M. Silva, R. Santos, C. Sales, and J. C. W. A. Costa, "Genetic-based EM algorithm to improve the robustness of gaussian mixture models for damage detection in bridges," *Structural Control and Health Monitoring*, vol. 24, no. 3, p. e1886, May 2017.

[44] J. H. Pollard, "On distance estimators of density in randomly distributed forests," *Biometrics*, vol. 27, no. 4, pp. 991–1002, Dec. 1971.

[45] G. H. Golub and C. F. V. Loan, *Matrix computations*. JHU press, 2012, vol. 3.

[46] J. J. Verbeek, N. Vlassis, and B. Kröse, "Efficient greedy learning of gaussian mixture models," *Neural computation*, vol. 15, no. 2, pp. 469–485, 2003.