Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-024-01004-3

Revealing the structural behaviour of Brunelleschi's Dome with machine learning techniques

Stefano Masini[1] · **Silvia Bacci**[2] · **Fabrizio Cipollini**[2] · **Bruno Bertaccini**[2]

Received: 1 July 2023 / Accepted: 2 January 2024

Abstract
The Brunelleschi's Dome is one of the most iconic symbols of the Renaissance and is among the largest masonry domes ever constructed. Since the late 17th century, first masonry cracks appeared on the Dome, giving the start to a monitoring activity. In modern times, since 1988 a monitoring system comprised of 166 electronic sensors, including deformometers and thermometers, has been in operation, providing a valuable source of real-time data on the monument's health status. With the deformometers taking measurements at least four times per day, a vast amount of data is now available to explore the potential of the latest Artificial Intelligence and Machine Learning techniques in the field of historical-architectural heritage conservation. The objective of this contribution is twofold. Firstly, for the first time ever, we aim to unveil the overall structural behaviour of the Dome as a whole, as well as that of its specific sections (known as webs). We achieve this by evaluating the effectiveness of certain dimensionality reduction techniques on the extensive daily detections generated by the monitoring system, while also accounting for fluctuations in temperature over time. Secondly, we estimate a number of recurrent and convolutional neural network models to verify their capability for medium- and long-term prediction of the structural evolution of the Dome. We believe this contribution is an important step forward in the protection and preservation of historical buildings, showing the utility of machine learning in a context in which these are still little used.

Keywords Artificial intelligence · Cultural heritage preservation · Dimensionality reduction techniques · Forecasting, Multivariate time series · Sensor data

Responsible editor: Eamonn Keogh.

Extended author information available on the last page of the article

Published online: 06 February 2024

1 Introduction

The Italian artistic and architectural heritage represents a cultural and economic resource of inestimable value that must be safeguarded and conserved. The city of Florence reached its economic and cultural zenith during the 15th and 16th centuries under the rule of the Medici family. That was a period of exceptional artistic activity, thanks to which Florence is nowadays known throughout the world as the cradle of the Italian Renaissance.

One of its most representative and famous monuments is the cathedral of Santa Maria del Fiore (Fig. 1), whose history is rich and complex but not of interest to the purposes of this article.

Suffice it to know that by 1418 the only remaining task for the cathedral's completion was the construction of the dome. However, the challenge was considerable as no one at the time knew how to build a dome of that size, given that it was to be even larger than the Pantheon's Dome in Rome and that no dome of that size had been built since antiquity. On August 20th, 1418, a special competition was announced for the construction of the dome: the project presented by Filippo Brunelleschi won the competition.

Composed of over 4 million bricks, standing 116 ms tall, and weighing 37,000 tons, Brunelleschi's Dome was hailed as the greatest architectural achievement in the Western world upon its completion. Because of its octagonal supporting tambour, the dome presents eight "slice" webs rising in height and joined at the center by a summit lantern, also designed by Brunelleschi. Even today, it remains the largest masonry dome ever constructed, with a special curvature which differed from the hemispherical domes that had been built up to that time.

The left panel of Fig. 2 reports the plan of the cathedral with the webs of the Dome traditionally numbered counter-clockwise starting from the one facing the nave. In the right panel of the same figure a vertical section of the Dome is represented. From an architectural point of view, the vault consists of two nested domes

Fig. 1 The Florentine Cathedral of Santa Maria del Fiore with Brunelleschi's Dome

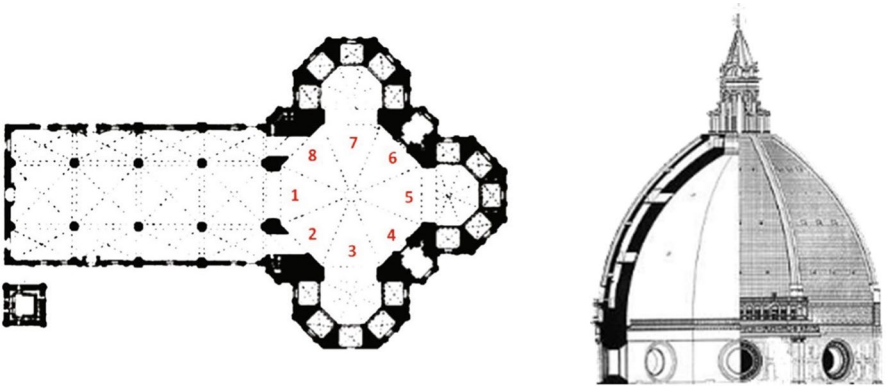

Fig. 2 Plan of the cathedral of Santa Maria del Fiore with the numbering of the slice webs of the Dome in evidence (left panel), and vertical section of the Dome (right panel; view from web 4)

(i.e., an internal dome and an external dome) separated by a walkway leading to the lantern. The maximum diameter of the internal dome is 45.5 m, while that of the external one is 54.8 m. A series of ingenious intuitions allowed Brunelleschi to close the dome building site in just sixteen years. The main one was definitely the brick laying strategy known as "Brunelleschi's herringbone pattern", a self-balanced construction technique that allowed him to set up a construction site suspended from the ground without shoring (Corazzi and Conti 2011; Paris et al 2020).

Unfortunately, cracks began to appear in the masonry of the Dome a few decades after its completion, at the end of the 15th century. In 1695, the Grand Duke of Tuscany established a first commission with the task of investigating the stability of the Dome: in its final relation, Vincenzo Viviani stated that "the weight of the upper dome and of the lantern exceeds the resistance of the (tambour at the) base of the monument" (Galluzzi 1977). Over the centuries, there have been other interpretations regarding the origin of the crack patterns that today involves all webs, but mainly webs 4 and 6, both opposite the nave (Corazzi and Conti 2011).

Therefore, the presence of cracks in the Dome has always been a source of concern for the preservation of the monument. For this reason, among the other world records attributed to it, Brunelleschi's Dome is also one of the most monitored architectural monuments in the world. Currently, the Dome is endowed with two main monitoring networks: a detection system based on 22 mechanical devices installed in 1955, and another detection system based on 166 electronic instruments installed in 1987, recently integrated by other instruments (including an anemometer and a seismograph). In particular, all of the mechanical devices and most of the electronic instruments are deformometers installed across the cracks to measure their width. Other devices such as thermometers and piezometers complete the monitoring system (Ottoni et al 2010).

Managing and analysing sensor data is often challenging (Nieto et al 2021). This is particularly true in the present case, because over time the monitoring system has produced a huge amount of data highlighting a sort of "breathing mechanism" of the vault: namely, the cracks tend to expand and contract cyclically according to

seasons and their relative temperatures, some in a harmonious relationship, others in the opposite way (Ottoni and Blasi 2014; Bertaccini 2015). Therefore, the Dome may be assimilated to what in physics is defined as a "closed system": the structural constraints define the relationship of forces among the various cracks, which in turn are affected by the action of the "surrounding" environment (Bertaccini 2015). Nevertheless, the statistical analyses carried out to date have generally been conducted on a single or on a limited set of devices, showing a progressive increase in the size of the main cracks and, at the same time, a clear relationship with meteorological and seismic variables (Bartoli et al 1996; Ottoni et al 2010).

Since the behaviour of the Dome as a whole (or of its wide masonry surfaces) cannot be fully understood from single sensor measurements alone, a more comprehensive analysis than what has been carried out so far is necessary. Recently, Bertaccini (2015) and Bertaccini et al (2020) – fully aware that the structural behaviour of the Dome and its response to the surrounding environment are "hidden" within the data collected by the sensors – tried to model the complex relationships between endogenous and exogenous variables involved in the monitoring process in order to describe the underlying structural (latent) behaviour of the Dome. In the light of the nature of the variables involved (mainly, multivariate time series data with exogenous and endogenous variables), both Bertaccini (2015) and Bertaccini et al (2020) relied on the estimation of latent variables models (Bartholomew et al 2011), as structural equation models (Hox and Bechger 1998; Bollen et al 2008) and dynamic factor models (Pena and Yohai 2016; Asparouhov and Muthén 2018). Both of these studies aimed to assess the possibility of developing software tools for predicting the structural evolution of the monument. Unfortunately, the vast amount of data and the numerous model parameters led the authors to limit their proposals, both in space (a single web; Bertaccini 2015) and time (measurements gathered in a specific year; Bertaccini et al 2020).

The need to develop a forecasting software that may help in suggesting possible improvement policies for the safety and stability of the Dome represents the main objective of a project funded by the National Research Center in High Performance Computing, Big Data and Quantum Computing foreseen within Mission 4 (Education and Research) of the "National Recovery and Resilience Plan" (NRRP) that is part of the Next Generation EU (NGEU) program (https://www.italiadomani.gov.it/en/). Within this project, the present contribution aims at evaluating the potentialities of some machine learning (ML) techniques to overcome the limits of the above mentioned studies. In the intention of the authors, these ML algorithms should form the core of a specific software for the prediction of the structural evolution of the monument both as a whole and in its specific elements (i.e., its eight slice webs). Such software will integrate the current monitoring system through a real-time acquisition of the measurements of its sensors.

