Deep learning and structural health monitoring: A TFT-based approach for anomaly detection in masonry towers

Fabrizio Falchi[*], Maria Girardi[*], Gianmarco Gurioli[†], Nicola Messina[*]
Cristina Padovani[*], Daniele Pellegrini[*]

December 15, 2023

Abstract

Detecting anomalies in the vibrational features of age-old buildings is crucial within the Structural Health Monitoring (SHM) framework. The SHM techniques can leverage information from onsite measurements and environmental sources to identify the dynamic properties (such as the frequencies) of the monitored structure, searching for possible deviations or unusual behavior over time. In this paper, the Temporal Fusion Transformer (TFT) network, a deep learning algorithm initially designed for multi-horizon time series forecasting and tested on electricity, traffic, retail, and volatility problems, is applied to SHM. The TFT approach is adopted to investigate the behavior of the Guinigi Tower located in Lucca (Italy) and subjected to a long-term dynamic monitoring campaign. The TFT network is trained on the tower's experimental frequencies enriched with other environmental parameters. The transformer is then employed to predict the vibrational features (natural frequencies, root mean squares values of the velocity time series) and detect possible anomalies or unexpected events by inspecting how much the actual frequencies deviate from the predicted ones. The TFT technique is used to detect the effects of the Viareggio earthquake that occurred on 6 February 2022, and the structural damage induced by three simulated damage scenarios.

[*]Institute of Information Science and Technologies "A. Faedo", ISTI-CNR, Via Moruzzi 1, 56124, Pisa (Italy).

[†]Dipartimento di Matematica e Informatica "Ulisse Dini", University of Florence, Firenze (Italy).
Corresponding author: Maria Girardi

1 Introduction

Maintaining heritage structures requires paying attention to possible damages caused by natural calamities, environmental degradation, seismic events, aging, and atmospheric changes. Such damages must be promptly detected to plan timely restoration and strengthening operations. These activities are the final aim of Structural Health Monitoring (SHM), an integrated approach to controlling and maintaining historical buildings via long-term, continuous, dynamic monitoring. The environmental excitations of natural and anthropic origin (e.g., earthquakes, wind, traffic, machinery, moving crowds) acting on civil and historic constructions induce vibrations that can be measured, recorded, stored, and analyzed to get information on their structural behavior. Using ambient vibrations to characterize the dynamic properties of constructions dates back to the 1970s [1] and spread out in the last decades, thanks to the availability of high-sensitivity instruments that measure very low levels of accelerations and velocities. Monitoring ambient vibrations suits heritage buildings since it is not invasive and allows measurements without artificial excitations. Data recorded by the velocimeters and accelerometers installed on the building in operating conditions are processed using suitable numerical procedures to determine its dynamic properties, such as frequencies, damping ratios and mode shapes. This approach is known as Operational Modal Analysis (OMA) [2]. Tracking the variation of frequencies over time, assessing environmental effects, and analyzing changes and anomalies in the dynamic behavior of a building are essential aspects of SHM. Applying SHM to a building can provide helpful information on the vibration sources, the variation of dynamic properties, and the structural "safety" state of the structure, making damage detection possible; in fact, changes in the dynamical properties over time can represent structural damage indicators [3], [4], [5].

Literature on anomaly detection techniques is relatively recent and encompasses model-based approaches exploiting Finite Element (FE) models of the monitored structure and data-driven approaches, using data collected during the monitoring campaigns. Model-based approaches use experimental frequencies and mode shapes of a structure before and after the damage to predict damage location and quantification in its FE counterpart within the framework of inverse problems. Relevant applications are described in [6], [7] and [8]. Data-driven methods relying on regression and output-only techniques, like the principal component analysis of the frequencies time-histories, are adopted in [9], [5], [10].

The availability of large datasets recorded during long-term monitoring campaigns makes it possible to investigate the rise of anomalies by comparing the frequencies of the recent time intervals with the ones of the past. In particular, the

frequencies of future days can be considered as variables of interest to be predicted at multiple future time steps by accessing the already-known series of data in the past. From this data-driven perspective, the collected data could be the training dataset of artificial intelligence algorithms to extract features and predict the frequencies at future time instants. This approach enters the perimeter of machine learning (see, e.g., [11, 12, 13, 14, 15, 16, 17, 18]) and, more recently, of Deep Learning (DL) techniques for time series forecasting (see, e.g., [19, 20]) and anomaly detection (see e.g. [21, 22, 23, 24]), encountering several different architectures, such as convolutional neural networks, capsule neural networks, recurrent neural networks, and, more recently, transformer networks (see the review [25]).

The recorded time series coming from sensors on the buildings more naturally fit the paradigm of multi-horizon time series forecasting, where a heterogeneous mix of input data is considered, with no prior information about the relationship between them and the target variables (e.g., the frequencies) to predict. In order to learn such a correlation, deep state-space models for time series forecasting (see, e.g., [26, 27, 28]) were joined by Recurrent Neural Networks (RNNs)[29, 30, 31, 32, 33, 34, 35, 36, 37], including auto-regressive models and attention-based methods [38, 39]. In particular, RNNs extract patterns from temporal sequences and cope with the temporal dependencies between temporally distant inputs, thanks to the short-term memory cabled into their recurrent connections. Such networks are expected to model a larger class of processes versus feed-forward neural networks and seem to filter noise more effectively (see, e.g., [40]). We recall that recent DL methods are generally based on two different categories: iterative schemes (see, e.g., [39, 27, 36]) and direct methods (see, e.g., [38, 37]). The first class of methods uses one-step-ahead prediction models and multi-step predictions by recursively feeding predictions into future points but needs to assume that all the exogenous inputs are known into the future, while time series forecasting usually deals with a certain number of inputs changing with time. Direct strategies, in turn, are founded on multiple predefined horizons at each time step, relying on sequence-to-sequence models. However, both approaches depend on many nonlinear activations that make it difficult to understand how the target prediction is effectively learned from the input, at the risk of hindering debug operations. In this respect, the transformer architecture introduced in [41], through the attention mechanisms, is able to select relevant partitions of inputs and the pairwise relationships between input tokens that are mostly required to solve the downstream task, yielding more interpretable models. Initially, the transformer showed impressive results on natural language tasks, thanks to their ability to capture long-range dependencies and therefore to handle large contexts, unlike RNNs. However, in time-series processing literature, many important results are still

obtained by employing RNNs, which are less data-hungry and capable of handling short time series better. For example, [42] proposes modifying the RNN architecture where the hidden layer is partitioned into separate modules, each processing inputs at its temporal granularity, making computations only at its prescribed clock rate. In addition, [43] presents a novel RNN model processing inputs sampled at asynchronous times. To improve interpretability, also post-hoc explanation/inherently interpretable methods (like [44, 45, 46, 47, 48]) have been exploited, yet not able to capture the time ordering of the inputs.

DL approaches have been used in several fields, including road/internet traffic, retail and health forecasting, but their application in the SHM of heritage structures for anomaly detection is still relatively recent and limited ([49, 50, 51, 52, 53, 54, 55]). For instance, in [53], data coming from a finite element analysis is used to train a DL Convolutional Neural Network (CNN) classifier, while combining CNN with Long Short Term Memory (LSTM) networks could help to learn long-term dependencies ([49]). Also, the paper [54] relies on CNNs for the estimation of the nodal loads of a bridge structure and the spotting of damage information. Outliers detection via DL-enhanced principal component analysis is performed in [50], moving towards the online damage localization of [52], where simulation-based classification strategies relying on model order reduction and fully convolutional networks are exploited. However, almost all these papers focus on the safety and maintenance of bridges and modern and slender towers (such as the television tower in [51]), with no reference to transformer networks applied to the SHM of age-old masonry towers, widely spread in the Italian architectural heritage.

For these reasons, in this work, we will employ a state-of-the-art transformer model – the Temporal Fusion Transformer (TFT) an attention-based deep neural network architecture for multi-horizon forecasting introduced in [56] – to understand structure dynamics and perform effective anomaly detection. First, the network could contrast the already mentioned black-box characteristic of RNNs thanks to specific components that facilitate interpretability. In addition, it is coupled with a series of gating mechanisms to minimize the contribution of the less relevant inputs. Furthermore, the TFT implements a sequence-to-sequence layer to process locally known and observed inputs and a temporal self-attention decoder for the long-term learning of dependencies within the dataset. Thanks to such a structure, TFT has been shown to help identify globally relevant variables for the prediction problem, persistent temporal patterns, and significant events.

TFT has been experimented on electricity, traffic, retail, and volatility problems, and the only application to SHM is described in [57], where TFT has been preliminary used to detect anomalies in the large dataset recorded during a long-term monitoring

campaign conducted on the San Frediano bell tower in Lucca (Italy).

This paper more deeply inspects the application of TFT to SHM, considering the Guinigi Tower located in Lucca, which is often the target of long-term monitoring campaigns. The TFT network is trained on the hourly natural frequencies calculated by processing the velocities recorded by high-sensitivity sensors installed on the tower, enriched with other environmental parameters measured in nearby sites.

The frequencies are computed by a Python procedure that uses the Stochastic Subspace Identification (SSI) method [58] implemented in different open-source and commercial software (e.g. [59, 60]), equipped with a fully-automatic strategy based on an unsupervised hierarchical clustering scheme [61], whose aim is to get rid of actions by the user as in the frameworks of [62] and, more recently, [63, 64].

The transformer is then employed to predict the vibrational features of the Guinigi Tower (natural frequencies, root mean squares values as a measure of the "energy" of the tower's response) and detect possible anomalies or unexpected events by inspecting how much the actual response deviates from the predicted one. The case study presented in the paper is deeply investigated. The TFT technique is used to detect the effects of the Viareggio earthquake that occurred on 6 February 2022 and the structural damage induced by three simulated damage scenarios. In addition, the paper investigates the predictive potential of TFT as an alternative tool to compute the natural frequencies, besides the traditional approaches [2, 58].

The paper is organized as follows. The automatic DL strategy for SHM and anomaly detection using TFT is described in Section 2. The results of the dynamic monitoring campaign on the Guinigi Tower are reported in Section 3, where the dynamic behavior of the tower is recalled along with a description of the dataset to be used by the TFT network. Numerical tests concerning anomaly detection and the use of the TFT network for predicting the frequencies of the tower under standard conditions are considered in Section 4, also including a comparison with the predictions by an Auto-Regression with eXogenous inputs (ARX) model. Finally, some concluding remarks are sketched in Section 5.

2 An automatic deep learning strategy for SHM

This section describes a new approach to perform SHM and anomaly detection via DL techniques; this approach fits within the unsupervised learning framework. From a macroscopic viewpoint, the idea is to learn the normal dynamics of a structure (like its natural frequencies under standard conditions) directly from the past time series of velocity/acceleration and environmental data, and infer the future dynamics at required time instants.

To this aim, we train the TFT network in [56] on a portion of the available dataset, splitting the remaining data into testing and validation sets. As usual, the validation set is considered to tune the hyperparameters of the network, while the test set serves for previsions and performance evaluations. The more significant deviations from the learned structure's dynamics are considered anomalies.

Such a strategy relies on two steps that will be deepened in the following subsections: the suitable assembling of a representing dataset, essential for an effective learning phase, and the proper use of the TFT network for anomaly detection.

2.1 Assembling a representative reference dataset

Generating an appropriate dataset is a challenging starting point in itself, which can have a crucial influence on the success of the subsequent learning phase of the TFT model. This dataset D is a matrix of $m \times n$ real values, m representing the total number of samples and n the total number of variables. In the h-th row ($h \in \{1, ..., m\}$) of the dataset, the first two columns list the reference date and the hour slot in the monitoring period, the last n_1 entries are filled with the values of environmental parameters measured from weather stations and other relevant data (e.g. markers for working and festival days, etc.), while the remaining $n_2 = n - n_1 - 2$ columns report the values of the natural frequencies for such date and hour slot. The natural frequencies are calculated from the velocities time series recorded by the sensors installed on the structure, using the numerical method described below. In the case study presented in Section 3, along with the frequencies, the dataset contains the values of the root mean squares of the velocities at each hour slot of each monitored day.

The dynamic identification of the structure in operating conditions is conducted within the framework of Operational Modal Analysis (OMA) [2], exploiting the structure's vibrations induced by environmental excitations (like wind, pedestrians, and traffic).

Among the broad domains of non-parametric (frequency-domain) and parametric (time-domain) identification methods, we align with the SSI framework [65] of the time-domain strategies, motivated by their more effective impact on civil engineering applications [66]. In particular, the reference-based covariance-driven stochastic realization scheme in [58] is employed, whose discrete-time stochastic state-space formulation is briefly recalled below.

Given the $r \times r$ state matrix $\mathbf{A}$, the $s \times r$ influence matrix $\mathbf{C}$ (r, $s \in \mathbb{N}$), the state vector at the k-th time $\mathbf{x}_k \in \mathbb{R}^r$ and the vector of observations $\mathbf{y}_k \in \mathbb{R}^s$, the

stochastic state-space model (of order r) is defined as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, \end{aligned} \tag{1}$$

where $\mathbf{w}_k \in \mathbb{R}^r$ is the processes noise due to disturbance/modeling inaccuracies and $\mathbf{v}_k \in \mathbb{R}^s$ is the measurement noise caused by sensor inaccuracy; $\mathbf{w}_k$ and $\mathbf{v}_k$ are unmeasurable and modeled as zero mean, stationary, white noise random processes. The state matrix $\mathbf{A}$ describes the dynamics of the system, whose modal properties are characterized by the eigenvalues of $\mathbf{A}$. [58].

In order to perform a frequencies' estimation at a given time slot (lasting 1 hour), the method takes as input a matrix of $n_c \times n_t$ components, encoding the n_c time series of length n_t recorded by each of the n_c channels of the sensors installed on the structure (each sensor can handle one or more channels), measuring with a given sampling frequency. To avoid the hourly generation of incomplete input matrices caused by channels that skip data at some instants, isolated missing data in each time series are restored through linear interpolation, neglecting the registered time series affected by massive missing data. Pre-processing operations, like signal detrending and filtering, are then carried out.

Commercial programs devoted to dynamic identification via SSI require users to manually select the more significant frequencies in the stabilization diagram. Such a feature hinders the use of these codes when the identification task has to be integrated into cloud-based or wireless sensor network platforms for the massive acquisition of data time series (an example of these platforms is described in [67]). For this reason, we have realized a Python implementation of the covariance-driven method in [58], embedded into a suitable clustering strategy to automatically select the significant frequency values of each hour slot [61].

In practice, each of the hourly pre-processed $n_c \times n_t$ matrix is the input of the Python solver, together with the sampling frequency of the sensors and suitable options for the stabilization diagram, consisting of the maximum and minimum order method, the relative accuracy on the frequency, the damping and MAC values to verify the stability of the poles and the maximum tolerance on the damping ratio for the corresponding stable pole to be considered. The solver then displays the stabilization diagram to outline the stable poles (in frequency, damping and mode shapes), storing the list of stable poles with related damping ratios and mode shape vectors. Once this phase is completed, the automatic selection of the hourly most representative poles is carried out by adopting the agglomerative hierarchical clustering strategy ([68, 69]) proposed in [61, section 3], which does not require manual selections by the user. The algorithm first computes the pairwise Euclidean distances between pairs

of poles of the list. The resulting matrix is used to perform a linkage, returning a new matrix encoding a dendrogram (bottom-up tree) that contains the hierarchical clusters of the list of poles. The shortest distance criterion has been adopted for computing the clusters distances within the linking procedure. Then, each of the poles in the list is assigned to the corresponding cluster based on a cutoff threshold for the heights and the inconsistency coefficients of nodes in the agglomerative hierarchical cluster tree of the linkage phase. The subsequent computation of the centroids of each cluster enables detaching the resulting closest stable frequencies, which are finally included in the j-th sample of the dataset to be given as input for the TFT model, whose main features are recalled in the following subsection.

2.2 The TFT network for SHM

We here start by recalling the main features of the TFT network in [56]. Given a reference input dataset D with collected data from time $t = 0$ up to time $t = T$, let $i \in I$ be the indexes of the output variables. For each time t and each output index i, we have a target value $y_{i,t}$ (e.g., the i-th experimental frequency) to be predicted at future time instants. For $i \in I$ and $t \in [0, T]$, $\boldsymbol{\chi}_t = (\mathbf{z}_t^\top, \mathbf{x}_t^\top)^\top$ represents the vector of time-dependent inputs, split into the inputs $\mathbf{z}_t \in \mathbb{R}^{m_z}$ measured at each $t \in [0, T]$ by the sensors (the observed inputs) and the inputs $\mathbf{x}_t \in \mathbb{R}^{m_x}$ that are known without measurements (such as the date at a prescribed time t).

The inputs of the TFT architecture are the classes mentioned above of observed and known time-varying inputs, leveraging several architectural components to perform forecasting efficiently. In this respect, the depth of the network can be adaptively adjusted, making superfluous parts of the architecture inactive on the basis of the dataset and the inputs at hand. Especially with noisy/small datasets, there could be scenarios in which simpler models are beneficial. Except from such gating mechanisms, TFT also deals with selecting of the most relevant input variables at each time step, attempting to reduce the impact of noisy or unnecessary inputs. A temporal fusion decoder is then employed to learn short and long-term temporal relationships from the observed and known inputs, resulting in a sequence-to-sequence (recurrent) layer for local processing and a multi-head attention block to capture long-term dependencies.