The following two main Research Goals drive the selection of the best set of ML techniques:

Goal1 (Description). We aim at describing the overall (latent) structural behaviour of a wide masonry surface, like a single web or the entire dome vault, using a set of measurements collected over an

extended period of time by multiple deformometers installed across cracks on that surface. We also aim at characterising the structural behaviour of such a masonry surface in relation to the different weather seasons.

Goal2 (Prediction). Taking into account that the Dome is a "closed system" strongly affected by the weather seasons, we aim at making medium- and long-term forecasts of the structural evolution of the various webs and, consequently, of the entire vault.

Among the various ML approaches, we focus on some dimensionality reduction techniques (Schölkopf et al 1997; Tenenbaum et al 2000; van der Maaten and Hinton 2008) to reach Goal1 and on some recurrent and convolutional neural network models (Hochreiter and Schmidhuber 1997) to achieve Goal2.

To the best of our knowledge, this is the first time that ML techniques are combined to analyse multiple sensor data in the context of historic-architectural heritage protection, with explanatory and predictive purposes. Indeed, the application of ML in the field of preserving cultural and historical heritage is a research area that is yet to mature, primarily due to the scarcity of large datasets. Among the first contribution, Zhu et al (2011) introduced a novel distance measure and algorithms which allow efficient and effective data mining on human-made carved markings on stone, known as petroglyphs. Recently, Fiorucci et al (2020) proposed a wide review of works involving application of ML to cultural heritage data, outlining how the most active areas of study relate to archaeological artefacts (e.g., inferring the intensity of human use of ancient potteries, classifying ceramic artefacts based on their chemical composition) and paintings (e.g., detecting fake artworks, attributing authorship to various artworks, classification of artworks into different artistic categories). Another interesting review contribution is due to Mishra (2021) that focused on applications of ML techniques for the structural health monitoring of heritage buildings. This review discusses several applications (see references therein) of ML approaches for detecting and forecasting cracks in masonry structures and stone monuments, as well as evaluating their remaining lifespan. Such studies relied on various data sources, including laboratory testing, non-destructive testing, high-resolution images, and simulation models. A different stream of the literature focuses on statistical methods and/or ML techniques used to analyse sensor data collected to monitor the health of an architectural feat. Vespier et al (2011) modelled the structural health of a major Dutch highway bridge by clustering time series data collected by the monitoring system installed on it. More recently, Xu et al (2023) and Gomez-Cabrera and Escamilla-Ambrosio (2022) proposed an interesting review of the current machine-learning algorithms implemented respectively in concrete and steel bridge and in buildings structural health monitoring systems, evaluating their effectiveness in detecting damages caused by material deterioration. Abbas et al (2023) presented a study for underground metro shield tunnels, using a deep

learning auto-encoder to detect structural damages by incorporating raw vibration signals.

The rest of the paper is structured as follows. Section 2 describes the data and provides insights into the engineering activity necessary to make the dataset more suitable for data visualization and for the ML techniques being used. Section 3 illustrates results obtained through the application of some dimensionality reduction techniques. A particular focus is devoted to describe the structural behaviour of wide masonry surfaces starting from the measurements collected by the sensors installed on them. In Sect. 4 these results are used to predict the future behaviour of each web based on the application of some ML time series forecasting techniques. In Sect. 5, some concluding remarks end the contribution.

All the analyses conducted in this contribution have been performed with R (version 4.2.1) and Python (version 3.7.8) languages integrated with the ScikitLearn (version 1.0.2; Pedregosa et al 2011) and TensorFlow (version 2.9.1; Abadi et al 2015) libraries.

2 Data description, reduction and reconstruction

Data used in this work (kindly provided by the "Opera del Duomo" Foundation) have been acquired by the electronic sensors composing the Brunelleschi's Dome monitoring system, which were installed in 1987 and started recording in January 8th, 1988.

The main part of such instruments is represented by 57 deformometers, which are special sensors for highly repeatable automated measures, generally used in the field of deformability tests. On the Dome, deformometers are installed on the walls across cracks to measure their width, mostly on the even webs due to their worst crack patterns. As shown in Fig. 3, during installation this type of sensor is calibrated on value 0; subsequently, it records negative values when the crack widens, and positive values when the crack narrows.

The width of the cracks is clearly related to the action of the environment; in particular, the masonry volume tends to expand and shrink cyclically according to seasons and related temperatures: the higher the temperature, the greater the masonry volume (and vice-versa). For most cracks, an increase in the volume of the masonry corresponds to a contraction of their width (which, as mentioned above, causes the relative deformometer to register a positive value). However, not all cracks exhibit the same behaviour over time. Some cracks show a contrary pattern, widening when temperatures rise. This is due to the fact that the Dome is comparable to a "closed system", where structural constraints define the relationships of forces between the various cracks (Bertaccini 2015). For this reasons, a set of thermometers are installed on each web of the Dome to measure the air and the masonry temperature, for a total of 13 air thermometers and 47 masonry thermometers; see Table 1 for the distribution of masonry thermometers and deformometers on each web. The other sensors that complete the monitoring system are piezometers, plumb lines, telecoordinometers and (but only since a few years) an anemometer and a seismograph.

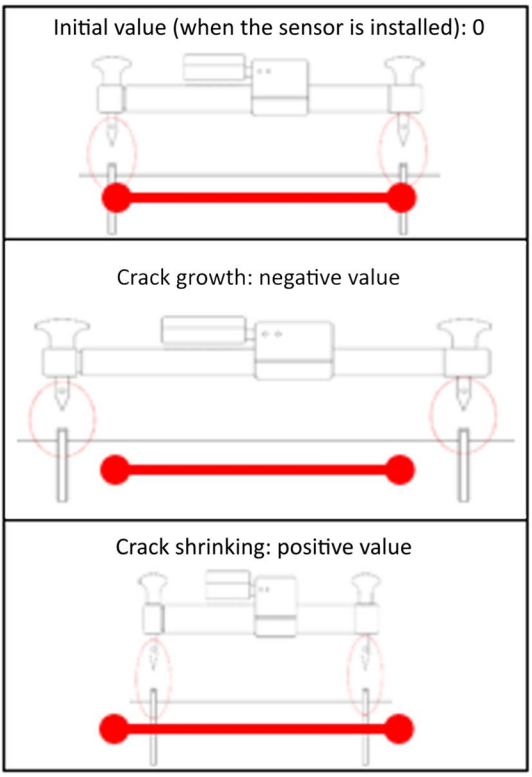

Fig. 3 How a deformometer works

Table 1 Number of masonry thermometers and deformometers installed on each web

Web	Masonry therm	Deformom
1	3	6
2	12	10
3	8	3
4	3	12
5	3	3
6	3	10
7	12	3
8	3	10
Total	47	57

Since its installation, the electronic part of the Dome monitoring system has been recording data at least every 6 h, resulting in a large database of measurements. Moreover, some sensors have broken and have been replaced (resetting their initial values), others have been added over time. For this reason, all the analyses conducted and illustrated in this paper are limited to the data acquired from January 1st, 1997 (when all the current deformometers were active) to February 28th, 2017. Data

were subsequently reduced computing the average daily measurements for each sensor. This approach was taken because the fluctuations in crack amplitude and temperature within the 24 h were considered irrelevant (Bertaccini 2015). This process resulted in a conclusive dataset comprising 7364 observations per sensor.

Unfortunately, electronic sensors are susceptible to temporary faults, mainly due to storms and blackouts. Deformometers, in particular, can sometimes be thrown out of calibration by thunderbolts, which can cause anomalous oscillations or full-scale values. Such outliers, in view of their lack of information power, were deleted and treated as missing data. A specially designed statistical method based on the estimation of a quadratic-sinusoidal regression model per sensor was used to impute missing values (Bertaccini 2015) and complete the data matrix.

3 The Dome dynamics

In this section, for the first time ever, we are able to describe the (latent) structural evolution of the Dome by using the average daily measurements of cracks. We start by exploring the existence of seasonal periodicity in the measurements acquired by deformometers, and we then investigate how to suitably synthesize these measures to characterise the overall structural behaviour of the eight webs. More specifically: Sect. 3.1 is devoted to compare the ability of alternative unsupervised ML techniques to synthesize the dynamics of the entire Dome on the basis of all 57 deformometers; the approach selected in this section is then applied in Sect. 3.2 to synthesize the dynamics of each web.

3.1 Seasonality in the evolution of the crack patterns

To reduce the dimensionality of data we evaluated the effectiveness of some ML techniques capable of capturing and preserving both linear and non-linear relationships among the deformometer measures: Kernel Principal Component Analysis (KPCA; Schölkopf et al 1997), Isometric mapping (Isomap; Tenenbaum et al 2000), and *t*-distributed Stochastic Neighbor Embedding (*t*-SNE; van der Maaten and Hinton 2008).

Principal Component Analysis (PCA; Hotelling 1933; Härdle and Simar 2015) is a widely used technique for dimensionality reduction that identifies the directions of maximum variance in the data under the assumption that data are linearly correlated. PCA projects the data onto a smaller number of orthogonal dimensions (principal components) that capture the most important information while minimizing the loss of variability. Kernel PCA (KPCA) is an extension of PCA, an unsupervised learning algorithm that can capture non-linear relationships between variables by mapping data into a higher-dimensional feature space than the original one using a non-linear function (kernel), and then performing PCA on the mapped data. While it can be computationally demanding, KPCA can effectively model complex and non-linear relationships in the data. Performance of KPCA relies on some tuning parameters (hyperparameters), namely the *kernel* type (e.g., linear, polynomial, ...)

and the *gamma* space regularization parameter. Increasing the *gamma* value makes each instance's neighbours set smaller because it reduces the probability to select the farthest points. As an unsupervised algorithm, there is no naive measure for tuning KPCA hyperparameters, so we rely on minimizing the reconstruction error following Weston et al (2003). We evaluated 50 possible gamma values ranging from 0.1 to 5 and 4 different kernel types (linear, polynomial, gaussian radial basis function, sigmoid kernel), selecting as best hyperparameters a polynomial kernel and a gamma value equal to 0.1.

Isomap is one of the earlier solutions to manifold learning (Huo and Smith 2008). It is a dimensionality reduction algorithm that aims to preserve the global structure of the data looking for a low-dimensional representation that captures the intrinsic geometric relationships between data points. Isomap uses a graph-based approach to estimate the pairwise distances between data points, and then embeds the data points into a low-dimensional space using multi-dimensional scaling. The key idea behind Isomap is to model the geodesic distances (shortest paths along the manifold) between data points rather than their Euclidean distances in the high-dimensional space. By doing so, Isomap can capture non-linear relationships between variables and can reveal underlying structures that may not be apparent in the original data. This main characteristic makes Isomap suitable in a wide range of applications, including image recognition, speech recognition, and natural language processing. Isomap is based on only one hyperparameter: the number k of neighbors taken into account during the reduction operations. To ensure optimal performance, we rely on cross-validation to select the best value for this parameter. In our case, we evaluated different values of k and found that using 50 neighbors provided the best results for our data.

t-SNE is a manifold learning technique for visualizing high-dimensional data in a lower-dimensional space (usually 2D or 3D) that can preserve both local and global structure in the data. *t*-SNE is particularly effective at capturing non-linear relationships between variables and revealing clusters and patterns that may be difficult to detect with other techniques. To accomplish this, *t*-SNE searches for a subset of neighbors using a hyperparameter known as the perplexity parameter. This parameter can be thought of as similar to the number of neighbors parameter in the Isomap manifold. By selecting an appropriate value for the perplexity parameter, *t*-SNE can effectively capture the local structure of high-dimensional data, preserving both small and large-scale relationships between data points, revealing patterns and relationships that may not be apparent using other techniques. A higher value of perplexity parameter implies a higher probability for each data point to be included in the subset of neighbors of another data point. In this way one probability distribution is defined for each data point and the algorithm tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. Relying on cross-validation, we select a value of the perplexity parameter equal to 45.

The first two mapped dimensions obtained from the three aforementioned techniques are displayed in Fig. 4 (top panel: KPCA; middle panel: Isomap; bottom panel: *t*-SNE). All the plots represented in the figure are generated by assigning a color to each point, representing the synthesis of each daily observation based on

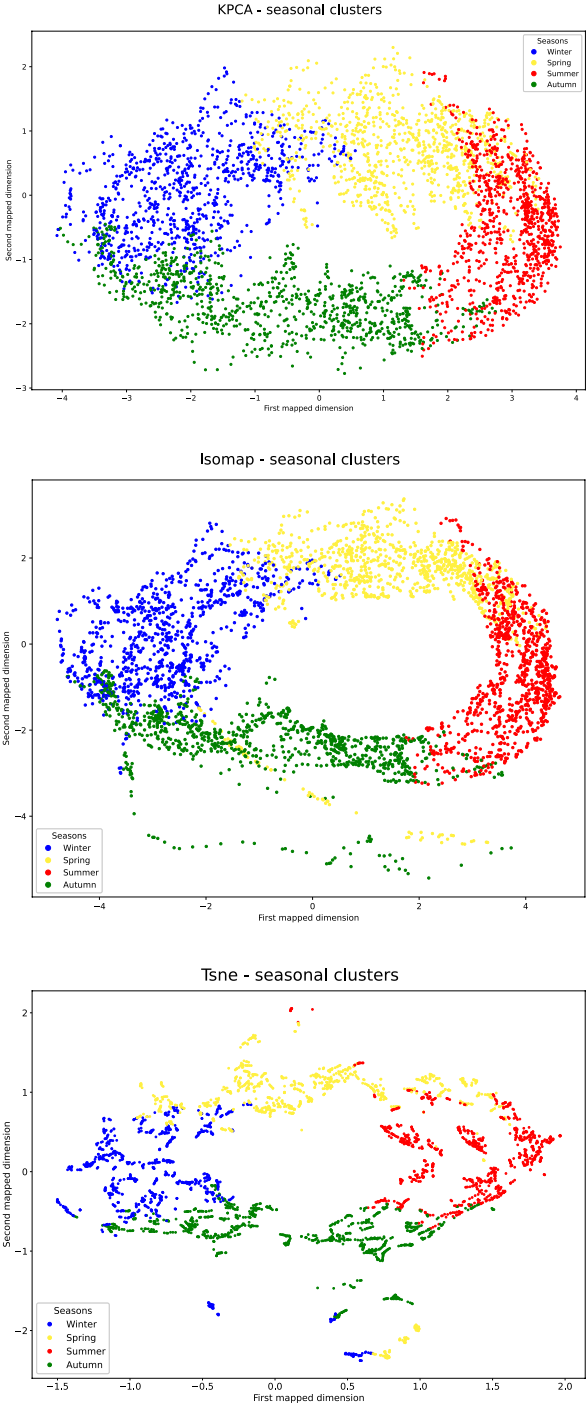

Fig. 4 Data visualization of seasonal clusters with KPCA (top panel), Isomap (middle panel) and *t*-SNE (bottom panel)

its corresponding astronomical season. Note that all the previously described techniques aim to capture the underlying structure of the data by finding a low-dimensional representation that preserves the maximum amount of variability observed in the original high-dimensional space. Consequently, the values of the coordinates in the low-dimensional space are arbitrary and can be rotated or mirrored without affecting the overall structure of the representation. Therefore, the sign of the extracted dimensions is not relevant in either aforementioned technique, and multiple runs of the same algorithms may produce different signs for the same coordinates. For these reasons, to facilitate the comparison among the three different techniques, the coordinates in the first two mapped dimensions have been standardised and always plotted according with the expected behaviour of the masonry during the seasons (as explained in Sect. 2). The different shapes of the three plots are due to the different way in which distances between points are calculated by the three adopted dimensionality reduction techniques: Euclidean distance in KPCA, geodesic distance in Isomap, and Kullback–Leibler divergence in *t*-SNE. However, the different distance metrics do not affect the interpretation of the results. Looking at the plots, we can see that all three techniques agree in outlining the seasonal periodicity of the time series of observed cracks. The clusters corresponding to winter and summer seasons are well separated and do not overlap, while the clusters corresponding to spring and autumn remain distinguishable but sometimes overlap with each other and with the clusters of summer and winter.

Table 2 shows the overall residual variance ratio computed for each of the three dimensionality reduction techniques adopted. The residual variance represents the portion of variance in the data that is not captured by the low-dimensional embedding, which, in this case, is limited to the first two dimensions.

In summary, all three applied techniques clearly indicate the existence of a strong association between the observed crack patterns and season-related variables. However, in general, *t*-SNE are better suited for high-dimensional data with complex structures such as images, text or gene expression data. On the other hand, Isomap can effectively capture the underlying structure of data that have a manifold structure (data that can be described as a continuous surface with a complex and variable shape) like time series data may have. KPCA is also particularly useful when there is a complex nonlinear relationship between the variables in the dataset and it is difficult to identify the underlying structure using linear methods like PCA. Time series data collected by the monitoring system of the Dome have this characteristic. Additionally, Isomap and *t*-SNE can be computationally expensive and harder to optimize than KPCA (Anowar et al 2021). Finally, limiting the dimensionality

Table 2 Overall residual variance ratio with respect to the first two dimensions

	First dimension	First two dimensions
KPCA	0.309	0.125
Isomap	0.568	0.160
t-SNE	0.393	0.254

reduction to the first two dimensions, KPCA results in a lower overall residual variance ratio compared to the other techniques. Therefore, we chose to continue our analysis using the first two dimensions extracted with KPCA.