TFT can simultaneously predict various percentiles (e.g., the 1-st, 50-th and 99-th) of the targets at each future time step of interest, from here the name of multi-horizon forecasting, which can be denoted by

$$\hat{y}_i(q, t, \tau) = f_q(\tau, y_{i,t-k:t}, \mathbf{z}_{t-k:t}, \mathbf{x}_{t-k:t+\tau}; \boldsymbol{\omega}), \tag{2}$$

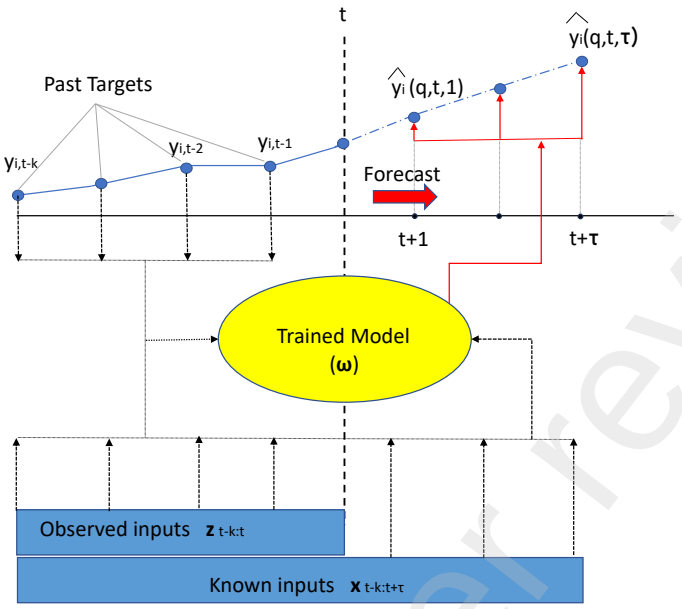

Figure 1: Illustrative diagram of multi-horizon forecasting using past observed and known inputs.

where $\hat{y}_i(q,t,\tau)$ is the predicted quantile $q \in \mathcal{Q} = \{0.01, 0.5, 0.99\}$ of the i-th target ($i \in I$) referring to the τ-step ahead the starting time t, for $\tau \in \{1, ..., \tau_{\max}\}$ (t, $t+\tau \in [0, T]$) and $f_q(\cdot; \boldsymbol{\omega})$ is the model induced by the TFT architecture, depending on the parameters $\boldsymbol{\omega}$. The model f_q takes as input the past information from the starting time t up to k time steps before, given by $y_{i,t-k:t} = \{y_{i,t-k}, y_{i,t-k+1}, ..., y_{i,t}\}$ (known past target values), $\mathbf{z}_{t-k:t} = \{\mathbf{z}_{t-k}, \mathbf{z}_{t-k+1}, ..., \mathbf{z}_t\}$ (observed inputs) and $\mathbf{x}_{t-k:t+\tau} = \{\mathbf{x}_{t-k}, \mathbf{x}_{t-k+1}, ..., \mathbf{x}_{t+\tau}\}$ (known inputs). A synthetic scheme of the multi-horizon forecasting is shown in Fig. 1, inspired by [56].

As one may easily notice, the evaluation of f_q on each set of the prescribed input variables $(\tau, y_{i,t-k:t}, \mathbf{z}_{t-k:t}, \mathbf{x}_{t-k:t+\tau})$ depends on the parameters vector $\boldsymbol{\omega}$, that can be obtained as the minimizer of the following Quantile Loss:

$$\mathcal{L}(\boldsymbol{\omega}) = \sum_{y_{i,t} \in \Omega} \sum_{q \in \mathcal{Q}} \sum_{\tau=1}^{\tau_{\max}} \frac{QL(y_{i,t}, \hat{y}_i(q, t-\tau, \tau), q)}{M \tau_{\max}}, \tag{3}$$

with

$$QL(y_{i,t}, \hat{y}_i(q, t-\tau, \tau), q) \\ = q \max\{0, y_{i,t} - \hat{y}_i(q, t-\tau, \tau)\} + (1-q)\max\{0, \hat{y}_i(q, t-\tau, \tau) - y_{i,t}\}, \tag{4}$$

where Ω is the domain of the training data containing M samples and the prediction $\hat{y}_i$ is defined by (2). As stated by the authors of [56], the benefits of a quantile regression are usually shown when targets are not well captured by Gaussian distributions.

We refer to the original work [56] for more details on TFTs and related issues.

2.2.1 Anomaly Detection

Once the TFT has been trained on a set of non-anomalous data, we indeed expect it to get the prediction $\hat{y}_i(0.5, t, \tau)$ of each of the desired targets $y_{i,t+\tau}$, $\tau \in \{1, ..., \tau_{\max}\}$, in the test set, together with its 1-st and 99-th percentiles, namely $\hat{y}_i(0.01, t, \tau)$ and $\hat{y}_i(0.99, t, \tau)$. An anomaly occurs at time $t+\tau$ if the observed value $y_{i,t+\tau}$ lies outside the confidence interval $[\hat{y}_i(0.01, t, \tau), \hat{y}_i(0.99, t, \tau)]$ predicted by the network.

For computing the anomaly scores $a_{i,t}$ for target i and timestep t, we forecast a single future sample ($\tau = 1$), and check how much this prediction is far from the observed value $y_{i,t+1}$. Let us define the following quantities:

$$o_{i,t} = \max(0, \max(y_{i,t+1} - \hat{y}_i(0.99, t, 1), \hat{y}_i(0.01, t, 1) - y_{i,t+1})), \tag{5}$$

$$w_{i,t} = \hat{y}_i(0.99, t, 1) - \hat{y}_i(0.01, t, 1), \tag{6}$$

where $o_{i,t}$ quantifies how much the observed value falls outside the confidence interval predicted by the network and $w_{i,t}$ is the width of the confidence interval at each timestep. Notice that $o_{i,t} = 0$, when $\hat{y}_i(0.01, t, 1) \leq y_{i,t+1} \leq \hat{y}_i(0.99, t, 1)$, and increases linearly outside that boundaries. At this point, the anomaly scores are scaled on $w_{i,t}$ and normalized as follows:

$$a_{i,t} = \min\left(1, \frac{n_{i,t} - \min_t n_{i,t}}{1 - \min_t n_{i,t}}\right) \qquad \text{where} \qquad n_{i,t} = \frac{o_{i,t}}{w_{i,t}}. \tag{7}$$

Through this formula we are setting the anomaly score to 1 when the observed value overshoots the estimated window size by 100%. If we overshoot by over 100%, we clamp the output to 1.

Finally, we compute the final anomaly score for each timestep t by taking the maximum over all the frequencies, $s_t = \max_i a_{i,t}$. This score can be used with an appropriate color map to visualize the magnitude of the anomaly lines, as in the anomaly plots shown in Section 4.

3 Case study: The Guinigi Tower

The Guinigi Tower, shown in Fig. 2, is a medieval tower sited in the historical center of Lucca, adjacent to the homonymous palace (Fig. 3), which incorporates the first 23 m out of the about 44 m of total height. The main feature of the tower, regularly open to the public, is the hanging garden on the top terrace, which makes the Guinigi Tower one of the most famous and visited monuments in Lucca. The rectangular underground basement, covering an area of 6.5×7.9 m^2, is not accessible to tourists, who start their visit from the ground level and follow the stairs crossing the Guinigi Palace to the level of 23 m. At this level, a metallic staircase (Fig. 2) allows one to enter the tower's structure and reach the roof terrace. The masonry walls have a constant thickness of about 1 m along the height.

In 2021, ISTI-CNR undertook an experimental campaign on the tower in collaboration with the Istituto Nazione di Geofisica e Vulcanologia (INGV). The structure has been continuously monitored by high-sensitivity seismic stations recording the structure's response to the environmental excitations of natural and anthropic origin, like earthquakes, wind, traffic, machinery, tourists, and moving crowds.

A dynamic monitoring system was installed on the tower, made up of three tri–axial SARA velocimeters with eigenfrequency 4.5 Hz and a triaxial velocimeter with eigenfrequency 2 Hz, each associated with a 24-bit digitizer (Fig. 2). In particular, two stations were placed right below the terrace at 40 m (Fig. 3), one at 18 m height and one at the underground level [70]. Two thermo-hygrometers with a sampling time of 60 s completed the monitoring system. The seismic stations have been acquiring data from August 2021 to July 2022, with a sampling frequency of 100 Hz.

3.1 Dynamic behavior of the tower

A detailed analysis of the results of the dynamic monitoring is presented in [70] and [71], where the dependence of frequencies on temperature and humidity, as well as the effect of the visitors and some seismic events on the tower's dynamic behavior, were investigated. In the following, we limit ourselves to recall the main results of such analysis.

Table 1 reports information about the first six natural frequencies of the Guinigi Tower calculated via the automated SSI-cov procedure described in [61], applied to the entire monitoring period after dividing the dataset of the velocities recorded by the two velocimeters placed at a height of 40 m (Fig. 3) into 1-hour long sequences. The time history of each frequency measured from August 2021 to July 2022 is analyzed to calculate the minimum value Min, the 1st percentile Min_1, average Avg,

Figure 2: External and internal views of the Guinigi Tower.

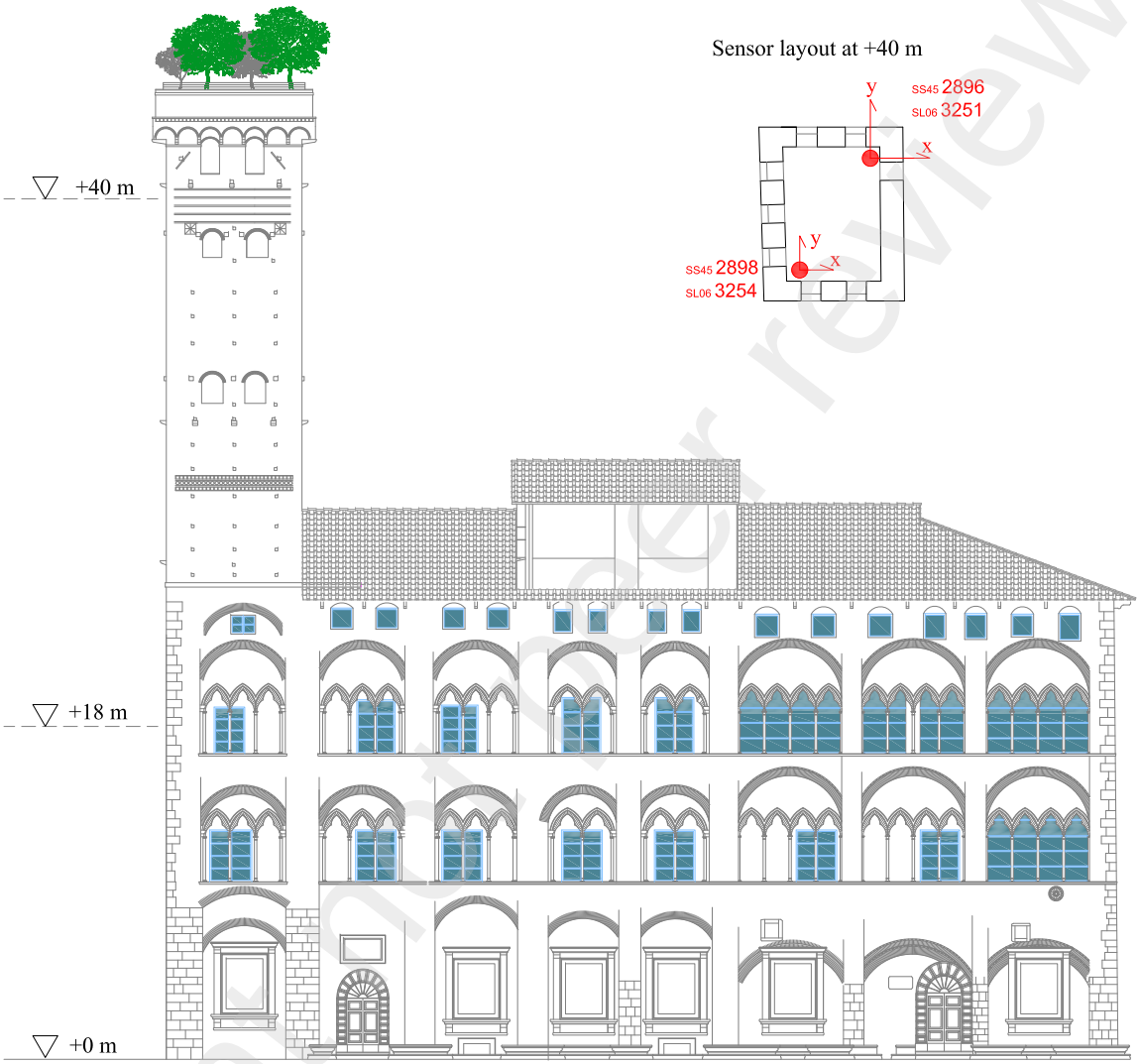

Figure 3: The Guinigi Palace and its tower: view from the S. Andrea street (courtesy of the Municipality of Lucca) and layout of the seismic stations installed at +40 m.

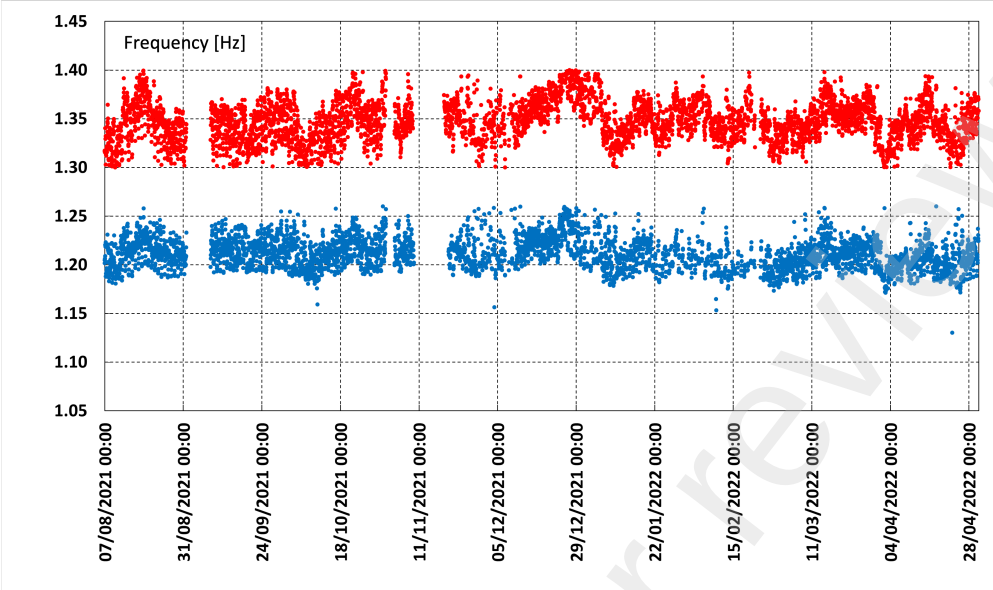

Figure 4: The tower's first (blue) and second (red) natural frequency measured from August 2021 to April 2022.

99th percentile Max_{99}, maximum value Max and relative difference $\Delta = (Max_{99} - Min_1)/Min_1$. The relative differences Δ of the first two frequencies and the sixth frequency are at most in the order of 7.72 % and these variations are in agreement with the results presented in [72] and [10]. For the remaining three frequencies, Δ reaches peaks up to 14.47 %. Regarding the frequencies' daily trend, the maximum daily variations are in the order of 3 - 4 %.

Several long–term vibration monitoring campaigns on ancient masonry towers ([4], [9], [5], [10]) have pointed out that the frequencies tend to increase with temperature. The Guinigi Tower does not exhibit such a trend. Fig. 4, reporting the plots of the tower's first two frequencies measured from August 2021 to April 2022, shows a very low correlation between temperature and frequencies, which, in turn, are relatively sparse. Without a specific explanation of this experimental result, it is worth noting that this scarce correlation also characterizes the Asinelli Tower in Bologna, which, like the Guinigi Tower, is open to the public and visited annually by many people [73].

The average velocities measured on the tower, located in a limited traffic area, are very low and are mainly influenced by the wind and visitors. Regarding the velocity peaks, Fig. 5 reports the maximum absolute values per hour of the velocities recorded

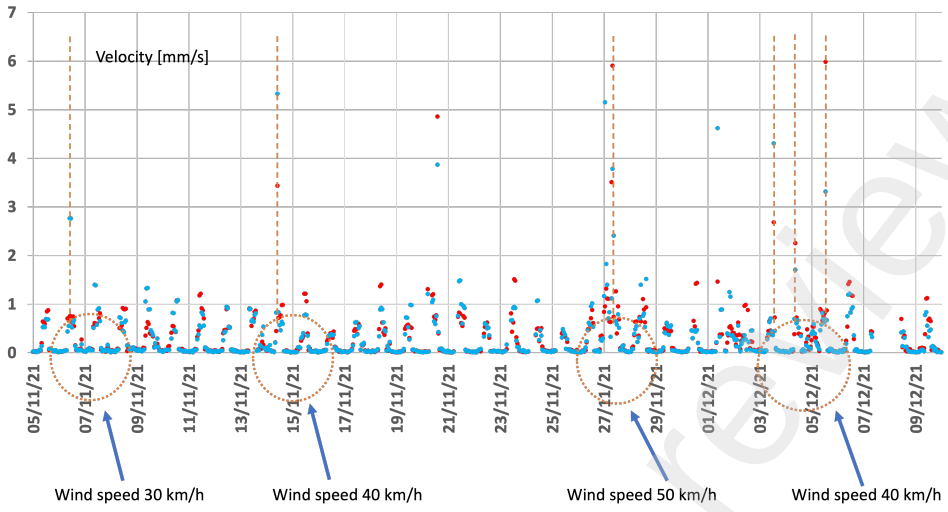

Figure 5: Maximum absolute values per hour of the velocities recorded by the seismic station set at +40 m in the x (red) and y (blue) directions from 5 November to 9 December 2021.

by a velocimeter set at 40 m in the x and y directions from 5 November to 9 December 2021. Velocity values greater than 2 mm/s occur in the windiest hours of the day – highlighted in the dashed circles – when the wind velocity exceeds 30 km/h. Data on the maximum wind speeds were recorded at Pieve di Compito, about 10 km from the Lucca historic center*.

Between July 2021 and May 2022, the seismic stations recorded 17 earthquakes of magnitude between 3.0 and 4.2 in a circular area centered on the Guinigi Tower with a radius of 108 km. Among them, the Viareggio earthquake of 6 February 2022 with magnitude 3.7, located 20 km from Lucca, induced the maximum acceleration on the top of the tower. Strong amplification of the signal along the tower's height was observed in the horizontal directions, along which the velocity at the top of the tower was more than 3.5 times that recorded at the base.

Fig. 6 shows the spectrograms of the signals recorded by the velocimeters from 1 to 7 February 2022 in the x (upper) and y (lower) directions. Every day there is an evident increase in the signal's power spectral energy due to the people visiting the tower (black rectangles). This phenomenon is connected with the relationship between the velocities and the presence of visitors in the tower. People moving in

*Data are available at `www.sir.toscana.it`

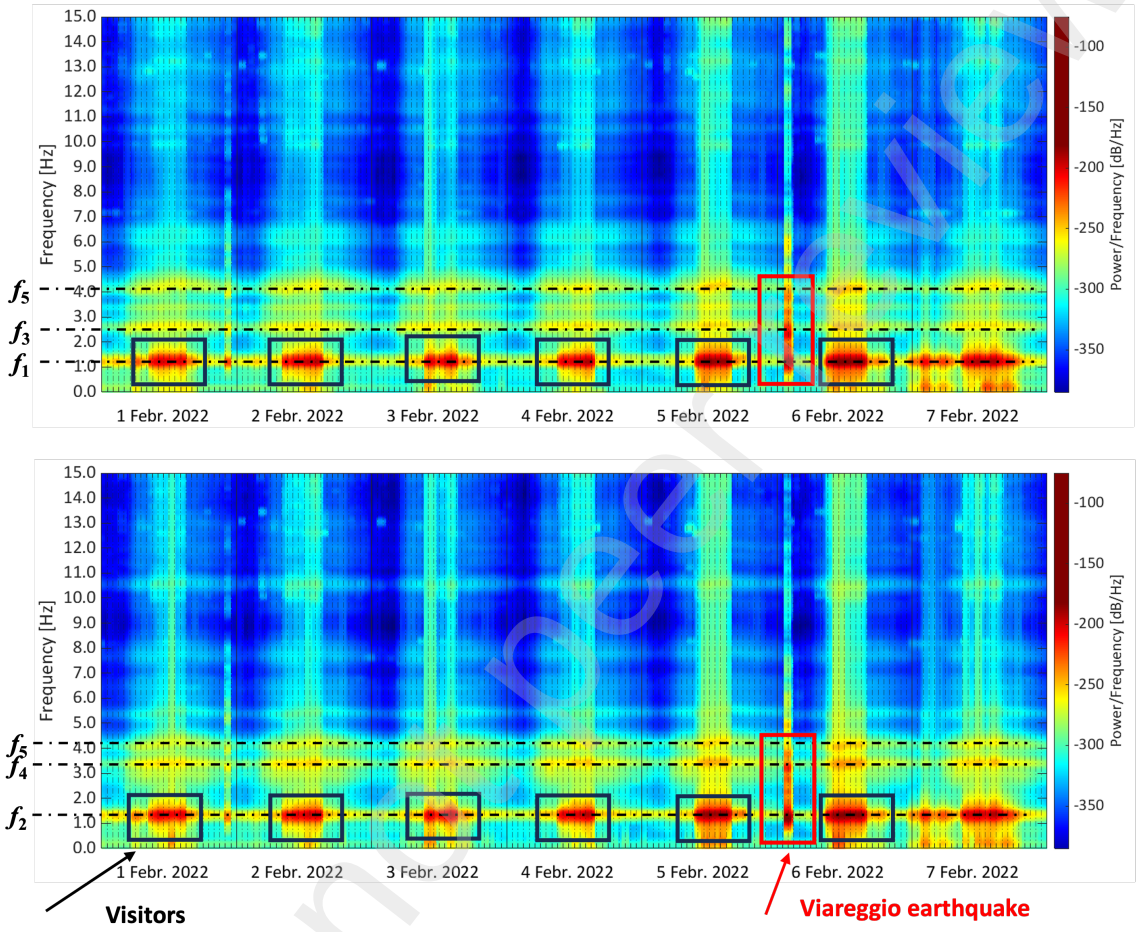

Figure 6: Spectrogram of the signal recorded at +40 m in the x (top) and y (bottom) directions from 1 to 7 February 2022. The Viareggio earthquake that occurred on 6 February at 1:36 UTC is highlighted in the red boxes. The effects of visitors on the tower's dynamics are highlighted in the black boxes.

Mode	Natural frequencies					
	Min [Hz]	Min_1 [Hz]	Avg [Hz]	Max_{99} [Hz]	Max [Hz]	Δ [%]
1 (bending x)	1.12	1.183	1.210	1.255	1.275	6.05
2 (bending y)	1.276	1.293	1.345	1.393	1.823	7.72
3 (bending x)	2.195	2.570	2.679	2.942	3.125	14.47
4 (bending y)	2.860	3.107	3.375	3.518	3.825	13.23
5 (torsional)	3.656	3.889	4.108	4.358	4.653	12.05
6 (axial)	5.906	6.232	6.346	6.498	6.711	4.27

Table 1: The tower's natural frequencies calculated via the SSI–cov algorithm during the one-year monitoring period: minimum Min, 1st percentile Min_1, average Avg , 99th percentile Max_{99}, maximum Max, relative differences $\Delta = (Max_{99} - Min_1)/Min_1$.

the tower (the tower's entrance is scheduled from 9:30 a.m. to 7:30 p.m., weekends included) produce an increase in the velocity, which rapidly vanishes at the end of the visiting hours [71].

In addition, the occurrence of the Viareggio earthquake is visible in the spectrogram, which exhibits an energy peak of about -100 dB/Hz on 6 February at 1:36 UTC (red rectangles). It is worth noting that the energy increase due to the visiting people is comparable to that induced by the Viareggio earthquake.

3.2 Description of the dataset for the TFT network

The reference dataset D of Subsection 2.1 is here given by a $m \times n$ matrix, with $m = 8384$ samples and $n = 19$ variables. The construction of the h-th row of D is described in Table 2. It is worth noticing that, at each row of the dataset, the date, the hour slot, and the weekend days can be regarded as known inputs ($\mathbf{x}$ in Subsection 2.2), the remaining entities are observed inputs $\mathbf{z}$. The target values $\mathbf{y}$ are the frequencies and/or the RMSs.

We recall that the six frequency values reported in each of the rows of dataset D (see Table 2) are obtained by means of the SSI-cov Python solver described in subsection 2.1. The values of the parameters used for the pre-processing operations (just based on detrending), the tolerances considered to get the stabilization diagram and the values of the parameters related to the clustering phase are summarized in Table 3. In particular, the table reports the minimum min_{ord} and maximum max_{ord} order method needed for getting the stabilization diagram; the relative errors on frequency (δ_f), damping (δ_d) and mode shape (δ_m); the upper bound κ_d on the

Column index/indices	h-th row of dataset D	Description
1	*date*	Reference date
2	*hour slot*	Reference slot of hour
3–8	$f_1,\ f_2,\ f_3,\ f_4,\ f_5,\ f_6$	First six natural frequencies [Hz] ($f_1 < f_2 < ... < f_6$) calculated via the automatic SSI-cov Python solver
9–11	*x-RMS* , *y-RMS* , *z-RMS*	x, y and z-*RMS* (Root Mean Squares) values of the time series of velocities registered by one of the two triaxial sensors of the terrace
12–13	$temp_1,\ hum$	Temperature [°C] and humidity (%) values registered by the thermo-hygrometers on the tower
14–15	$temp_2,\ rain$	Temperature [°C] and rain [mm] measurements from the Botanic Garden of Lucca
16–17	$wind_v$, $wind_d$	Wind speed [m/s] and direction (degrees from the North, clockwise positive direction) measured in Pieve di Compito
18	*tickets*	Number of sold tickets
19	*we*	Boolean variable to mark weekend days

Table 2: Entries of the h-th row of dataset D ($h = 1, ..., 8384$) with related descriptions.

Description	Parameter	Value
Minimum order method for the stabilization diagram	min_{ord}	2
Maximum order method for the stabilization diagram	max_{ord}	80
Relative error on frequencies	δ_f	10^{-2}
Relative error on dampings	δ_d	$5 \cdot 10^{-2}$
Relative error on mode shapes	δ_m	$2 \cdot 10^{-2}$
Upper bound on the damping ratio for a stable pole to be accounted	κ_d	$5 \cdot 10^{-1}$
Cutoff distance value considered in the automatic estimation of the modal parameters	*cutoff*	10^{-1}
Minimum number of elements of a cluster	$size_c$	5

Table 3: Parameters adopted in the automatic SSI-cov Python solver.

damping ratio for a stable pole to be accounted; the cutoff distance value *cutoff* considered within the automatic estimation of the modal parameters in which clusters with less than $size_c$ elements are neglected.

For the numerical tests, the splitting of the available data into the training, validation, and test sets has been reported in Table 4. Data normalization was performed using the standard Z-score normalization on the training set, i.e., the mean and the variance were estimated over the entire training set and re-used at inference time. Despite the need of retraining the network in case of consistent changes in the tower dynamics, this normalization enables the network to overfit the current dynamics, thus allowing a stronger response in case of severe structural damage. In the following, we refer to this normalization as *training normalization*. For comparison, in Subsection 4.1, we also report another criterion for normalization, called *encoder normalization*, which instead computes mean and variance over the moving time window traversed by the encoder during the inference.

The value of $k = 96$ has been employed in equation (2), i.e. a time window of 96 hours has been considered to individuate the past steps to be used for predictions at time $t + \tau$, $\tau \in \{1, ..., \tau_{\max}\}$. In particular, the value of $\tau_{\max}$ has been set to 1 for performing anomaly detection based on the predicted frequencies and/or the predicted RMS values (Subsections 4.1 and 4.2), while $\tau_{\max} = 96$ has been considered in Subsection 4.3, i.e. for predicting the main frequencies of vibrations over long

Training set	Validation set	Test set
Data of all the hour slots in the following periods:	*Data of all the hour slots in the following periods:*	*Data of all the hour slots in the following periods:*
1 Aug. – 10 Sep. 2021 25 Nov. 2021 – 27 Jan. 2022 17 Feb. – 14 July 2022	3 Nov. – 24 Nov. 2021 15 July – 31 July 2022	10 Oct. – 31 Oct. 2021 23 Jan. – 18 Feb. 2022

Table 4: Splitting of the available data into the training, validation, and test sets.

time periods. Throughout the numerical experimentation, different TFT models have been defined, depending on which of the n variables have been considered in the training phase of the TFT model. The considered TFT models are listed below, along with the vectors of the known ($\mathbf{x}$), observed ($\mathbf{z}$), and target ($\mathbf{y}$) values.

- **TFT**: the TFT is trained on all the variables in Table 2 to predict frequency and RMS values:

  $\mathbf{x} = \{date, hour\ slot, we\}$

  $\mathbf{z} = \{temp_1, temp_2, rain, hum, wind_v, wind_d, tickets,$
  $f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$

  $\mathbf{y} = \{f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$

- $\mathbf{TFT_{freq}}$: the TFT is trained considering only the environmental parameters and the frequency variables to predict the frequency values:

  $\mathbf{x} = \{date, hour\ slot, we\}$

  $\mathbf{z} = \{temp_1, temp_2, rain, hum, wind_v, wind_d, tickets, f_1, f_2, f_3, f_4, f_5, f_6\}$

  $\mathbf{y} = \{f_1, f_2, f_3, f_4, f_5, f_6\}$

- $\mathbf{TFT_{rms}}$: the TFT is trained considering only the environmental parameters and the RMS variables to predict the RMS values:

  $\mathbf{x} = \{date, hour\ slot, we\}$

  $\mathbf{z} = \{temp_1, temp_2, rain, hum, wind_v, wind_d, tickets, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$

  $\mathbf{y} = \{x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$

- $\mathbf{TFT_{no\text{-}env}}$: the TFT is trained considering only the frequency and the RMS values to predict frequency and RMS values:

$$\mathbf{x} = \{date, hour\ slot, we\}$$
$$\mathbf{z} = \{f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$$
$$\mathbf{y} = \{f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$$

- $\mathbf{TFT_{no\text{-}tick}}$: the TFT is trained neglecting the tickets information, to predict frequencies and RMS values:

$$\mathbf{x} = \{date, hour\ slot, we\}$$
$$\mathbf{z} = \{temp_1, temp_2, rain, hum, wind_v, wind_d, f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$$
$$\mathbf{y} = \{f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$$

- $\mathbf{TFT_{no\text{-}wind}}$: the TFT is trained neglecting the wind information, to predict frequencies and RMS values:

$$\mathbf{x} = \{date, hour\ slot, we\}$$
$$\mathbf{z} = \{temp_1, temp_2, rain, hum, tickets, f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$$
$$\mathbf{y} = \{f_1, f_2, f_3, f_4, f_5, f_6, x\text{-}RMS, y\text{-}RMS, z\text{-}RMS\}$$

4 Anomaly detection: the Viareggio earthquake and simulated damage

This section investigates the potentiality of the TFT algorithm in predicting the experimental values and revealing the anomalies of the tower's dynamic behavior. The analysis focuses on the testing set ranging from 23 January to 18 February 2022 (Table 4), during which the Viareggio earthquake occurred. Two damage indicators are considered in the analysis: the natural frequencies of the tower (in subsection 4.1) and the RMS of the signals (in Subsection 4.2). When not explicitly declared in the text, the results are reported for the *training* normalization rule. In Subsection 4.3 the predictive performance of the TFT technique is compared with the well established Auto-Regression with eXogenous inputs (ARX) model.