3.2 Structural behaviour of single webs

To synthesize the dynamic behaviour of each web of the Dome, we performed eight independent KPCAs, one for each set of deformometers located on the same web. In Table 3, columns 2 and 3 report the hyperparameters selected as described in Sect. 3.1, while columns 4 to 6 show the amount of variance explained by the first three principal components.

As shown by the results in Table 3, the first two principal components explain a share of variance ranging from over 77% (web 3) to just under 99% (web 5). Based on this evidence, the dimensionality reduction analysis for each web is performed using only the first two principal components (PC1, PC2).

In PCA (and KPCA), the extracted components are orthogonal vectors, and the variance of their sum is equal to the sum of their respective variances. To capture as much variance as possible, we sum the first two principal components. Figure 5 shows the pattern of the sum of the first two principal components for each web (top panel: odd webs; bottom panel: even webs), together with the pattern of the daily average masonry temperatures (central panel). To ensure clarity, the figure specifically refers to a two-year time window.

Based on Fig. 5, webs in odd position exhibit similar sinusoidal movements, while webs in even position also follow a sinusoidal pattern, but with an opposite movements compared to the odd webs. The movement of the webs appears to be closely related to the temperature patterns in the masonry. When temperatures increase, cracks on even webs tend to narrow while cracks on odd webs tend to widen, and the opposite happens when temperatures decrease. A closer inspection of Fig. 5 reveals that there is some compensation in the movements of odd and even webs even if this behavior is not perfect. Specifically, the even webs exhibit a greater degree of fluctuation compared to the odd ones. Among the even webs (bottom panel), web 4 shows the widest fluctuations, followed by web 6, while

Table 3 Best KPCA hyperparameters and variance ratio explained by the first three principal components (PC1, PC2, PC3) for each web

	Hyperparameters		Explained variance ratio		
Web	Gamma	Kernel	PC1	PC2	PC3
Web 1	0.8	poly	0.496	0.364	0.088
Web 2	0.3	poly	0.514	0.385	0.067
Web 3	0.7	poly	0.436	0.340	0.224
Web 4	0.1	linear	0.563	0.267	0.114
Web 5	0.5	poly	0.857	0.130	0.013
Web 6	0.1	poly	0.669	0.205	0.100
Web 7	0.7	poly	0.477	0.349	0.174
Web 8	0.1	poly	0.588	0.235	0.099

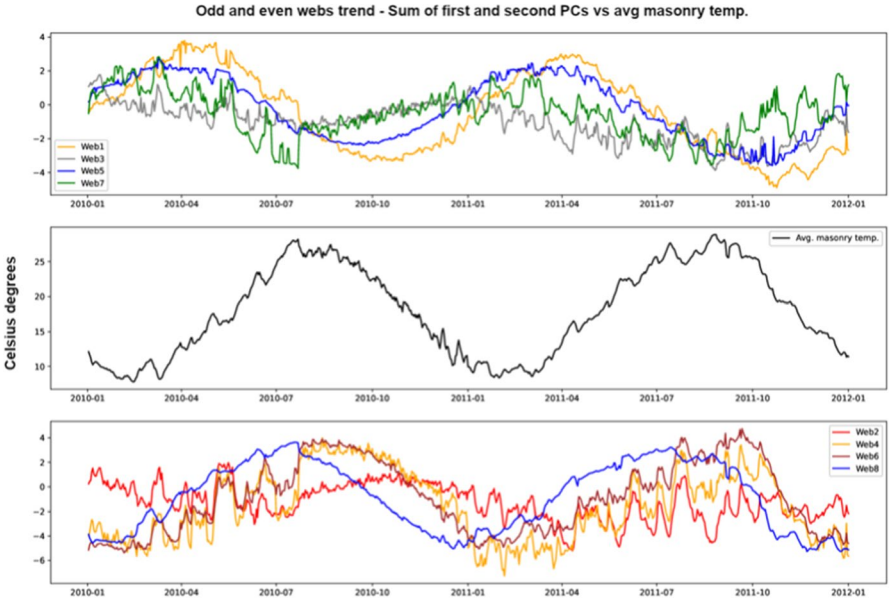

Fig. 5 Sum of the first two principal components of the eight webs (top panel: odd webs, bottom panel: even webs) and average masonry temperatures in Celsius degrees (central panel), along a two-years window (from January 1st, 2010 to January 1st, 2012)

web 2 exhibits the least variation. Among the odd webs (top panel), the patterns of webs 3 and 7 display a significant overlap and are relatively more constant compared to those of webs 1 and 5. Additionally, the pattern of web 1 appears to be somewhat delayed compared to the other webs. Trends observed in Fig. 5 are not limited to the specific two-year time window shown, but repeat with a similar yearly periodicity over the entire 20-year observation period (see Fig. 6).

These results provide insights for a "bellows movement" characterising the breathing mechanism of the Dome. Our study represents a significant departure from previous research (Bartoli et al 1996; Ottoni et al 2010; Ottoni and Blasi 2014; Bertaccini 2015) by revealing the critical role played by the webs in maintaining the structural stability of the entire monument. Even webs also reveal a deteriorating trend throughout the entire observation period, with webs 2 and 8 widening their cracks more than the others (see Fig. 6, bottom panel). On the other hand, only web 1 among the odd webs shows a significant deteriorating trend, while web 5 remains constant over time (Fig. 6, top panel).

In conclusion, the movements detected by individual sensors were not very relevant on their own. Moving from the goal of describing the structural behavior of wide masonry surfaces (*Goal1*), using ML techniques we reconstructed the (latent) dynamics of the whole Dome and of its eight webs, finding evidence of some compensating patterns that ensure the stability of the monument as a whole.

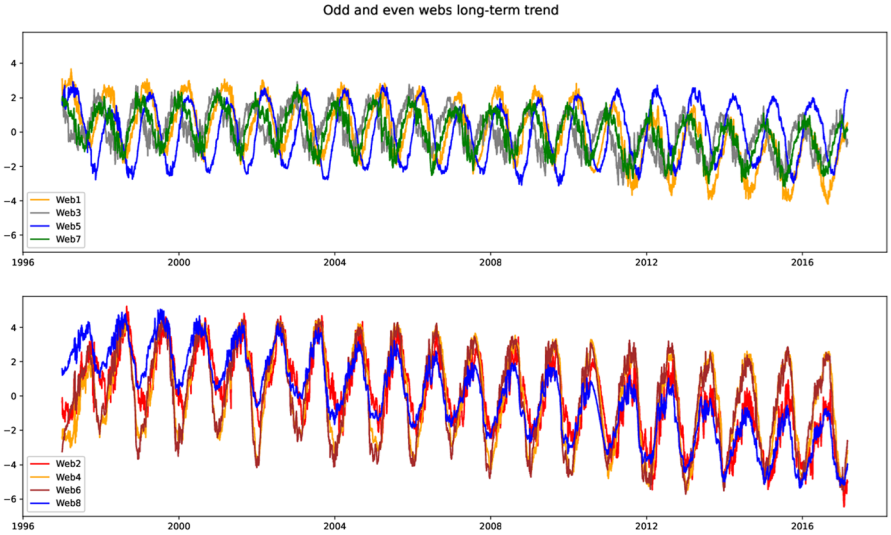

Fig. 6 Long-term trend of single webs

4 Forecasting

The dimensionality reduction analysis carried out in the previous section provided insights into the behavioural mechanism of the Dome. By the synthesis provided by the sum of the first two principal components of the eight webs, we gained a better understanding of the relationship between the observed cracks and season-related exogenous variables. Building on this understanding, we now shift our focus to predicting the future movements of each web.

The time series forecasting is a well-documented and extensively studied issue: scholars have developed numerous models and methodologies to deal with this goal, focusing on accuracy, efficiency, and adaptability to various types of data and forecasting horizons. In the ML literature, time series forecasting is typically approached pursuing single-step or multi-step forecasting. In single-step forecasting, the aim is predicting the value at the next time based on the preceding observations. This approach is used when the primary interest is in predicting the immediate future. In multi-step forecasting, the goal is instead predicting values relative to many future times ahead. This approach is applied when one is interested in forecasting a longer-term outlook or anticipate future patterns over an extended period. Both strategies have their own sets of techniques and algorithms tailored to their respective objectives. In both cases, models typically use a sliding window approach, where consecutive samples are used for training. By learning from these sequences, the models can identify patterns and relationships within the data, enabling them to make accurate predictions.

In this work, we address forecasting by employing Deep Neural Networks (DNNs, Goodfellow et al 2016). We emphasize that the aim of this research is not to introduce new ML models or techniques, but, rather, to optimize existing models for a particular application domain, specifically the preservation of cultural heritage.

In what follows, we outline the process of optimizing the DNN models to effectively address the unique challenges and requirements of this application domain and then present the results obtained.

4.1 DNN model selection and hyperparameter optimization

DNNs are commonly employed for predicting multivariate time series data. These algorithms leverage the hierarchical structure of neural networks with multiple hidden layers to capture complex patterns and dependencies in the data. Examples of DNN algorithms suitable for time series forecasting include, among others, Recurrent Neural Networks (RNNs), like Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU) networks, and Convolutional Neural Networks (CNNs); see, for instance, Lim and Zohren (2021) and Jacome et al (2022).

A RNN is a type of neural network that is designed to work with sequential data. In a RNN, information is passed from one step to the next through a latent state. This allows the network to maintain memory of the past inputs and use that information to make predictions about the future. A CNN is particularly effective in identifying spatial and temporal patterns in data and extracting relevant features. In a CNN, the input is passed through a series of convolutional layers, each of which applies a set of filters to the input to extract features. These algorithms have shown promising results in capturing temporal dependencies and making accurate predictions in various time series forecasting applications.

After a long preliminary process of selection, testing and optimization of many different models, we selected four different RNN and CNN models for each web. The models were trained using a sliding window approach, with consecutive samples shifted one day forward, from a training dataset that covered the initial 70% of the entire time series. The following 20% of the time series is used as a validation set and the last 10% makes up the test set. The prediction target was the sum of the first two principal components resulting from the dimensionality reduction procedure described in Sect. 3.2. Table 4 presents the structure of the dataset used for forecasting on web 2, serving as an example.

Selected models were trained using the Root Mean Squared Error (RMSE) as loss function and the Mean Absolute Error (MAE) as metric. The final layer of the models depends on the desired length of the prediction. All the trained models have an

Table 4 Structure of input dataset used in forecasting - Web 2

day	month	week	mason t	winter	spring	summer	autumn	PC1+PC2
1997-01-01	1	1	10.32	1	0	0	0	−0.1691
1997-01-02	1	1	9.12	1	0	0	0	−0.1147
1997-01-03	1	1	8.26	1	0	0	0	−0.2887
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2017-02-26	2	9	11.02	1	0	0	0	−1.2240
2017-02-27	2	9	11.14	1	0	0	0	−1.2197
2017–02-28	2	9	11.19	1	0	0	0	−1.1963

output layer of 100 time steps in the future (hyperparameter *output_size*); the number of features (*number_of_features*) for each web is always 8 (as shown in Table 4). Selected and trained models were:

- **LSTM classic model:** it is a classic LSTM model (Hochreiter and Schmidhuber 1997) with 16 LSTM cells layer followed by a hidden layer with *ouput_size* × *number_of_features* units.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 16)	1600
dense (Dense)	(None, 800)	13600
reshape (Reshape)	(None, 100, 8)	0

Total params: 15,200
Trainable params: 15,200
Non-trainable params: 0

- **LSTM bidirectional model:** the first layer is a bidirectional layer with 64 LSTM units (Schuster and Paliwal 1997) (Hochreiter and Schmidhuber 1997). The second layer is a hidden layer with *ouput_size* × *number_of_features* units.

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	(None, 128)	37376
dense (Dense)	(None, 800)	103200
reshape (Reshape)	(None, 100, 8)	0

Total params: 140,576
Trainable params: 140,576
Non-trainable params: 0

- **GRU bidirectional model:** the first layer is a bidirectional (Schuster and Paliwal 1997) layer with 64 GRU units (Cho et al 2014). The second layer is a hidden layer with *ouput_size* × *number_of_features* units.

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	(None, 128)	28416
dense (Dense)	(None, 800)	103200
reshape (Reshape)	(None, 100, 8)	0

Total params: 131,616
Trainable params: 131,616
Non-trainable params: 0

- **CONV GRU bidirectional model**: the first layer of the model consists of a convolutional layer with 40 filters, each having a size of 6x6. The convolutional layer is particularly effective in detecting patterns within long time sequences (Zhao et al 2018). Due to this capability, this model excels in long-term forecasting. The second layer is a bidirectional layer with 20 GRU units. The third layer is a hidden layer with 120 units and the fourth layer is a hidden layer with *ouput_size* × *number_of_features* units.

Layer (type)	Output Shape	Param #
reshape (Reshape)	(None, 300, 8)	0
conv1d (Conv1D)	(None, 148, 40)	1960
bidirectional (Bidirectional)	(None, 40)	7440
dense (Dense)	(None, 120)	4920
dense_1 (Dense)	(None, 800)	96800
reshape_1 (Reshape)	(None, 100, 8)	0

Total params: 111,120
Trainable params: 111,120
Non-trainable params: 0

4.2 Single- *vs* multiple-step forecasting

In single-step forecasting, a smaller window is typically required compared to multi-step forecasting because the objective is to predict the value at the next time step only. As a result, the model used in single-step forecasting is typically simpler compared to the models required for multi-step forecasting. Table 5 presents the evaluation results of the four trained models for each web. The evaluation was performed on the test set, which, as detailed above, consisted of fresh data corresponding to the last 10% of each time series. The model with the best performance in accomplish this task was the *GRU bidirectional model* with a sliding window of 30 consecutive time steps. For each web, RMSE and MAE computed for this particular model were consistently lower than the corresponding indices obtained by applying the other models. This indicates that the model outperformed the others in terms of accuracy and predictive performance for the given dataset.

In multi-step forecasting, a wider window is typically required due to the objective of predicting a longer-term future. The model needs to capture relationships that can only be observed over a substantial time interval. For this particular task, a window size of 300 consecutive time steps was defined to forecast the next 100 steps into the future. Similarly to the single-step forecasting, the evaluation was conducted on the test set, which consisted of fresh data corresponding to the last 10% of the time series. Table 6 displays the prediction error measures of the four trained models for each web: the *CONV GRU bidirectional model* emerged as the best performing model in each web, with a superior accuracy and predictive capability compared to the other models. Figures 7, 8, 9 and 10 in the Appendix show the forecasting results. To ensure accurate interpretation of the graphs, it is important to note that the y-axes of the plots differ between even and odd webs. Specifically, the oscillations of the odd webs (plots on the left) span from -1.00 to $+1.00$, while the

Table 5 Single-step forecasting: RMSE and MAE on test set

	Web 1	Web 2	Web 3	Web 4	Web 5	Web 6	Web 7	Web 8
LSTM classic model								
RMSE	0.033	0.092	0.034	0.049	0.088	0.049	0.053	0.045
MAE	0.031	0.064	0.026	0.042	0.078	0.069	0.041	0.022
LSTM bidirectional model								
RMSE	0.074	0.080	0.034	0.042	0.074	0.037	0.079	0.057
MAE	0.053	0.062	0.027	0.036	0.060	0.029	0.060	0.051
GRU bidirectional model								
RMSE	**0.031**	**0.075**	**0.031**	**0.040**	**0.071**	**0.037**	**0.049**	**0.029**
MAE	**0.025**	**0.052**	**0.025**	**0.035**	**0.051**	**0.025**	**0.037**	**0.019**
CONV GRU bidirectional model								
RMSE	0.082	0.091	0.049	0.065	0.082	0.046	0.082	0.065
MAE	0.083	0.063	0.036	0.056	0.050	0.050	0.070	0.061

Table 6 Multi-step forecasting: RMSE and MAE on test set

	Web 1	Web 2	Web 3	Web 4	Web 5	Web 6	Web 7	Web 8
LSTM classic model								
RMSE	0.231	0.658	0.235	0.598	0.377	0.483	0.355	0.480
MAE	0.121	0.567	0.198	0.602	0.324	0.341	0.194	0.362
LSTM bidirectional model								
RMSE	0.181	0.498	0.199	0.262	0.364	0.368	0.314	0.374
MAE	0.084	0.221	0.104	0.132	0.181	0.208	0.154	0.197
GRU bidirectional model								
RMSE	0.202	0.513	0.212	0.285	0.336	0.325	0.285	0.363
MAE	0.091	0.314	0.113	0.153	0.171	0.156	0.142	0.192
CONV GRU bidirectional model								
RMSE	**0.177**	**0.451**	**0.194**	**0.250**	**0.227**	**0.305**	**0.279**	**0.358**
MAE	**0.071**	**0.238**	**0.102**	**0.132**	**0.108**	**0.152**	**0.137**	**0.175**

oscillations of the even webs (plots on the right) cover a wider range, ranging from −3.00 to +2.00. This observation aligns with the univariate analysis conducted for each deformometer, indicating that the deformometers installed on even webs experience greater oscillation compared to those on odd webs.

5 Conclusions

In this study, we employed various Machine Learning (ML) techniques to analyse data collected from multiple sensors within the domain of historical-architectural heritage preservation. By applying ML techniques to the monitoring data obtained from Brunelleschi's Dome, we have successfully achieved two significant objectives.