All the TFT runs described in the following sections were obtained on an AMD Ryzen 7 1700 Eight-Core Processor (16 Cores), 32 Gb RAM, and an RTX 2080Ti for CUDA acceleration. We used the Pytorch Forecasting framework [†], which provides

[†] https://pytorch-forecasting.readthedocs.io/en/stable/

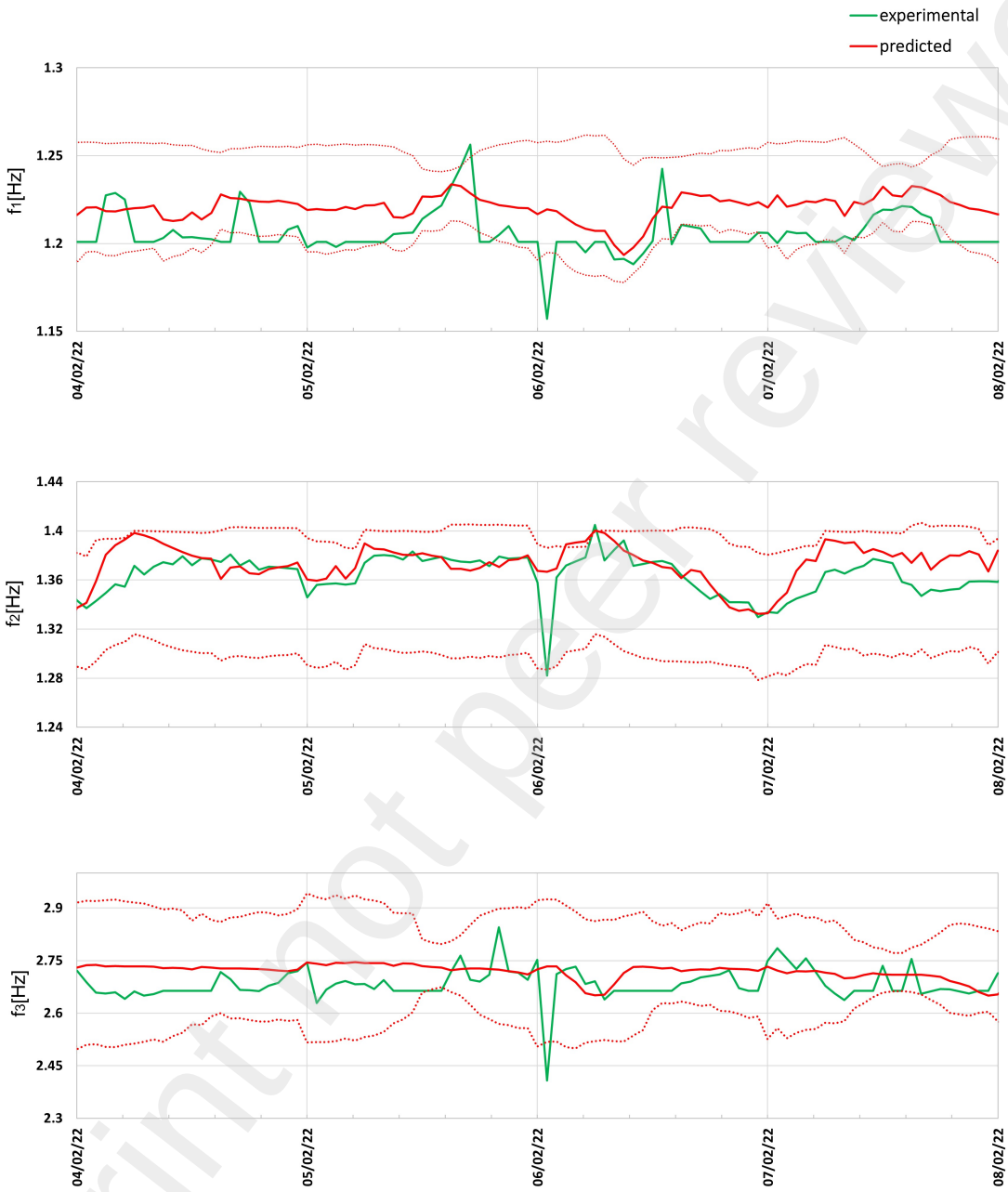

Figure 7: Period 4 - 8 February 2022, TFT model. First three predicted (red line) and experimental frequencies (green line). The red dashed lines represent the 1st and 99th percentiles predicted by the model.

an implementation of the TFT network [56]. Given the relatively small training set size, we found that instantiating the network with a few parameters helped achieve better generalization. Specifically, we used a general hidden size of 32, with a hidden size for continuous variables of 16 and only one attention head. We trained the network until the validation loss did not decrease for three consecutive epochs. On the hardware mentioned above, training took around 40 seconds per epoch. Considering that, on average, the best validation metrics are obtained after the first eight epochs, the total training phase took about 5 minutes. On the other hand, on our test set, inference required only about around 1.3 seconds, which is more than acceptable for practical applications of the system.

4.1 Frequency-based predictions and anomaly detection

This section reports the results of the numerical investigations for the models described in the previous section, when the frequencies are considered as damage indicators. Fig. 7 plots the observed (green line) and predicted (50th percentile, red line) values of the first three frequencies from 4 to 8 February 2022, along with the 1st and 99th percentiles (dashed line) evaluated by the TFT model. The anomaly caused by the Viareggio earthquake (6 February at 01:36 UTC) is highlighted in the figure; the event affects all three frequencies of the tower, the first and third, corresponding to bending mode shapes in the x direction, and the second corresponding to a bending mode shape in the y direction (Table 1).

Fig. 8–9 summarise the results for the period 23 January - 12 February 2022 and show the observed frequencies (orange line) and the ones predicted by the algorithm (corresponding to the 50th percentile, blue line), together with the estimate of uncertainty, which is represented by the grey area between the 1st and 99th percentiles at every prediction time step, for each predicted frequency (confidence interval). The vertical bars show the time locations where at least one predicted frequency does not fall within the estimated confidence interval. The magnitude scale introduced in the figures follows the rules defined in (5)–(7), in which $n_{i,t}$ is replaced by

$$n_{i,t} = \rho_i \, \frac{o_{i,t}}{w_{i,t}}, \qquad \rho_i = \frac{\bar{o}_1}{\bar{o}_i} \tag{8}$$

and $\bar{o}_i$ is the mean value of the i-th frequency over all the time axis. Coefficients $\rho_i \in [0, 1]$ are weights that reduce the contribution to the anomaly score of the higher frequencies, whose evaluation is usually uncertain.

As shown in the figures, the algorithm captures the anomaly behavior induced by the Viareggio earthquake on 6 February at 01.36 UTC, for the five models TFT,

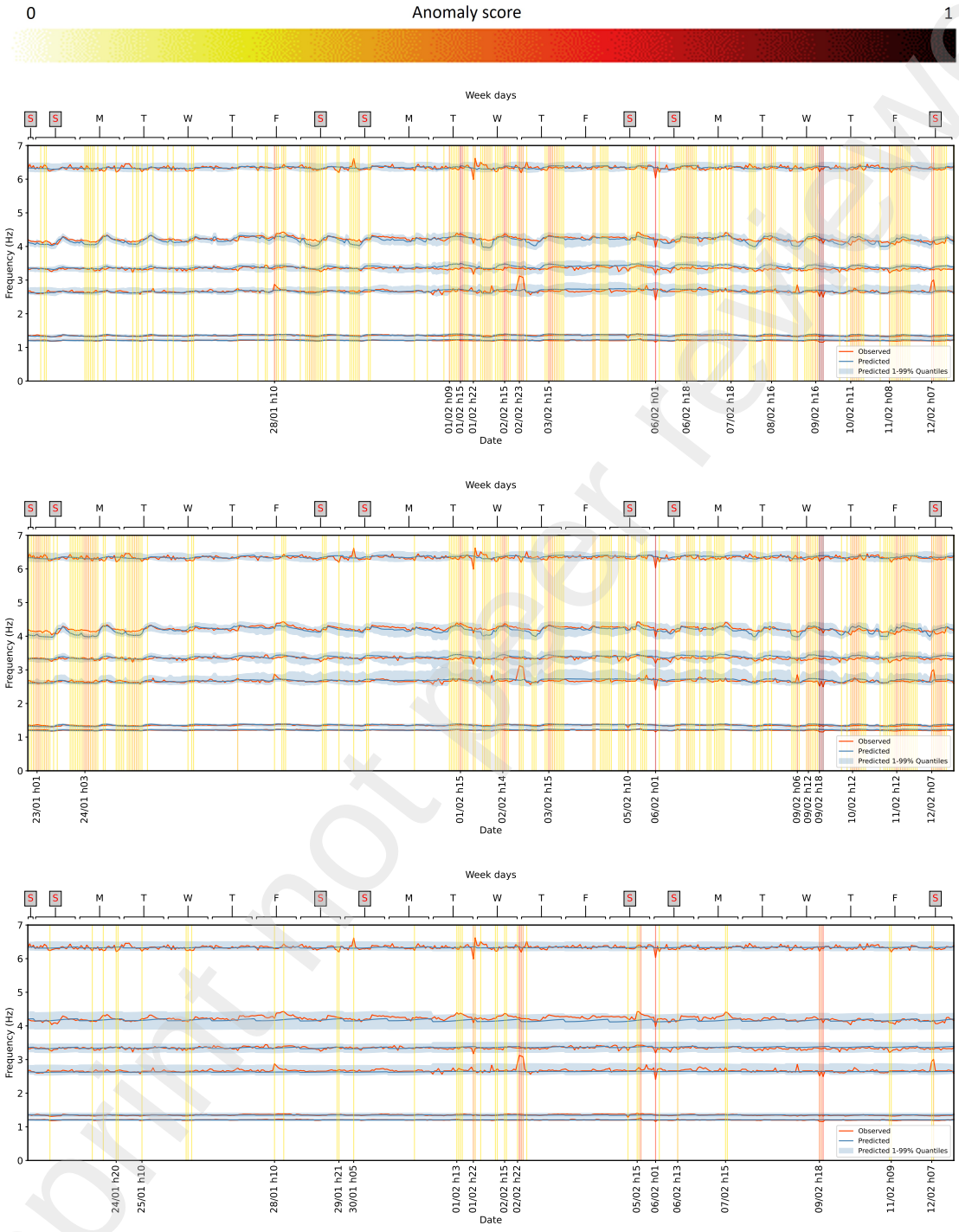

Figure 8: Period 23 January - 12 February 2022, Viareggio earthquake. Anomaly plots for models TFT (top), TFT_{freq} (middle) and $\text{TFT}_{\text{no-env}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

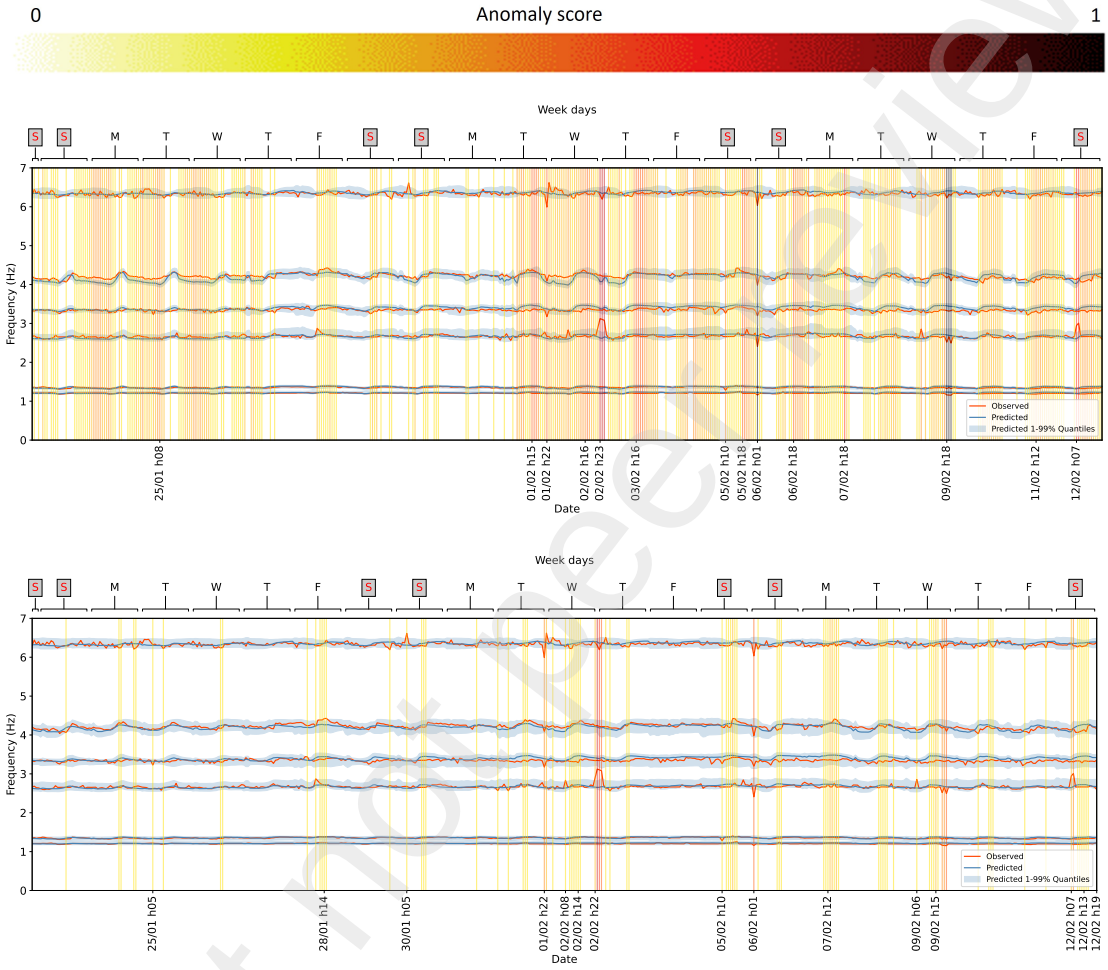

Figure 9: Period 23 January - 12 February 2022, Viareggio earthquake. Anomaly plots for models $\text{TFT}_{\text{no-tick}}$ (top) and $\text{TFT}_{\text{no-wind}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

$\mathrm{TFT_{freq}}$, $\mathrm{TFT_{no\text{-}env}}$, $\mathrm{TFT_{no\text{-}tick}}$ and $\mathrm{TFT_{no\text{-}wind}}$ introduced in Subsection 3.2 (model $\mathrm{TFT_{rms}}$ is used for the RMS prediction in Subsection 4.2). As expected, the seismic event does not represent the only anomaly detected in the period. In fact, several vertical lines are spotted in the figures and can be attributed mainly to the presence of the tourists on the tower; the magnitude of such events is mostly lower than that of the earthquake. The seismic event does not induce a permanent change in the dynamic behavior of the tower: in fact, the anomaly bar is focused on the hour of the earthquake, and in the next hours, the frequency predictions return to the canonical trend.

The procedure gives satisfactory results for every model shown in the Fig. 8–9. Nevertheless, the distribution of the anomaly bars changes depending on the model. In particular, models $\mathrm{TFT_{no\text{-}env}}$ and $\mathrm{TFT_{no\text{-}wind}}$, although still able to catch the Viareggio earthquake, seem less sensitive to abnormal values. Such models forecast large confidence intervals and the 50th percentile frequency values do not exhibit significant variations during time. This behavior suggests that the information on the environmental variables, which is missing in these models, increases the capability of the algorithm to understand the system's behavior in time. Moreover, model $\mathrm{TFT_{no\text{-}tick}}$ shows the narrower amplitudes of the confidence interval, thus suggesting that the information on the number of people in the tower has a low influence on the frequency prediction.

In order to test the algorithm's capability to detect possible damage, we have considered simulated damage scenarios in which the tower's six observed frequencies are reduced by a given percentage starting from 6 February at 1:00 UTC, the hour slot during which the Viareggio earthquake occurred.

Three damage scenarios are considered, in which the values of the six experimental frequencies $f_1 < f_2 < ... < f_6$ are lowered according to the following rules:

Scenario 1: by $2\%, 1\%, 0.5\%, 0.5\%, 0.5\%$ and 0.5%, respectively. This damage scenario is inspired by that observed on the Gabbia tower in Mantua after the Emilia earthquake of May 2012 [9].

Scenario 2: by $4\%, 2\%, 1\%, 1\%, 1\%$ and 1%, respectively.

Scenario 3: by 4% (uniform reduction of all frequencies).

Fig. 10 to 15 show the anomalies detected by the algorithm for the three damage scenarios and the five tested models. The algorithm highlights the change in the tower's dynamic properties after the earthquake for every model and every damage

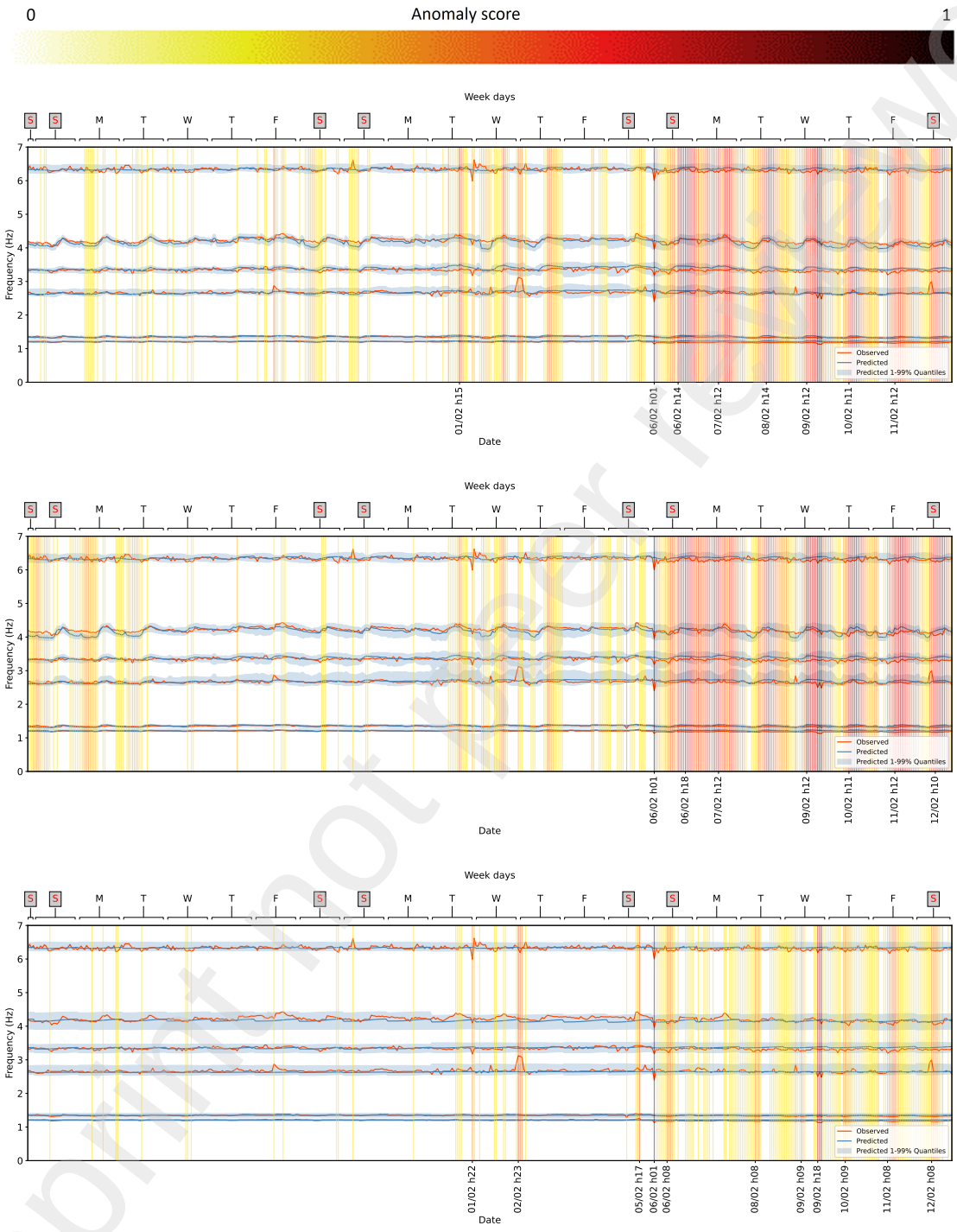

Figure 10: Simulated damage, **Scenario 1**. Anomaly plots for models TFT (top), TFT_{freq} (middle) and $\text{TFT}_{\text{no-env}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

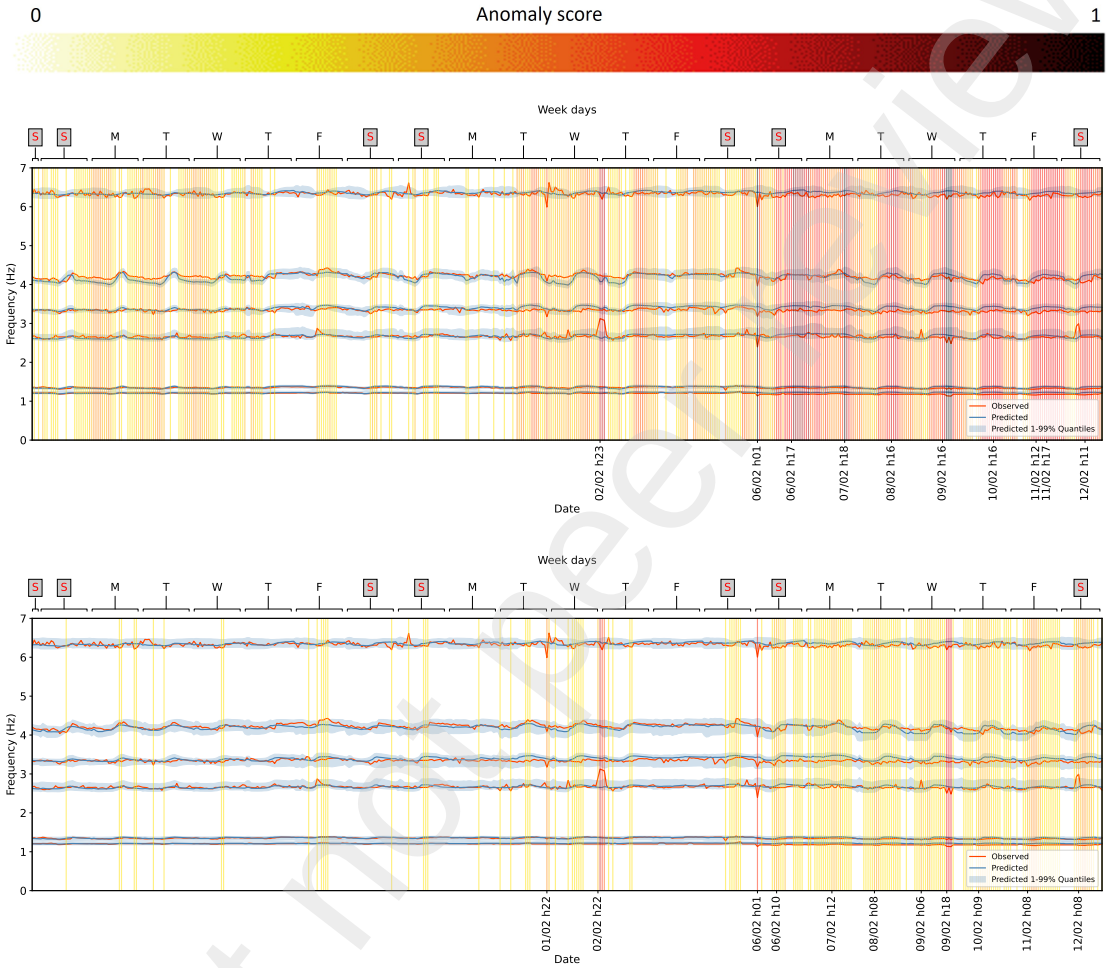

Figure 11: Simulated damage, **Scenario 1**. Anomaly plots for models $\text{TFT}_{\text{no-tick}}$ (top) and $\text{TFT}_{\text{no-wind}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

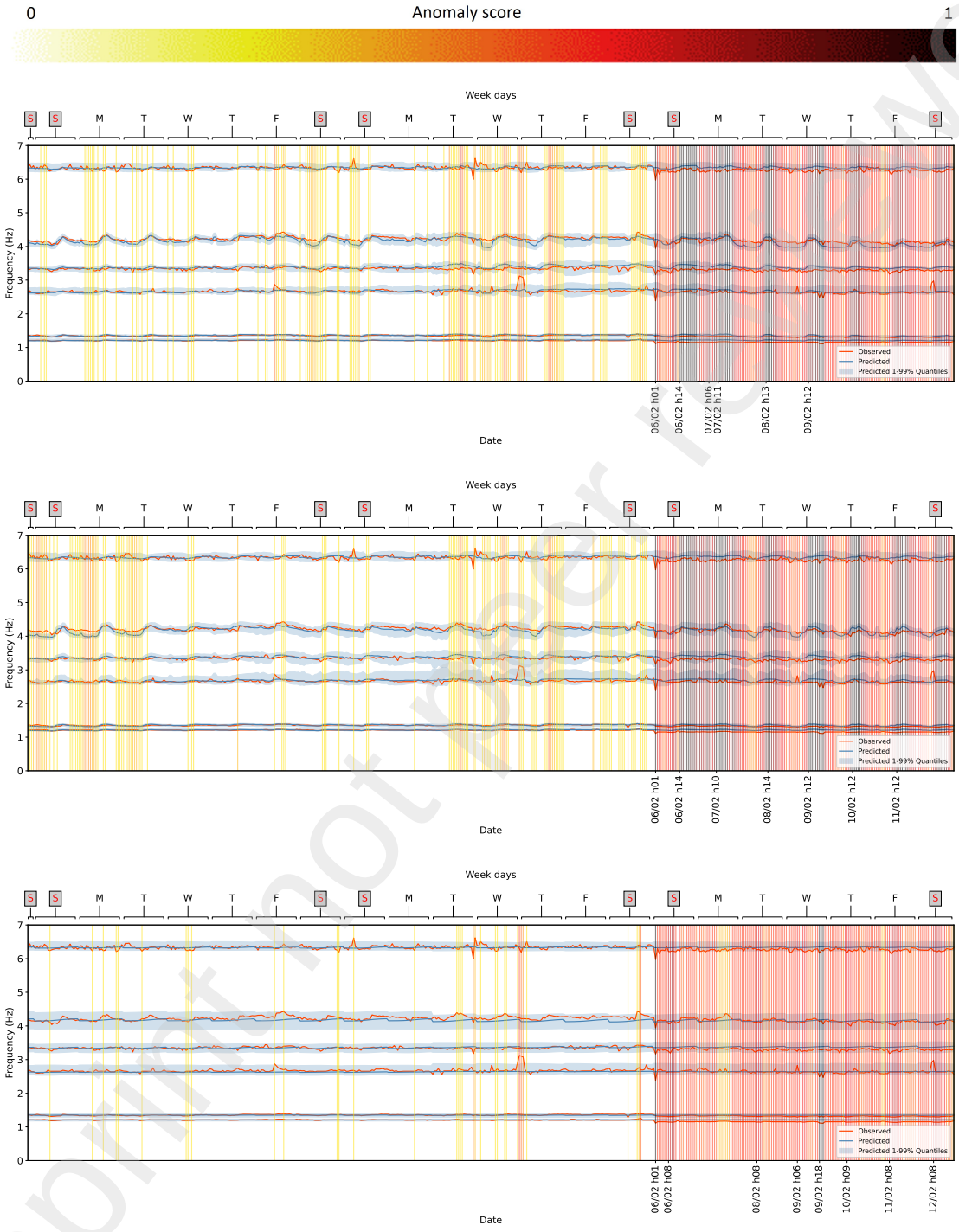

Figure 12: Simulated damage, **Scenario 2**. Anomaly plots for models TFT (top), TFT_{freq} (middle) and $\text{TFT}_{\text{no-env}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

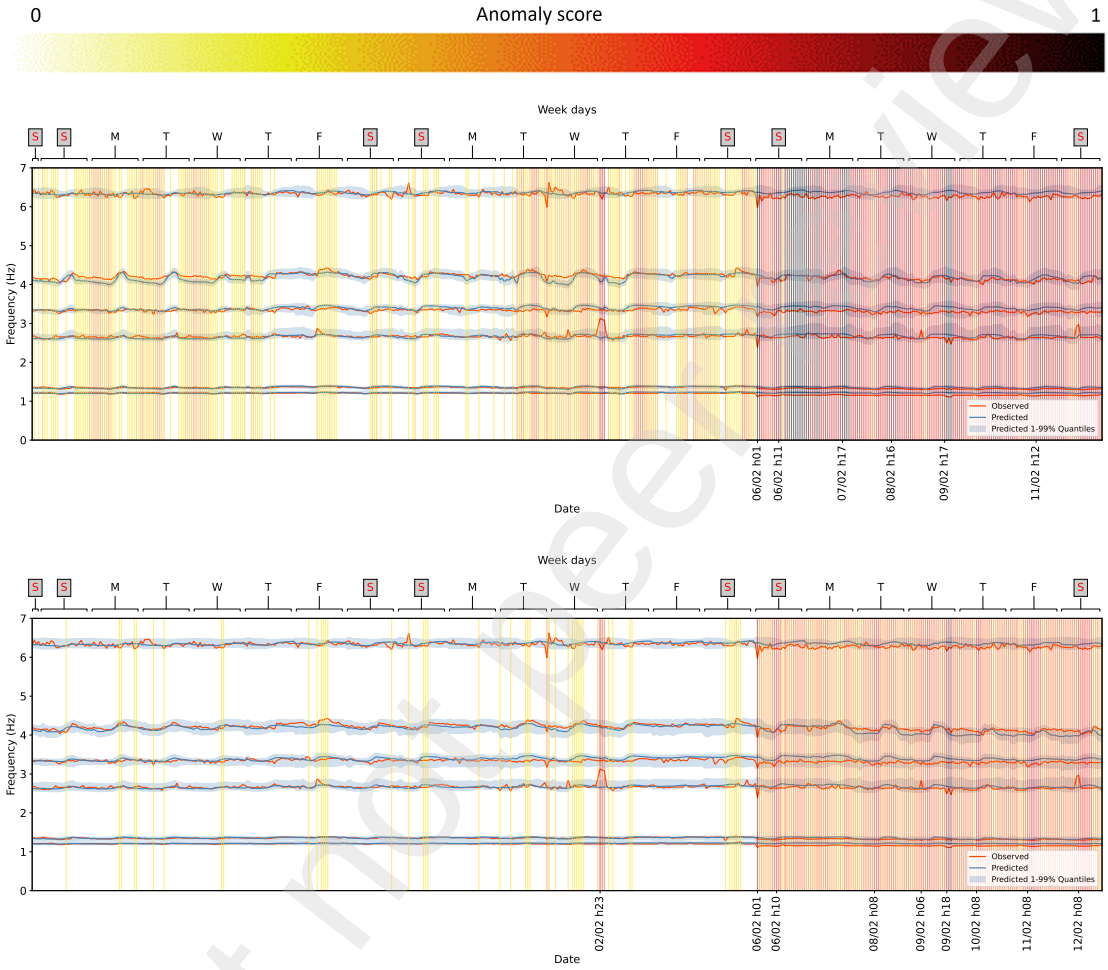

Figure 13: Simulated damage, **Scenario 2**. Anomaly plots for models $\text{TFT}_{\text{no-tick}}$ (top) and $\text{TFT}_{\text{no-wind}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

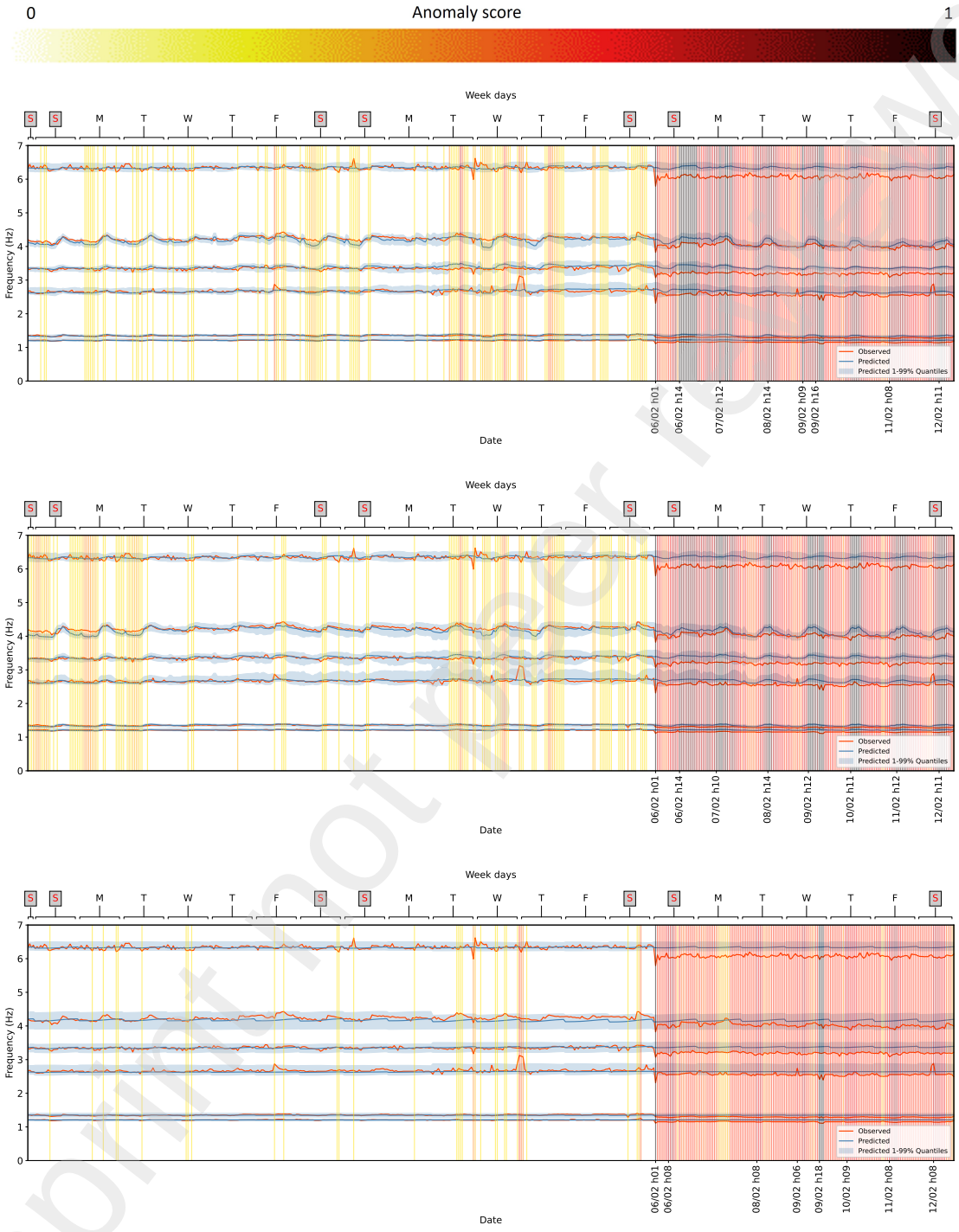

Figure 14: Simulated damage, **Scenario 3**. Anomaly plots for models TFT (top), TFT_{freq} (middle) and $\text{TFT}_{\text{no-env}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