First and foremost, our research has made a departure from previous studies that presented limitation in both spatial and temporal aspects. For the first time, we have been

able to unveil the comprehensive structural behaviour of the Dome as a whole. This has allowed us to gain a more holistic understanding of the Dome's structural dynamics, surpassing the limitations of previous investigations. This comprehensive understanding could never have been fully achieved solely through the analysis of individual sensors. By employing ML techniques, we were able to synthesize the movements of individual cracks, thereby highlighting the significant role played by the webs in maintaining the structural stability of the entire monument, through a compensative "breathing mechanism" between the webs in even position relative to those in odd position. Furthermore, the analysis has also outlined a close relationship between the masonry temperatures and the movements of the webs over time. Secondly, we used various DNNs to forecast the behaviour of each web, both single- and multi-step ahead. The results obtained from these approaches serve as foundation for the development of an active real-time monitoring and forecasting system. This system will undergo continuous training with new data and will have got the capability to identify anomalies in the Dome's movements over the medium term. It will provide valuable insights into the overall structural behaviour of the Dome, as well as its individual components, such as the webs.

Authors acknowledge some limitations of the present contribution that will be addressed in future works. The first limitation is that we did not consider the spatial location of the cracks on the surface of the Dome. Factors such as whether the cracks are on the internal or external surface, internal or external dome, or their proximity to the tambour or lantern were not taken into account. The second limitation is that we did not incorporate information related to exogenous variables beyond temperatures, such as wind or seismic movements happened in the past. Following the proposal of Palet et al (2023), an arbitrarily-high number of historical context variables could be integrated in the model to guide the learning task. Currently, only partial information regarding these additional variables is available.

Accordingly, in future works it would be interesting to enhance the dataset by including additional features to identify new relationships with the structural movements of the Dome, consequently improving the forecasting capability of ML algorithms. Moreover, we plan to investigate the performance of recent ML techniques based on the "attention" mechanism. These techniques have shown promising results in various time series contexts, although they require a very long time series window. Unfortunately, up to this point, with the available data limited to a 20 years monitoring window, we have been unable to apply these techniques. However, in future work, we aim to explore the potential of these attention-based ML techniques once we have access to a more extensive dataset.

Finally, as an alternative approach to the analysis of multiple time series, it might be interesting to consider the structural movements of the Dome as an evolving system. This may be tracked by Time Series Chains (Zhu et al 2019) or Novelets (Mercer and Keogh 2023), an emerging technique that allows detection of initially apparent anomalies that are later discovered to be previously unknown highly conserved behaviors.

Appendix - Multiple-step forecasts

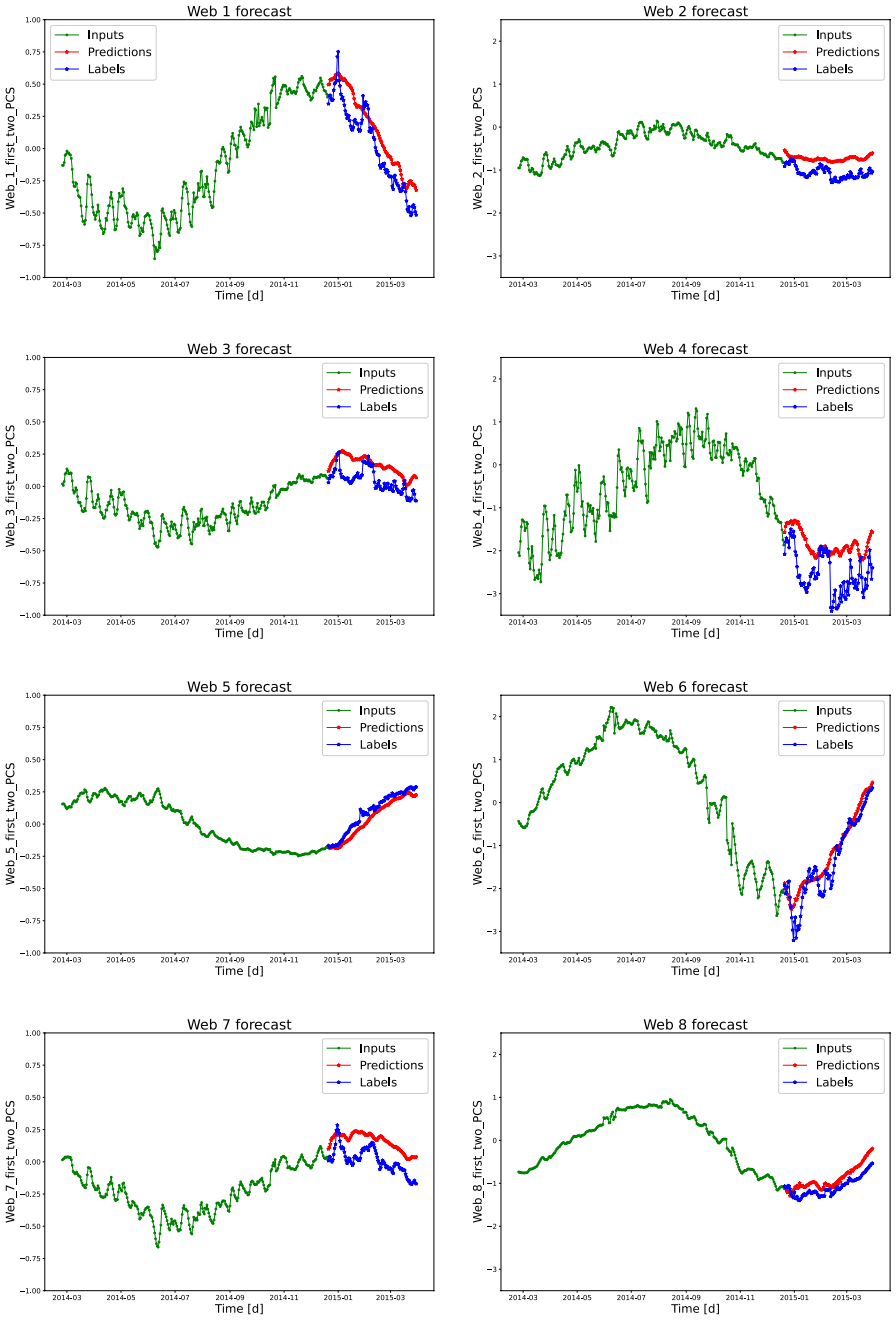

Fig. 7 Multi-step forecasting on first and second principal components - **winter season**: window size: 300, steps forward:100. Odd webs on the left, even webs on the right

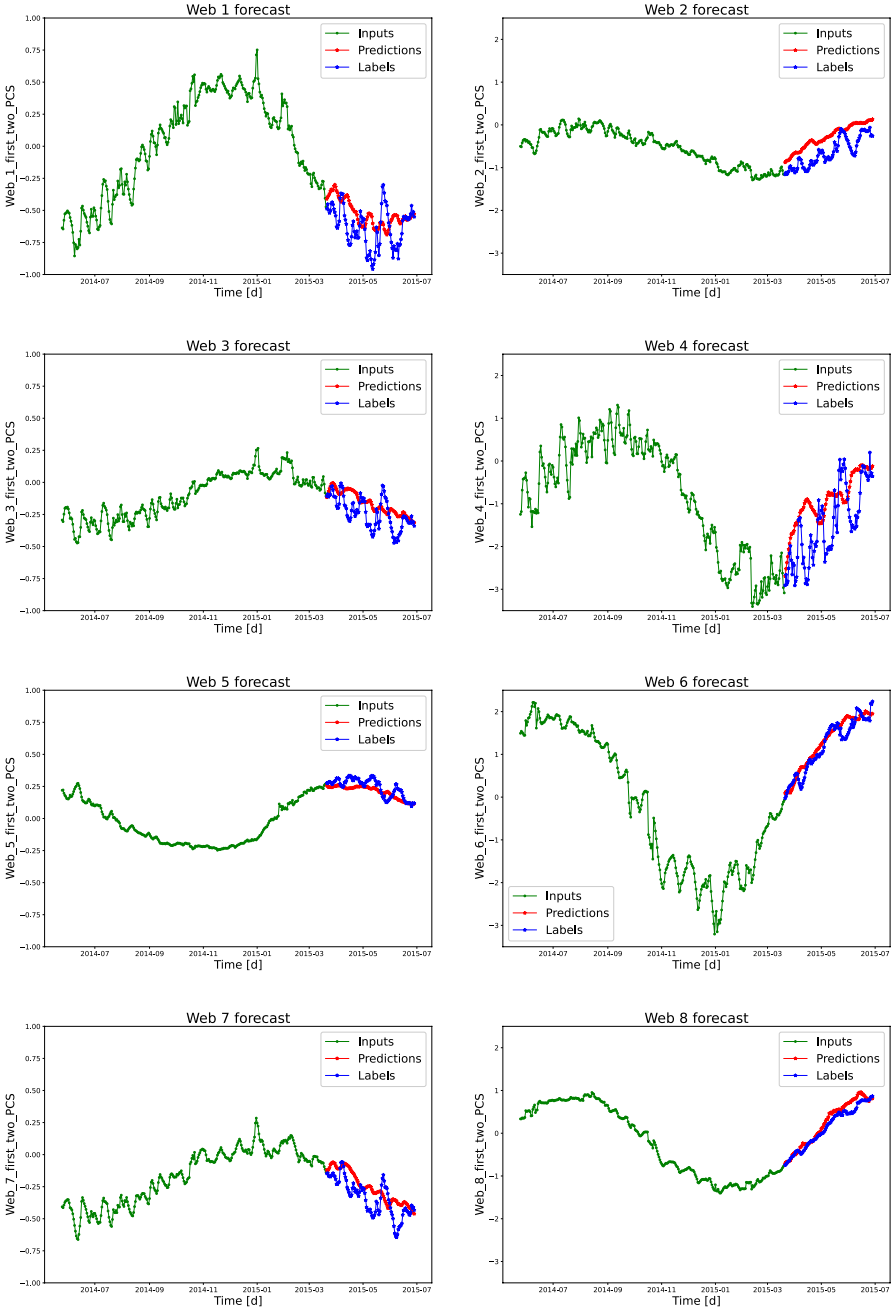

Fig. 8 Multi-step forecasting on first and second principal components - **spring season**: window size: 300, steps forward:100. Odd webs on the left, even webs on the right