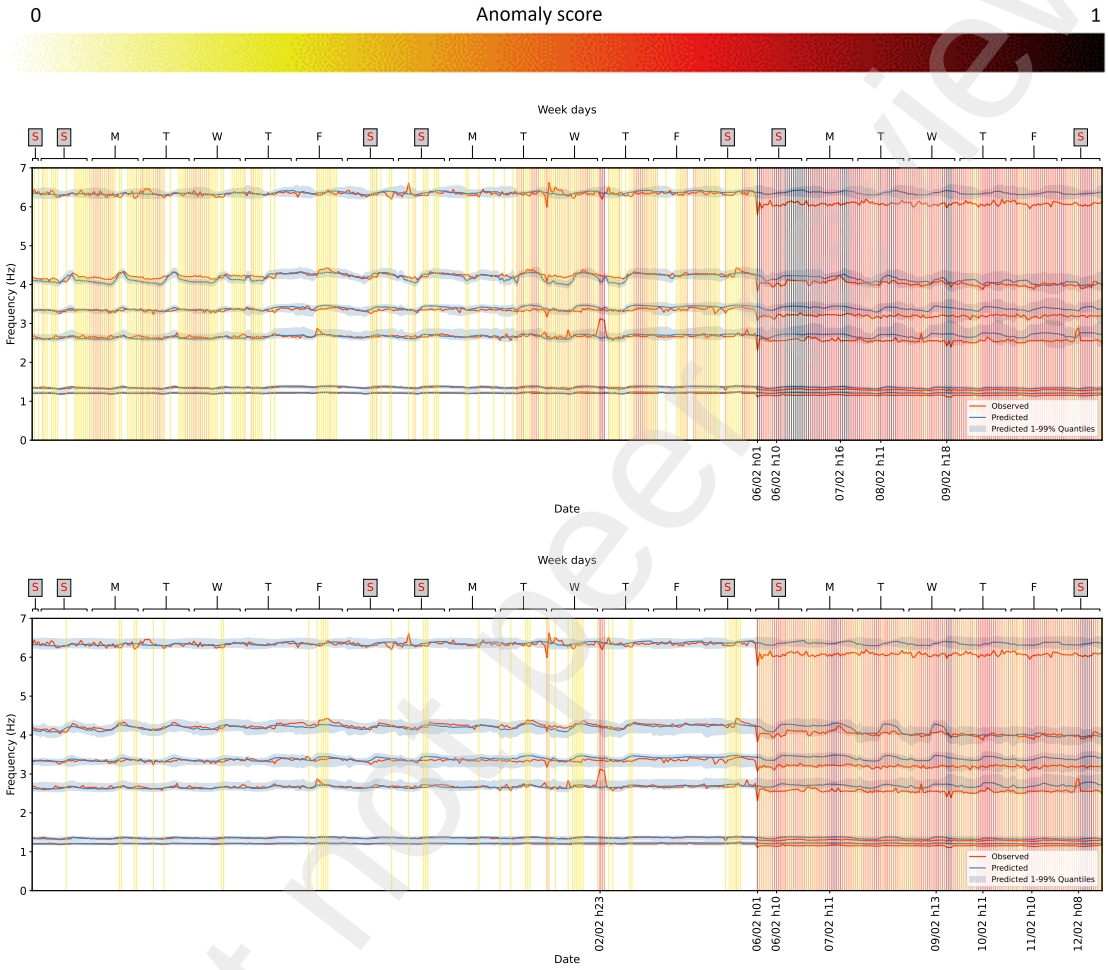

Figure 15: Simulated damage, **Scenario 3**. Anomaly plots for models $\text{TFT}_{\text{no-tick}}$ (top) and $\text{TFT}_{\text{no-wind}}$ (bottom). Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

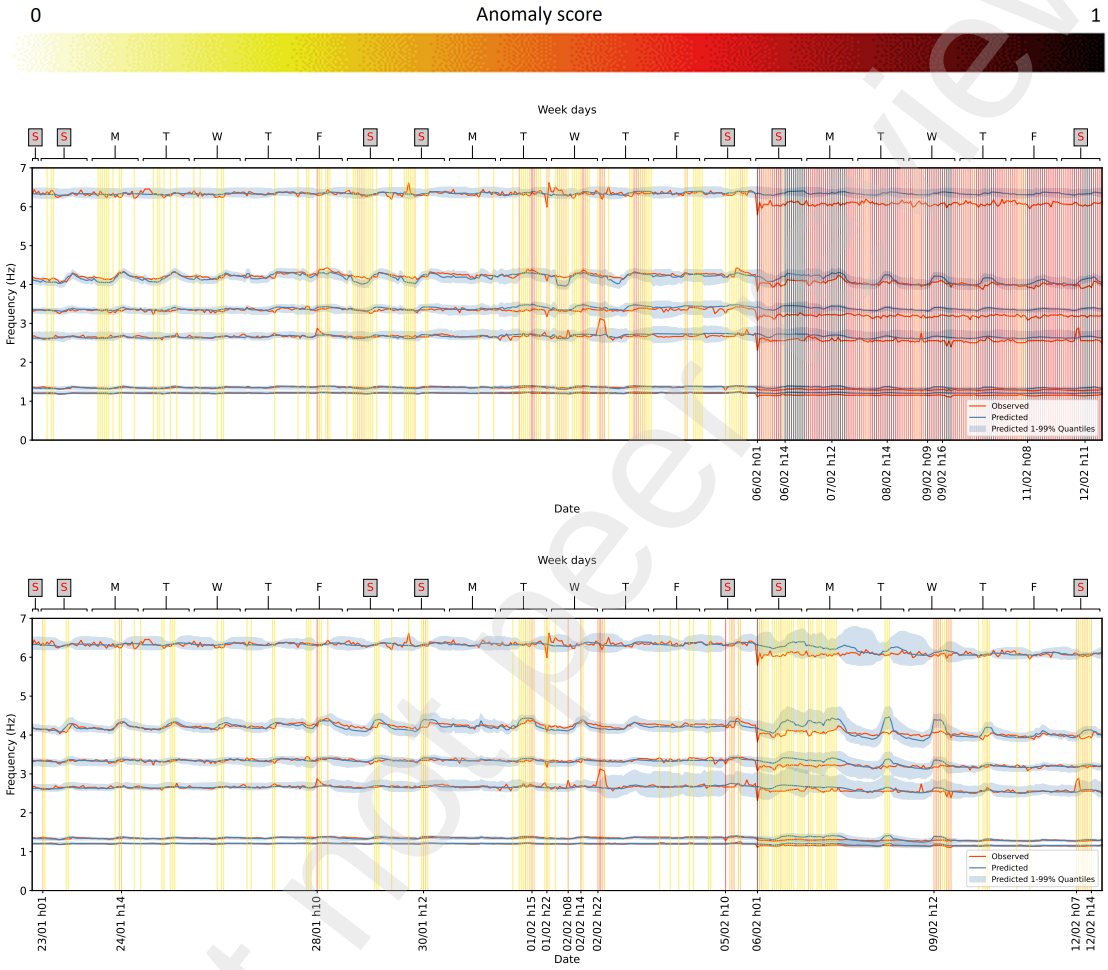

Figure 16: Simulated damage, **Scenario 3**. Anomaly plots for the TFT model using the *training* normalization (top) and *encoder* normalization. Observed (orange line) and predicted (blue line) frequencies (Hz) and confidence interval (grey shadow).

scenario. The magnitude of the detected anomalies increases as the damage intensity increases from Scenario 1 to Scenario 3. The anomaly is permanently highlighted after the earthquake in all the cases shown in the figures.

Nevertheless, in some applications, the algorithm is expected to learn from the data characterizing the damage scenarios and adapt the procedure to the new situation without further training. This result can be achieved using different normalization rules, as shown in Fig. 16. Here, the results obtained via *training* normalization are compared with those achieved by normalizing data on the time window k (equation (2)), thus updating the normalization during the test (*encoder* normalization). This kind of normalization allows the algorithm to estimate the mean and variance of input data from the moving time window having width k; as a result, in the case of Fig. 16, the procedure highlights the anomaly induced by Scenario 3 in the first hours after the earthquake (for about 36 hours) and, unlike the *train* normalization, progressively adapt the prediction to the new situation. After a few days, the algorithm has learned that the conditions of permanent damage have become the normality.

4.1.1 Analysis of the TFT internal state

Finally, we analyzed the TFT internal state change after the simulated damage. Specifically, we extracted 64-dimensional features from the internal self-attention mechanism of the TFT [56] at time $\tau_{\max} = 1$ for consecutive temporal windows in the period January–February 2022. We call this set of vectors *feature trajectories*, as it depicts the evolving internal state of the network. We then performed reduction to the main three components using the Principal Component Analysis (PCA) algorithm [74]. Specifically, we took the three main eigenvalues obtained from PCA, for each of the 64-dimensional vectors in the trajectory. Fig. 17 shows the obtained feature trajectories in this reduced 3-dimensional space. We can notice how the trajectories deviate after the damage (orange traces), with respect to the base case with no damage (blue traces). This suggests that the model's internal state is aware of the anomaly, and further studies may be conducted on the latent features of the TFT network for performing a post hoc classification of the anomalies (e.g., to distinguish structural damage from a temporarily overcrowded environment).

4.2 RMS-based anomaly detection

Fig. 18-19 report the results of a TFT anomaly detection procedure based on RMS predictions.

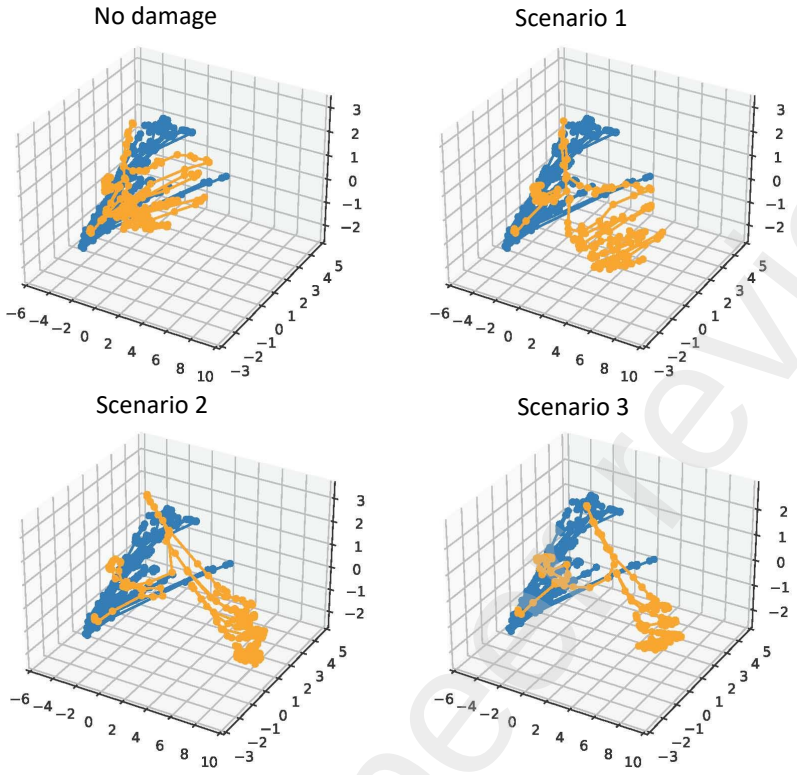

Figure 17: Feature trajectories in the space of the first three principal components for different damage scenarios. Orange lines represent the trajectories after the damage occurred.

The confidence intervals for the predicted values are larger than those predicted for the frequencies, probably due to the RMS's lower correlation with the environmental data and the low energy content of the signal.

In addition, the prediction of the $\mathrm{TFT}_{\mathrm{rms}}$ model (trained neglecting the frequencies) is much more accurate than that of the TFT model, in which the entire dataset is used to train the algorithm. This result suggests a weak correlation between RMSs and frequencies, acting as a noise which increases the model's uncertainty. Nevertheless, both models can detect the anomaly induced in the signals by the Viareggio earthquake. The models can also predict the trend of the RMSs during the central hours of the day when the tower is open to tourists and during the weekends when the number of visitors tends to increase. Finally, Fig. 20 shows the x- and y- RMS predictions of the $\mathrm{TFT}_{\mathrm{rms}}$ model and the corresponding observed values for the test-

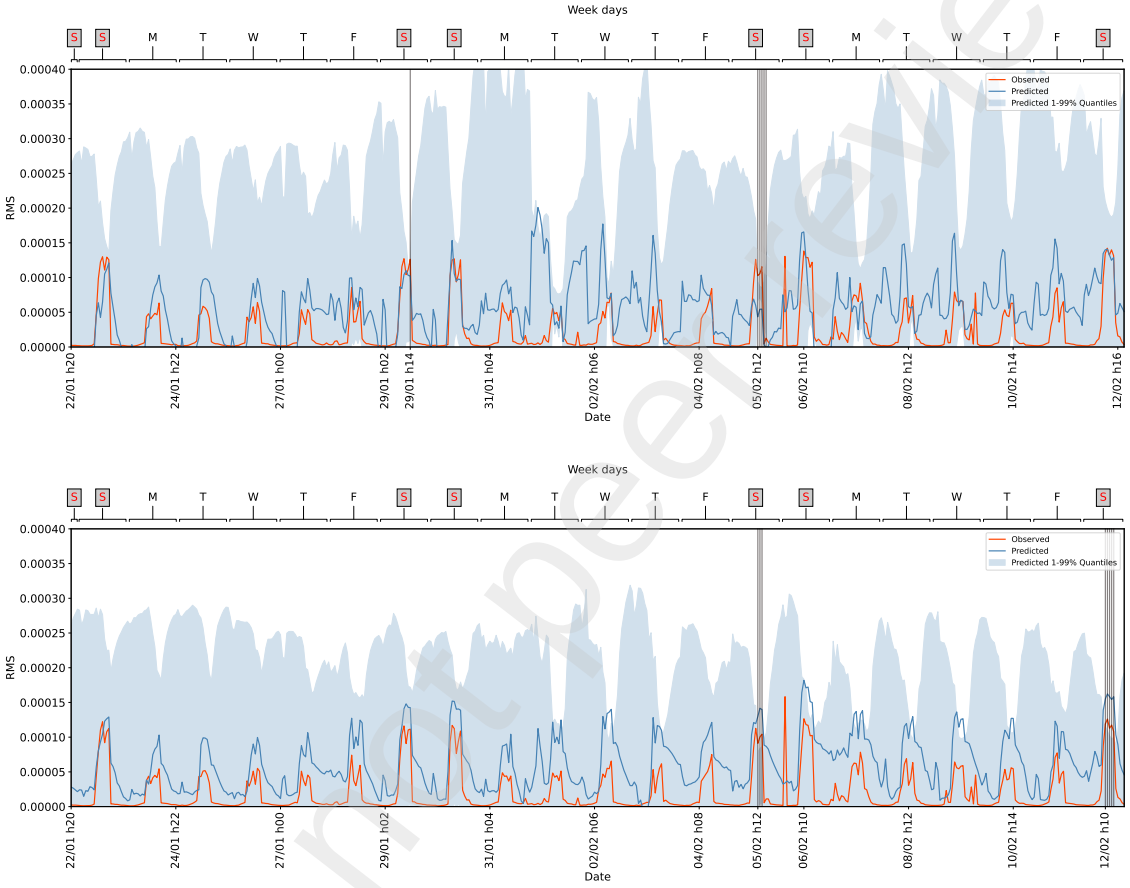

Figure 18: February 2022, Viareggio earthquake. TFT model, RMS-based anomaly detection for the signals in x (top) and y (bottom) direction. Observed (orange line) and predicted (blue line) RMS values (m/s) and confidence interval (grey shadow).

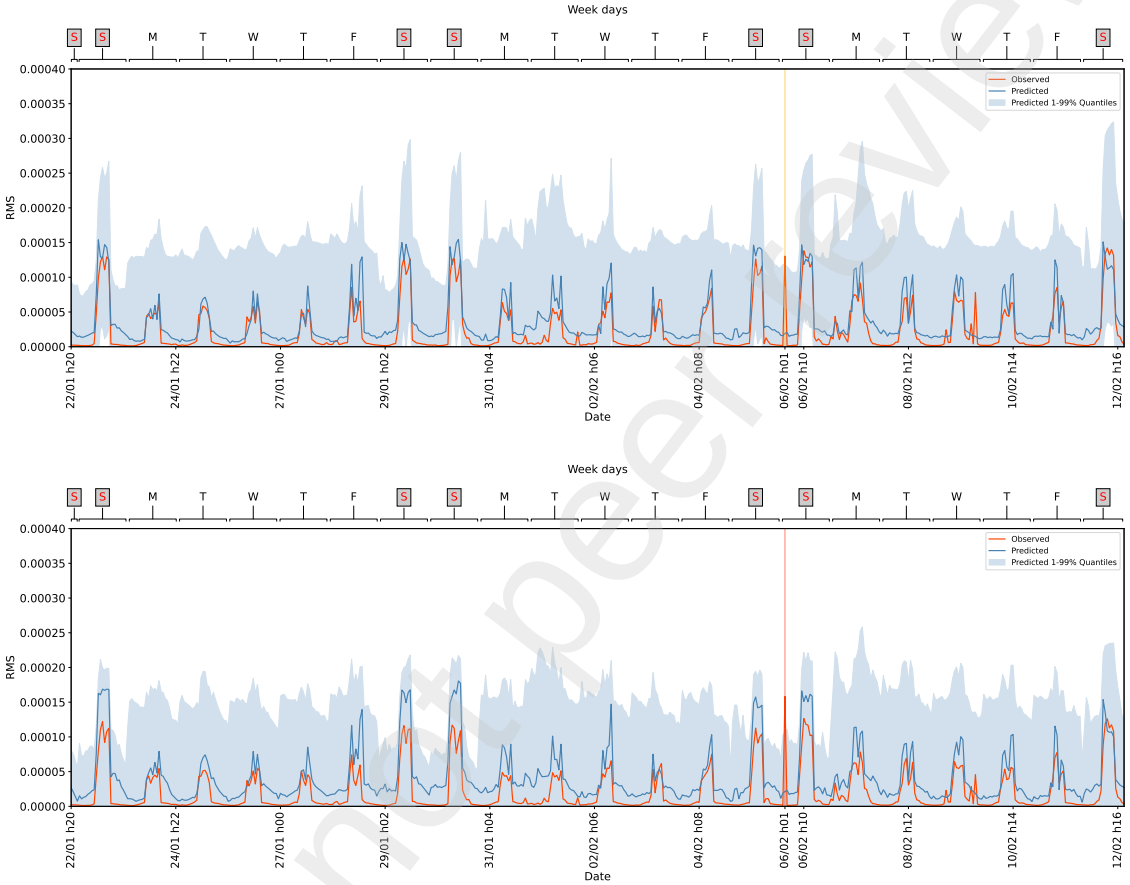

Figure 19: February 2022, Viareggio earthquake. $\mathrm{TFT}_{\mathrm{rms}}$ model, RMS-based anomaly detection for the signals in x (top) and y (bottom) direction. Observed (orange line) and predicted (blue line) RMS values (m/s) and confidence interval (grey shadow).

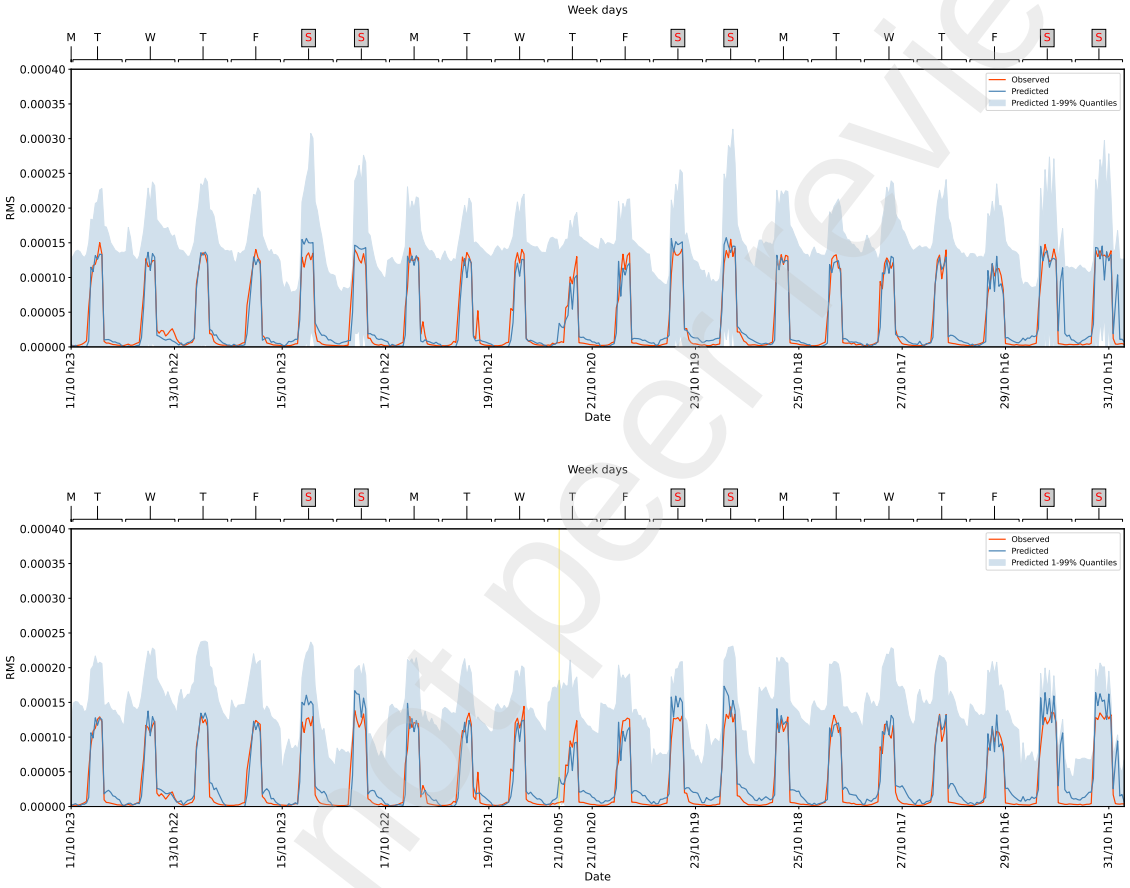

Figure 20: October 2021. TFT_{rms} model, RMS-based anomaly detection for the signals in x (top) and y (bottom) direction. Observed (orange line) and predicted (blue line) RMS values (m/s) and confidence interval (grey shadow).

ing period 10-31 October 2021: good accuracy is achieved by the model for these data in the absence of earthquakes or other relevant events.

4.3 Investigating the predictive model

In this section, we use the multi-horizon forecasting capabilities of the TFT network (Fig. 1) to predict the dynamic behavior of the Guinigi Tower. For such a purpose, we choose a temporal horizon of 96 hours ($k = \tau_{\max} = 96$ in equation (2)), on which the tower's frequencies are predicted by the TFT model and compared with the experimental values. In order to assess the performances of the network, we introduce a comparison with the well-established Auto-Regression with eXogenous inputs (ARX) model [75], whose description is briefly recalled in Subsection 4.3.1. The two models are compared in Subsection 4.3.2 in terms of prediction capabilities with respect to the experimental values measured on the tower.

4.3.1 The ARX baseline

In order to assess the predictive performance of TFT, numerical tests employing the ARX model implemented in Matlab have been performed. ARX models, devoted to modeling and analyzing the behavior of dynamic systems, are widely employed in SHM to simulate the dependence of experimental frequencies on environmental parameters and detect abnormal structural changes [9], [10]. They are a combination of an Auto-Regression (AR) and an eXogenous input (X) model. A time series is represented by a linear combination of its past values, a set of exogenous input variables and a stochastic process:

$$y_p^{(\ell)} = \sum_{i=1}^{n_a^{(\ell)}} a_i^{(\ell)} y_{p-i}^{(\ell)} + \sum_{j=1}^{n_b^{(1)}} b_j^{(1)} u^{(1)}_{p-n_p^{(1)}-j+1} + ... + \sum_{j=1}^{n_b^{(s)}} b_j^{(s)} u^{(s)}_{p-n_p^{(s)}-j+1} + e_p^{(\ell)}, \qquad (9)$$

where, with reference to the ℓ-th experimental frequency f_ℓ ($f_1 < f_2 < ... < f_6$), $\ell \in \{1, ..., 6\}$ in this case, we have

- $y_p^{(\ell)}$ ($\ell \in \{1, ..., 6\}$) is the output frequency at the p-th time instant;
- $a_i^{(\ell)}$, for $i \in \{1, ..., n_a^{(\ell)}\}$, are the $n_a^{(\ell)}$ autoregressive coefficients of the ℓ-th frequency;
- $y_{p-i}^{(\ell)}$, for $i \in \{1, ..., n_a^{(\ell)}\}$, are the $n_a^{(\ell)}$ past values of the ℓ-th frequency;

- $u^{(\nu)}_{p-n_p^{(\nu)}-j+1}$, for $j \in \{1, ..., n_b^{(\nu)}\}$, are $n_b^{(\nu)}$ past values of each of the exogenous input variable $u^{(\nu)}$, starting from the instant of time $p - n_p^{(\nu)}$, for $\nu \in \{1, ..., s\}$, with $s = 11$ the number of variables related to column indices 9–19 in Table 2.
- $b_j^{(\nu)}$, for $j \in \{1, ..., n_b^{(\nu)}\}$, are the $n_b^{(\nu)}$ coefficients that capture the relationship between each of the input variable $u^{(\nu)}$ and the output y_p^{ℓ}, for $\nu \in \{1, ..., s\}$;
- $n_a^{(\ell)}$ is the so-called auto-regressive order; for $\nu \in \{1, ..., s\}$, $n_b^{(\nu)}$ is the exogenous order of the input variable $u^{(\nu)}$, while $n_p^{(\nu)}$ is the time delay between the input variable $u^{(\nu)}$ and the output;
- $e_p^{(\ell)}$ is a white Gaussian disturbance noise.

With reference to equation (9), we underline that when $n_a^{(\ell)} = 0$ for $\ell \in \{1, ..., 6\}$, $n_b^{(\nu)} = 1$ and $n_p^{(\nu)} = 0$ for $\nu \in \{1, ..., s\}$ are considered, then the ARX model (9) reduces to a multidimensional (static) linear regression.
For each output $y_p^{(\ell)}$, $\ell \in \{1, ..., 6\}$, setting

$$\vartheta^{(\ell)} = \left(a_1^{(\ell)}, ..., a^{(\ell)}_{n_a^{(\ell)}}, b_1^{(1)}, ..., b^{(1)}_{n_b^{(1)}}, ..., b_1^{(s)}, ..., b^{(s)}_{n_b^{(s)}} \right)^{\top}, \tag{10}$$

$$\varphi_p^{(\ell)} = - \left(y^{(\ell)}_{p-1}, ..., y^{(\ell)}_{p-n_a^{(\ell)}}, u^{(1)}_{p-n_p^{(1)}}, ..., u^{(1)}_{p-n_p^{(1)}-n_b^{(1)}+1}, ..., u^{(s)}_{p-n_p^{(s)}}...., u^{(s)}_{p-n_p^{(s)}-n_b^{(s)}+1} \right)^{\top}, \tag{11}$$

the ARX model (9) can be recast as

$$y_p^{(\ell)} = \varphi_p^{(\ell)^{\top}} \vartheta^{(\ell)} + e_p^{(\ell)}. \tag{12}$$

Of course, once the past values on the scalar vector $\varphi_p^{(\ell)}$ are known and the white noise term $e_p^{(\ell)}$ is modelled, the computation of $y_p^{(\ell)} = y_p^{(\ell)}(\vartheta^{(\ell)})$ depends on the value of the scalar vector $\vartheta^{(\ell)}$, $\ell \in \{1, ..., 6\}$. Given this, assume that the values of $y_p^{(\ell)}$ are known (measured) for $p \in \{1, ..., N\}$ and denote such known quantities as $\bar{y}_p^{(\ell)}$, for $\ell \in \{1, ..., 6\}$. Then, the parameters vector $\vartheta^{(\ell)}$ can be obtained by solving a linear least squares problem,

$$\arg\min_{\vartheta^{(\ell)}} g(\vartheta^{(\ell)}) = \arg\min_{\vartheta^{(\ell)}} \frac{1}{N} \sum_{p=1}^{N} \left(\bar{y}_p^{(\ell)} - y_p^{(\ell)}(\vartheta^{(\ell)}) \right)^2. \tag{13}$$

Since the loss function $g(\vartheta^{(\ell)})$ in (13) is quadratic in $\vartheta^{(\ell)}$, a minimizer can be found requiring the vanishing of its gradient, thus leading to a system of normal equations that can be ill-conditioned when the dimension of the ${Z^{(\ell)}}^\top Z^{(\ell)}$ matrices is large, being $Z^{(\ell)}$ the matrix whose p-th row is ${\varphi_p^{(\ell)}}^\top$, $p \in \{1, ..., N\}$. For such a reason, Matlab uses of suitable built-in functions based on the QR factorization of the (rectangular) $Z^{(\ell)}$ matrix involved with the definition of $y^{(\ell)}(\vartheta^{(\ell)})$, to solve the over-determined linear system $y^{(\ell)}(\vartheta^{(\ell)}) = \bar{y}^{(\ell)}$ in the sense of least squares and, thus, estimating the parameters in (10) that minimize the loss $g(\vartheta^{(\ell)})$, for $\ell \in \{1, ..., 6\}$.

4.3.2 TFT-ARX comparison results

The performances of the TFT and ARX models are tested on a set of time intervals centred on the starting points of Table 5. The ARX parameters are calculated using as training sets the 96 hours before each starting point and then using the obtained ARX model to predict the six natural frequencies from each starting point up to the next 95 hours. For all the numerical tests, the orders $n_a^{(\ell)}$, $\{n_b^{(\nu)}\}_{\nu=1}^s$, $\{n_p^{(\nu)}\}_{\nu=1}^s$ of the linear combinations in (9) range in the interval $[0, 6]$. Table 5 summarises, for each frequency and each time interval listed in the first column, the optimal values g^* of the loss function at the end of the training phase, which mainly fall within the interval $[1.19 \cdot 10^{-5}, 8.64 \cdot 10^{-4}]$.

The prediction capabilities of the ARX and TFT models are compared in Table 6, highlighting that TFT outperforms ARX. Specifically, Table 6 reports the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) computed over the cumulative set of prediction hours given by the 95 hours from each of the starting points in Table 5. These metrics are widely used in regression problems and characterize the predictive capability of the developed system from the absolute (MAE, MSE) and relative (MAPE) perspectives. Remarkably, the metrics calculated by TFT are always better than the ARX counterparts and, in many cases, an order of magnitude below. In addition, to figure out representative plots about ARX and TFT predictions on the single frequencies, Fig. 21 is reported. The experimental values (blue) of the second and fourth frequencies of the Guinigi Tower are compared to the values predicted by the ARX (red) and TFT (black) models in the time interval centred on 15 October 2021, 00:00 a.m., confirming that TFT is more reliable in capturing the frequency evolution.

Starting point	g^*					
	f_1	f_2	f_3	f_4	f_5	f_6
10/10/21 6 a.m.	2.74e-05	4.67e-05	7.39e-05	2.74e-04	5.51e-04	4.66e-04
15/10/21 0 a.m.	1.88e-05	2.84e-05	3.06e-05	6.66e-04	1.78e-04	3.39e-04
25/10/21 6 p.m.	1.30e-05	4.49e-05	4.79e-05	1.38e-04	1.72e-04	2.42e-04
01/02/22 12 a.m.	6.15e-06	1.18e-05	1.74e-04	2.31e-04	1.85e-04	7.17e-04
04/02/22 4 p.m.	1.20e-05	9.00e-06	1.32e-03	2.68e-04	1.51e-04	1.61e-03
08/02/22 0 a.m.	2.74e-05	2.11e-05	6.43e-04	2.67e-04	6.77e-04	8.64e-04
12/02/22 4 a.m.	1.37e-05	1.91e-05	4.81e-04	3.09e-04	4.13e-04	7.93e-04

Table 5: Starting points and correspondent optimal values g^* of the loss function at the end of the ARX training phase, for each of the six natural frequencies.

	MAE		MSE		MAPE	
Frequency	ARX	TFT	ARX	TFT	ARX	TFT
f_1	1.51e-02	**7.96e-03**	6.17e-04	**1.27e-04**	1.25e-02	**6.57e-03**
f_2	1.19e-01	**1.57e-02**	1.33e-01	**3.92e-04**	8.78e-02	**1.16e-02**
f_3	1.29e-01	**3.98e-02**	1.24e-01	**3.86e-03**	4.84e-02	**1.48e-02**
f_4	4.78e-02	**2.78e-02**	1.15e-02	**1.35e-03**	1.44e-02	**8.35e-03**
f_5	9.30e-02	**6.73e-02**	2.65e-02	**7.06e-03**	2.23e-02	**1.60e-02**
f_6	7.63e-02	**3.58e-02**	3.33e-02	**2.53e-03**	1.21e-02	**5.64e-03**

Table 6: MAE, MSE and MAPE given by ARX and TFT models over the cumulative set of prediction hours related to the starting points in Table 5, for each of the six natural frequencies.

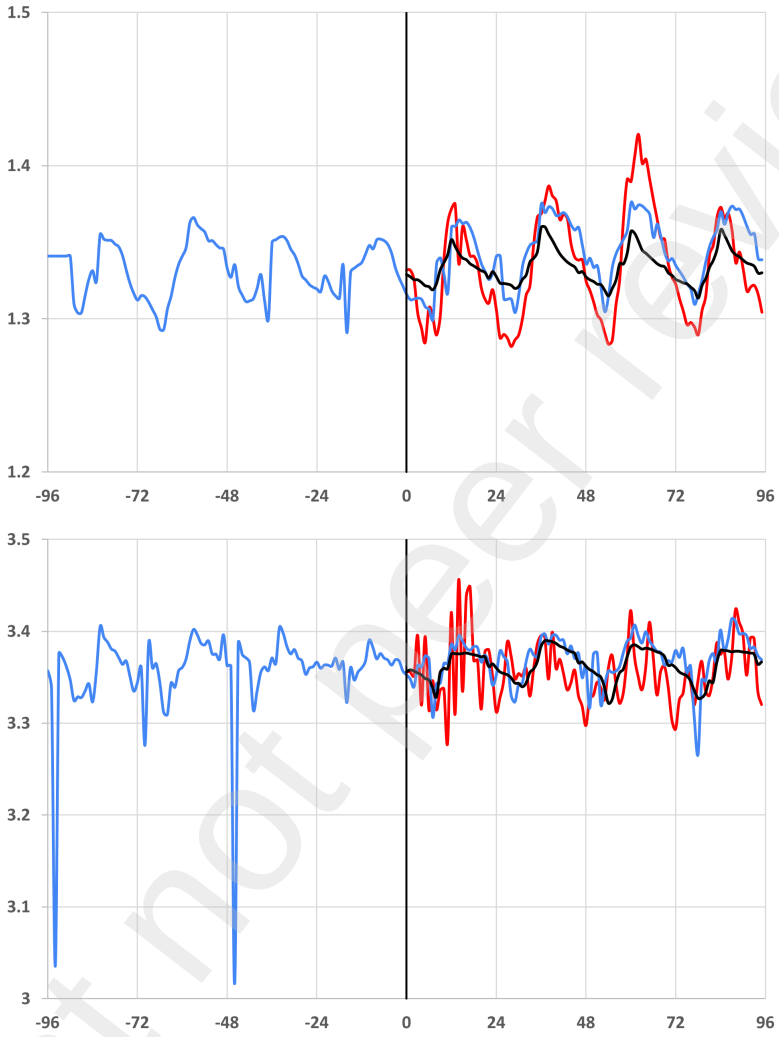

Figure 21: Second (top) and fourth (bottom) frequency [Hz] of the Guinigi Tower versus time [h]. Measured values (blue), ARX (red) and TFT (black) predicted values starting from 15 October 2021 at 0 a.m. (x=0).