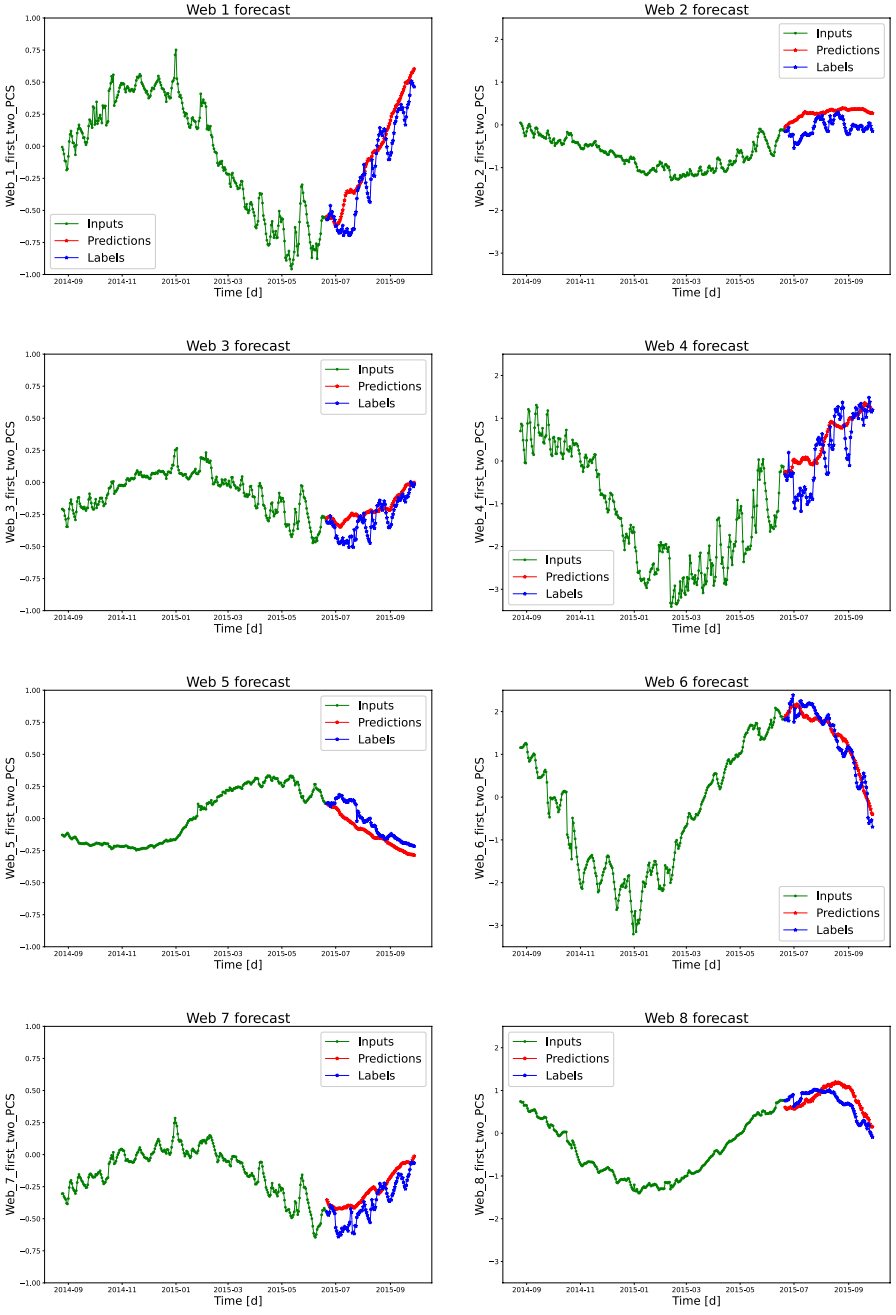

Fig. 9 Multi-step forecasting on first and second principal components - **summer season**: window size: 300, steps forward:100. Odd webs on the left, even webs on the right

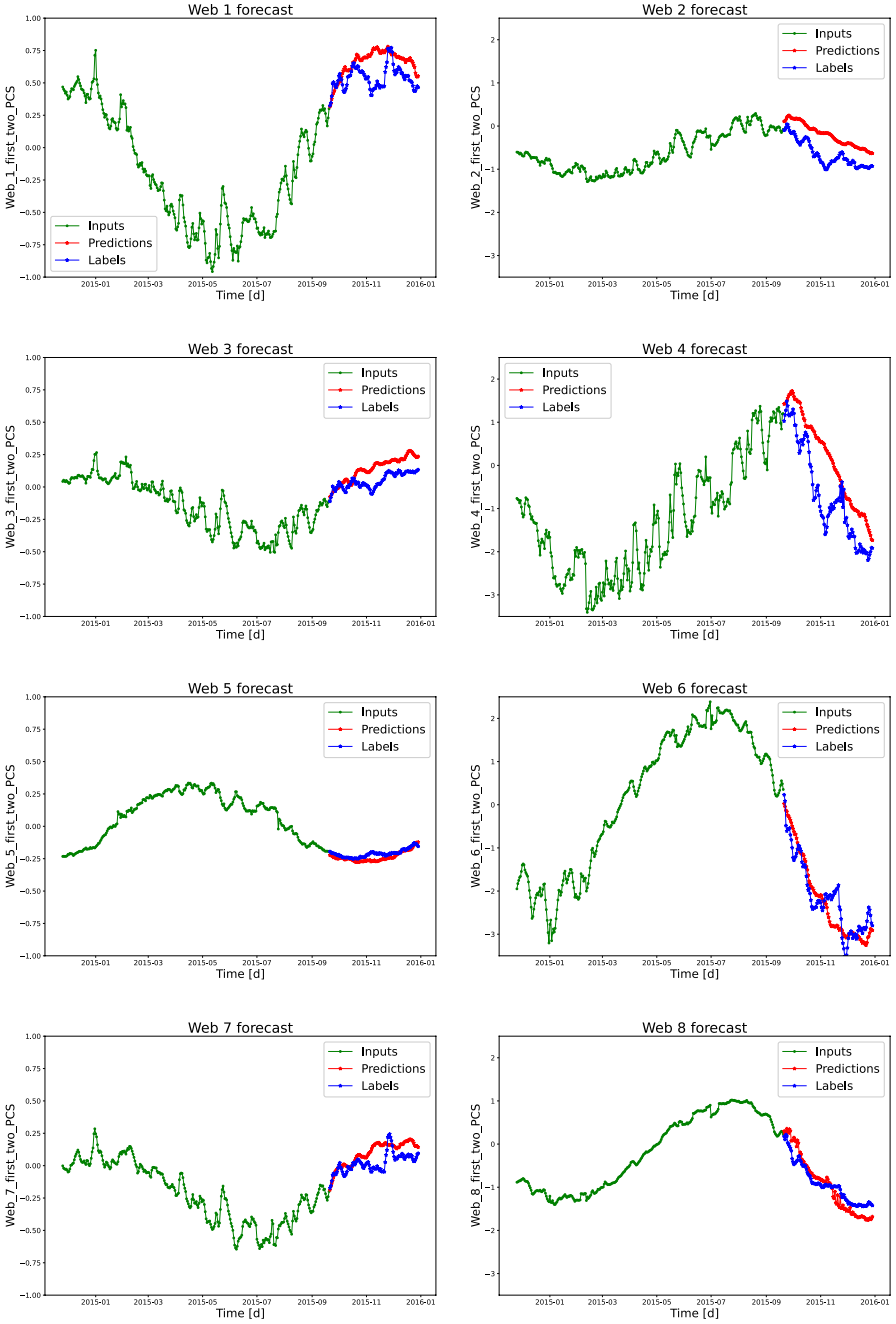

Fig. 10 Multi-step forecasting on first and second principal components - **autumn season**: window size: 300, steps forward:100. Odd webs on the left, even webs on the right

Acknowledgements Authors thank the "Opera del Duomo Foundation" for having making available the data.

Funding Open access funding provided by Università degli Studi di Firenze within the CRUI-CARE Agreement. The authors acknowledge the financial support of the National Research Center in High Performance Computing, Big Data and Quantum Computing foreseen within Mission 4 (Education and Research) of the "National Recovery and Resilience Plan" (NRRP) that is part of the Next Generation EU (NGEU) program https://www.italiadomani.gov.it/en/).

Declarations

Conflict of interest The authors declare the absence of competing interests.

References

Abadi M, Agarwal A, Barham P, et al (2015) TensorFlow: large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/, software available from tensorflow.org

Abbas N, Umar T, Salih R, et al (2023) Structural health monitoring of underground metro tunnel by identifying damage using ANN deep learning auto-encoder. Appl Sci 13(3). https://doi.org/10.3390/app13031332, https://www.mdpi.com/2076-3417/13/3/1332

Anowar F, Sadaoui S, Selim B (2021) Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). Comput Sci Rev 40(100):378. https://doi.org/10.1016/j.cosrev.2021.100378

Asparouhov EL, Muthén B (2018) Dynamic structural equation models. Struct Equ Model: Multidiscip J 25:359–388

Bartholomew DJ, Knott M, Moustaki I (2011) Latent Variable Models and Factor Analysis: A Unified Approach. John Wiley & Sons Ltd, Chichester, UK

Bartoli G, Chiarugi A, Gusella V (1996) Monitoring systems on historic buildings: Brunelleschi Dome. J Struct Eng 122(6):663–673. https://doi.org/10.1061/(ASCE)0733-9445(1996)122:6(663)

Bertaccini B (2015) Santa Maria del Fiore Dome behavior: Statistical models for monitoring stability. Int J Architect Herit 9(1):25–37. https://doi.org/10.1080/15583058.2013.774071

Bertaccini B, Bacci S, Crescenzi F (2020) A dynamic latent variable model for monitoring the Santa maria del Fiore Dome behavior, Lecture Notes in Computer Science, vol Computational Science and Its Applications - ICCSA 2020, Springer professional, pp 47–58. https://doi.org/10.1007/978-3-030-58811-3_4

Bollen KA, Rabe-Hesketh S, Skrondal A (2008) Structural equation models. In: BoxSteffensmeier JM, Brady H, Collier D (eds) The Oxford Handbook of Political Methodology. Oxford University Press, Oxford, pp 432–455

Cho K, van Merrienboer B, Gulcehre C, et al (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. https://doi.org/10.48550/ARXIV.1406.1078, https://arxiv.org/abs/1406.1078

Corazzi R, Conti G (2011) Il segreto della Cupola del Brunelleschi a Firenze—The Secret of Brunelleschi's Dome in Florence. Angelo Pontecorboli ed., 9788897080107, Firenze, IT

Fiorucci M, Khoroshiltseva M, Pontil M et al (2020) Machine learning for cultural heritage: a survey. Pattern Recognit Lett 133:102–108

Galluzzi P (1977) Le colonne fesse degli Uffizi e gli screpoli della Cupola : il contributo di Vincenzo Viviani al dibattito sulla stabilità della Cupola del Brunelleschi (1694-1697), Giunti Marzocco, Firenze, pp 71–111. Annali dell'Istituto e Museo di Storia della scienza di Firenze

Gomez-Cabrera A, Escamilla-Ambrosio PJ (2022) Review of machine-learning techniques applied to structural health monitoring systems for building and bridge structures. Appl Sci 12(21). https://doi.org/10.3390/app122110754, https://www.mdpi.com/2076-3417/12/21/10754

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, http://www.deeplearningbook.org

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Edu Psychol 24:417–441. https://doi.org/10.1037/h0071325

Hox J, Bechger TM (1998) An introduction to structural equation modeling. Family Sci Rev 11:354–373

Huo X, Smith A (2008) A survey of manifold-based learning methods. Recent Adv Data Min Enterprise Data. https://doi.org/10.1142/9789812779861_0015

Härdle WK, Simar L (2015) Applied Multivariate Statistical Analysis. Springer Berlin, 978-3-662-45171-7, Heidelberg, DE

Jacome R, Realpe M, Paillacho J, et al (2022) Time series in sensor data using state-of-the-art deep learning approaches: A systematic literature review, pp 503–514. https://doi.org/10.1007/978-981-16-4126-8_45

Lim, B., Zohren, S. (2021). Time-series forecasting with deep learning: a survey. Philosop Trans Royal Soc A 379:20200209. https://doi.org/10.1098%2Frsta.2020.0209

Mercer R, Keogh E (2023) Novelets: A new primitive that allows online detection of emerging behaviors in time series. Knowl Inf Syst pp 1–29. https://doi.org/10.1007/s10115-023-01936-0

Mishra M (2021) Machine learning techniques for structural health monitoring of heritage buildings: A state-of-the-art review and case studies. J Cult Herit 47:227–245

Nieto FJ, Aguilera U, de Ipiña DL (2021) Analyzing particularities of sensor datasets for supporting data understanding and preparation. Sensors 21(18):6063. https://doi.org/10.3390/s21186063

Ottoni F, Blasi C (2014) Results of a 60-year monitoring system for santa maria del fiore dome in florence. Int J Archit Herit 9:7–24. https://doi.org/10.1080/15583058.2013.815291

Ottoni F, Blasi C, Coisson E (2010) The crack pattern in Brunelleschi's Dome in Florence: damage evolution from historical to modern monitoring system analysis. Adv Mater Res 133–134:53–64. https://doi.org/10.4028/www.scientific.net/AMR.133-134.53

Palet J, Manquinho V, Henriques R (2023) Multiple-input neural networks for time series forecasting incorporating historical and prospective context. Data Min Knowl Discov. https://doi.org/10.1007/s10618-023-00984-y

Paris V, Pizzigoni A, Adriaenssens S (2020) Statics of self-balancing masonry domes constructed with a cross-herringbone spiraling pattern. Eng Struct 215(110):440. https://doi.org/10.1016/j.engstruct.2020.110440

Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Pena D, Yohai VJ (2016) Generalized dynamic principal components. J Am Statist Assoc 111:1121–1131

Schölkopf B, Smola A, Müller K (1997) Kernel principal component analysis. In: Gerstner W, Germond A, Hasler M, et al (eds) Artificial Neural Networks - ICANN 1997 - 7th International Conference, Proceeedings, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp 583–588, https://doi.org/10.1007/bfb0020217, publisher Copyright: © Springer-Verlag Berlin Heidelberg 1997.; 7th International Conference on Artificial Neural Networks, ICANN 1997 ; Conference date: 08-10-1997 Through 10-10-1997

Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Sign Proc 45:2673–2681

Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319

van der Maaten L, Hinton G (2008) Visualizing high-dimensional data using t-sne. J Mach Learn Res 9(nov):2579–2605. Pagination: 27

Vespier U, Knobbe A, Vanschoren J et al (2011) Traffic events modeling for structural health monitoring. Lect Note Comput Sci 7014:376–387. https://doi.org/10.1007/978-3-642-24800-9_35

Weston J, Schölkopf B, Bakir G (2003) Learning to find pre-images. In: Thrun S, Saul L, Schölkopf B (eds) Advances in Neural Information Processing Systems, vol 16. MIT Press, https://proceedings.neurips.cc/paper/2003/file/ac1ad983e08ad3304a97e147f522747e-Paper.pdf

Xu D, Xu X, Forde MC et al (2023) Concrete and steel bridge Structural Health Monitoring-Insight into choices for machine learning applications. Construct Build Mater 402(132):596 https://doi.org/10.1016/j.conbuildmat.2023.132596, www.sciencedirect.com/science/article/pii/S0950061823023127

Zhao H, Zarar S, Tashev I, et al (2018) Convolutional-recurrent neural networks for speech enhancement. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 2401–2405, https://doi.org/10.1109/ICASSP.2018.8462155

Zhu Q, Wang X, Keogh E et al (2011) An efficient and effective similarity measure to enable data mining of petroglyphs. Data Min Knowl Discov 23:91–127. https://doi.org/10.1007/s10618-010-0200-z

Zhu Y, Imamura M, Nikovski D et al (2019) Introducing time series chains: a new primitive for time series data mining. Knowl Inf Syst 60:1–27. https://doi.org/10.1007/s10115-018-1224-8

Authors and Affiliations

Stefano Masini[1] · Silvia Bacci[2] · Fabrizio Cipollini[2] · Bruno Bertaccini[2]

✉ Bruno Bertaccini
bruno.bertaccini@unifi.it

Stefano Masini
stefano.masini@unifi.it

Silvia Bacci
silvia.bacci@unifi.it

Fabrizio Cipollini
fabrizio.cipollini@unifi.it

[1] Department of Computer Science, University of Pisa, Largo B. Pontecorvo, 3, Pisa I-56127, Italy

[2] Department of Statistics, Computer Science, Applications "G. Parenti", University of Florence, v.le Morgagni, 59, Florence I-50134, Italy