5 Conclusions

This paper investigates the application of state-of-the-art time-series prediction models powered by transformer networks to the SHM of heritage structures. In particular, the Temporal Fusion Transformer model, an attention-based deep neural network architecture for multi-horizon time series forecasting, is employed to predict the dynamic properties of the Guinigi Tower in Lucca and detect anomalies in its behavior over time.

The paper represents the first detailed and comprehensive application of a TFT network to the SHM, already investigated by the Authors in 2022 in a preliminary research on the San Frediano bell tower in Lucca.

The approach adopted combines soundness and originality and relies on the theoretical setting of TFT, explicitly oriented to analyzing and predicting multivariate time series, such as the data sets collected during long-term continuous dynamic monitoring of historic buildings.

Anomaly detection has been performed on two damage indicators: the tower's natural frequencies and the measured signals' energy level. The paper has debated different aspects, from the procedure to automatically assemble a representative database to the influence of the various input variables and normalization techniques on the network's performance. The TFT predictions have been finally compared to those of the well-established ARX model. Such a comparison proves that the approach proposed by the Authors outperforms standard regression models regarding prediction accuracy.

Concluding, the use of TFT models seems well suited to SHM applications: the network proved to be sensitive to anomalies and accurate in the prediction. Moreover, once trained, the model can infer the results in a few seconds, and the training phase itself requires a few minutes to analyze a one-year monitoring database: these computation times are promising and suited to real-time applications.

Future work on the topic will focus on using the networks' internal features to discriminate among different anomaly categories (i.e., to distinguish severe structural damage from a temporarily overcrowded environment), which is of utmost importance for a fully automated pipeline for anomaly detection and characterization.

Acknowledgements. This research has been partially supported by the Italian National Research Council (REVOLUTION Project, Progetti di Ricerca @CNR, 2022–2024), by AI4Media - A European Excellence Centre for Media, Society, and Democracy (EC, H2020 n. 951911), and by PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 -"FAIR - Future Artificial Intelligence Research" - Spoke

1 “Human-centered AI”, funded by European Union - NextGenerationEU. Part of the research has been conducted within the framework of the STRENGTH project (fellowship programme “FSC Bando Assegni di Ricerca 2021”, funded by the Region of Tuscany and Infomobility S.r.l.). These supports are gratefully acknowledged.

References

[1] Trifunac M. D., Wind and microtremor induced vibrations of a twenty-two story steel frame building, Sc. Rep. of Earthquake Engineering Research Laboratory of California Institute of Technology, Pasadena, California 52 (1970).

[2] Brincker R., Ventura C., Introduction to operational modal analysis, John Wiley & Sons, 2015

[3] Gentile, C., Saisi, A., Ambient vibration testing of historic masonry towers for structural identification and damage assessment, Construction and building materials 21(6) (2007) 1311-1321.

[4] Ramos, L.F. , Marques, L. , Lourenço, P.B., De Roeck, G., Campos-Costa, A., Roque, J., Monitoring historical masonry structures with operational modal analysis: two case studies, Mechanical Systems and Signal Processing 24 (2010) 1291-1305.

[5] Ubertini, F. , Comanducci, G. , Cavalagli, N., Vibration-based structural health monitoring of a historic bell tower using output-only measurements and multivariate statistical analysis, Struct Health Monit. 15(4) (2016) 438-457.

[6] Ren, W. X., De Roeck, G., Structural damage identification using modal data. I: Simulation verification, Journal of Structural Engineering 128(1) (2002) 87-95.

[7] Ren, W. X., De Roeck, G., Structural damage identification using modal data. II: Test verification, Journal of Structural Engineering 128(1) (2002) 96-104.

[8] Teughels, A., De Roeck, G., Structural damage identification of the highway bridge Z24 by FE model updating, Journal of Sound and Vibration 278(3) (2004) 589-610.

[9] Gentile, C. , Guidobaldi, M. , Saisi, A., One-year dynamic monitoring of a historic tower: damage detection under changing environment, Meccanica 51(11) (2016) 2873-2889.

[10] Azzara R.M., De Roeck G., Girardi M., Padovani C., Pellegrini D., Reynders E., The influence of environmental parameters on the dynamic behaviour of the San Frediano bell tower in Lucca, Engineering Structures 156 (2018) 175-187.

[11] Facchini L., Betti M., Biagini P., Neural network based modal identification of structural systems through output-only measurement, Computers & Structures 138 (2014) 183-194.

[12] Farrar C.R., Worden K., Structural health monitoring: a machine learning perspective, John Wiley & Sons, 2012.

[13] Hughes A.J., Bull L.A., Gardner P., Barthorpe R.J., Dervilis N., Worden K., On risk-based active learning for structural health monitoring, Mechanical Systems and Signal Processing 167 (2022) 108569.

[14] Ibánez A.J.P., Bernal J.M.M., de Diego M.J.C., Sánchez, F.J.A., Expert system for predicting buildings service life under ISO 31000 standard. Application in architectural heritage, Journal of Cultural Heritage 18 (2016) 209-218.

[15] Lim J., Kim S., Kim H.K., Using supervised learning techniques to automatically classify vortex-induced vibration in long-span bridges, Journal of Wind Engineering and Industrial Aerodynamics 221 (2022) 104904.

[16] Magalhães F., Cunha Á., Caetano E., Vibration based structural health monitoring of an arch bridge: From automated OMA to damage detection, Mechanical Systems and signal processing 28 (2012) 212–228.

[17] Mishra M., Machine learning techniques for structural health monitoring of heritage buildings: A state-of-the-art review and case studies, Journal of Cultural Heritage 47 (2021) 227–245.

[18] Finotti R.P., de Souza Barbosa F., Cury A.A., Gentile C., A novel natural frequency-based technique to detect structural changes using computational intelligence, Procedia Engineering 199 (2017) 3314–3319.

[19] Avci O., Abdeljaber O., Kiranyaz S., Hussein M., Gabbouj M., Inman D.J., 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 147 (2021) 107077.

[20] Lim B., Zohren S., Time series forecasting with deep learning: a survey, Philosophical Transactions of the Royal Society A 379(2194) (2021) 20200209.

[21] Bao Y., Tang Z., Li H., Zhang Y., Computer vision and deep learning-based data anomaly detection method for structural health monitoring, Structural Health Monitoring 18(2) (2019) 401–421.

[22] Carrara F., Elias P., Sedmidubsky J., Zezula P., LSTM-based real-time action detection and prediction in human motion streams, Multimedia Tools and Applications 78 (2019) 27309–27331.

[23] Choi K., Yi J., Park C., Yoon S., Deep learning for anomaly detection in time series data: review, analysis, and guidelines, IEEE Access 9 (2021) 120043–120065.

[24] Munir M., Siddiqui S.A., Dengel A., Ahmed S., DeepAnT: A deep learning approach for unsupervised anomaly detection in time series, Ieee Access 7 (2018) 1991–2005.

[25] Rosso M.M., Cucuzza R., Marano G.C., Aloisio A., Cirrincione G., Review on deep learning in structural health monitoring, In Bridge Safety, Maintenance, Management, Life-Cycle, Resilience and Sustainability (2022) 309–315. CRC Press.

[26] Alaa A.M., Van der Schaar M., Attentive state-space modeling of disease progression, Advances in neural information processing systems 32 (2019).

[27] Rangapuram S.S., Seegar M.W., Gasthaus J., Stella L., Wang Y., Januschowski T., Deep state space models for time series forecasting, Advances in neural information processing systems 31 (2018).

[28] Torres J.F., Hadjout D., Sebaa A., Martínez-Àlvarez F., Troncoso A., Deep Learning for Time Series Forecasting: A Survey, Big Data 9(1) (2021) 3–21.

[29] Bai L., Yao L., Li C., Wang X., Wang C., Adaptive graph convolutional recurrent network for traffic forecasting, Advances in neural information processing systems 33 (2020) 17804–17815.

[30] Chen D., Chen L., Zhang Y., Wen B., Yang C., A multiscale interactive recurrent network for time series forecasting, IEEE Transactions on Cybernetics 52(9) (2021) 8793–8803.

[31] Connor J., Atlas L., Martin D., Recurrent networks and NARMA modeling, Advances in neural information processing systems 4 (1991).

[32] Ferreira A.A., Ludermir T.B., De Aquino R.R., Comparing recurrent networks for time series forecasting, In the 2012 International Joint Conference on Neural Networks (IJCNN) (2012) 1–8, IEEE.

[33] Gonon L., Grigoryeva L., Ortega J.P., Memory and forecasting capacities of nonlinear recurrent networks, Physica D: Nonlinear Phenomena 414 (2020) 132721.

[34] Jeon Y., Seong S., Robust recurrent network model for intermittent time series forecasting, International Journal of Forecasting 38(4) (2022) 1415–1425.

[35] Sagheer A., Kotb M., Time series forecasting of petroleum production using deep LSTM recurrent networks, Neurocomputing 323 (2019) 203–213.

[36] Salinas D., Flunkert V., Gasthaus J., Januschowski T., DeepAR: Probabilistic forecasting with autoregressive recurrent networks, International Journal of Forecasting 36(3) (2020) 1181–1191.

[37] Wen R., Torkkola K., Narayanaswamy B., Madeka D., A multi-horizon quantile recurrent forecaster, arXiv:1711.11053 (2017).

[38] Fan C., Zhang Y., Pan Y., Li X., Zhang C., Yuan R., Wu D., Wang W., Pei J., Huang H. (2019). Multi-horizon time series forecasting with temporal attention learning. In Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining, 2527–2535.

[39] Li S., Jin X., Xuan Y., Zhou X., Chen W., Wang Y. X., Yan X., Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, Advances in neural information processing systems 32 (2019).

[40] Gencay R., Liu T., Nonlinear modelling and prediction with feedforward and recurrent networks, Physica D: Nonlinear Phenomena 108(1–2) (1997) 119–134.

[41] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I., Attention is all you need, Advances in neural information processing systems 30 (2017).

[42] Koutnik J., Greff K., Gomez F., Schmidhuber, J., A clockwork rnn, In International conference on machine learning (2014) 1863–1871. PMLR.

[43] Neil D., Pfeiffer M., Liu S.C., Phased lstm: Accelerating recurrent network training for long or event-based sequences, Advances in neural information processing systems 29 (2016).

[44] Choi E., Bahadori M.T., Sun J., Kulas J., Schuetz A., Stewart W., Retain: An interpretable predictive model for healthcare using reverse time attention mechanism, Advances in neural information processing systems 29 (2016).

[45] Guo T., Lin T., Antulov-Fantulin N., Exploring interpretable lstm neural networks over multi-variable data, In International conference on machine learning (2019) 2494–2504. PMLR.

[46] Lundberg S.M., Lee S.I., A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[47] Ribeiro M.T., Singh S., Guestrin C., Why should I trust you? Explaining the predictions of any classifier, In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (2016) 1135–1144.

[48] Yoo H., Kim B., Kim J.W., Lee, J.H., Reinforcement learning based optimal control of batch processes using Monte-Carlo deep deterministic policy gradient with phase segmentation, Computers & Chemical Engineering 144 (2021) 107133.

[49] Dang H.V., Tran-Ngoc H., Nguyen T.V., Bui-Tien T., De Roeck G., Nguyen H.X., Data-driven structural health monitoring using feature fusion and hybrid deep learning, IEEE Transactions on Automation Science and Engineering 18(4) 2087–2103.

[50] Fernandez-Navamuel A., Magalhaes F., Zamora-Sánchez D., Omella A. J., Garcia-Sanchez D., Pardo D., Deep learning enhanced principal component analysis for structural health monitoring, Structural Health Monitoring 21(4) (2022) 1710–1722.

[51] Li J., Chen W., Fan G., Structural health monitoring data anomaly detection by transformer enhanced densely connected neural networks, Smart Structures and Systems 30(6) (2022) 613–626.

[52] Rosafalco L., Torzoni M., Manzoni A., Mariani S., Corigliano A., Online structural health monitoring by model order reduction and deep learning algorithms, Computers & Structures 255 (2021) 106604.

[53] Seventekidis P., Giagopoulos D., Arailopoulos A., Markogiannaki O., Structural Health Monitoring using deep learning with optimal finite element model generated data, Mechanical Systems and Signal Processing 145 (2020) 106972.

[54] Sun H., Song L., Yu Z. A deep learning-based bridge damage detection and localization method, Mechanical Systems and Signal Processing 193 (2023) 110277.

[55] Giglioni V., Venanzi I., Poggioni V., Milani A., Ubertini F. Autoencoders for unsupervised real-time bridge health assessment, Computer-Aided Civil and Infrastructure Engineering 38(8) (2023) 959-74.

[56] Lim B., Arik S.Ö., Loeff N., Pfister T., Temporal fusion transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting 37(4) (2021) 1748–1764.

[57] Carrara F., Falchi F., Girardi M., Messina N., Padovani C., Pellegrini D., Deep learning for structural health monitoring: An application to heritage structures, Materials Research Proceedings 26 (2023) 581–586.

[58] Peeters B., De Roeck G., Reference-based stochastic subspace identification for output-only modal analysis, Mechanical Systems and Signal Processing 13(6) (1999) 855–878.

[59] Pasca D.P., Aloisio A., Rosso M.M., Sotiropoulos S. PyOMA and PyOMA GUI: A Python module and software for Operational Modal Analysis, SoftwareX 20 (2022) 101216.

[60] Reynders E., Schevenels M., De Roeck G., MACEC 3.4: The Matlab toolbox for experimental and operational modal analysis. User's manual, Katholieke Universiteit, Leuven (2021).

[61] Croce T., Girardi M., Gurioli G., Padovani C., Pellegrini D., Towards a Cloud-Based Platform for Structural Health Monitoring: Implementation and Numerical Issues, In: Limongelli, M.P., Giordano, P.F., Quqa, S., Gentile, C., Cigada, A. (eds) Experimental Vibration Analysis for Civil Engineering Structures. EVACES 2023. Lecture Notes in Civil Engineering 432 (2023). Springer, Cham. https://doi.org/10.1007/978-3-031-39109-5_62.

[62] Reynders E., Houbrechts J., De Roeck G., Fully automated (operational) modal analysis, Mechanical systems and signal processing 29 (2012) 228–250.

[63] Mugnaini V., Fragonara L.Z., Civera M., A machine learning approach for automatic operational modal analysis, Mechanical Systems and Signal Processing 170 (2022) 108813.

[64] Rosso M.M., Aloisio A., Parol J., Marano G.C., Quaranta G., Intelligent automatic operational modal analysis, Mechanical Systems and Signal Processing 201 (2023) 110669.

[65] Van Overschee P., De Moor B., Subspace algorithms for the stochastic identification problem, Automatica 29(3) (1993) 649–660.

[66] Ren W. X., Zong Z.H., Output-only modal parameter identification of civil engineering structures, Structural Engineering and Mechanics 17(3–4) (2004) 429–444.

[67] Bartoli, G., Betti, M., Girardi, M., Padovani, C., Pellegrini, D., and Zini, G., Dynamic monitoring of a tunnel-like masonry structure using wireless sensor networks, Proceedings of the Institution of Civil Engineers-Structures and Buildings (2022) 1-12.

[68] Murtagh F., Contreras P., Algorithms for hierarchical clustering: an overview, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2(1) (2012) 86–97.

[69] Sasirekha K., Baby P., Agglomerative hierarchical clustering algorithm - A review, International Journal of Scientific and Research Publications 83(3) (2013) 83.

[70] Azzara R.M., Girardi M., Padovani C., Pellegrini D., From Structural Health Monitoring to Finite Element Modelling of Heritage Structures: The Medieval Towers of Lucca, Data Driven Methods for Civil Structural Health Monitoring and Resilience Latest Developments and Applications, Noori, Rainieri, Domaneschi, Sarhosis eds. ISBN 9781032308371 (2023) CRC Press.

[71] Azzara R.M., Girardi M., Padovani C., Pellegrini D., Experimental investigations and numerical modelling: a fruitful interaction for the nonlinear dynamical analysis of masonry structures, Continuum mechanics and thermodynamics S.I. : Nonlinear computational mechanics and related problems (2023) doi:10.1007/s00161-023-01264-2.

[72] Saisi, A., Gentile, C., Guidobaldi, M., Post-earthquake continuous dynamic monitoring of the Gabbia Tower in Mantua, Italy, Constr Build Mater. 81 (2015) 101-112.

[73] Baraccani, S., Azzara, R.M., Palermo, M., Gasparini, G., Trombetti, T., Long-term seismometric monitoring of the two towers of Bologna (Italy): modal frequencies identification and effects due to traffic induced vibrations, Frontiers in Built Environment 6 (2020) 85.

[74] Jolliffe, I. T., Cadima, J., Principal component analysis: a review and recent developments, Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences 374(2065) (2016) 20150202

[75] Ljung, L., System Identification, In: Procházka, A., Uhlìř, J., Rayner, P.W.J., Kingsbury, N.G. (eds). Signal Analysis and Prediction. Applied and Numerical Harmonic Analysis. Birkhäuser, Boston, MA (1998).