Model-based vs. Data-driven Approaches for Anomaly Detection in Structural Health Monitoring: a Case Study

Amirhossein Moallemi
DEI, University of Bologna
Bologna, 40136, Italy
amirhossein.moallem2@unibo.it

Alessio Burrello
DEI, University of Bologna
Bologna, 40136, Italy
alessio.burrello@unibo.it

Davide Brunelli
DII, University of Trento
Trento, 38123, Italy
davide.brunelli@unitn.it

Luca Benini
DEI, University of Bologna
IIS, ETH Zurich
lbenini@iis.ee.ethz.ch

***Abstract*—Modern Structural Health Monitoring (SHM) systems are becoming of pervasive use in civil engineering because they can track the structural condition and detect damages of critical and civil infrastructures such as buildings, viaducts, and tunnels. Although noticeable work has been done to improve anomaly detection for ensuring public safety, algorithms that can be executed on low-cost hardware for long-term monitoring are still an open issue to the community. This paper presents a new framework that exploits compression techniques to identify anomalies in the structure, avoiding continuous streaming of raw data to the cloud. We used a real installation on a bridge in Italy to test the proposed anomaly detection algorithm. We trained three compression models, namely a Principal Component Analysis (PCA), a fully-connected autoencoder, and a convolutional autoencoder. Performance comparison is also provided through an ablation study that analyzes the impact of various parameters. Results demonstrate that the model-based approach, i.e., PCA, can reach a better accuracy whereas data-driven models, i.e., autoencoders, are limited by training set size.**

***Index Terms*—Structural Health Monitoring, Edge computing, Anomaly detection, Deep Learning, Compression Techniques.**

I. Introduction & Related Works

Large-scale civil engineering buildings, such as bridges, dams, and tunnels, have complex structures that can suffer from stress and aging-induced deterioration. In these scenarios, Structural Health Monitoring (SHM) is a key technology [1], because its main objective is to track with continuous monitoring the condition of the infrastructure and to detect early-anomalies, triggering alerts in case of any mismatches with the planned safe state of the structure [2], [3], [4].

The effectiveness of SHM further increases in the case of real-time and online detection, where it can provide an immediate state of structure to avoid catastrophic collapses [1], [2], [4]. As authors in [5], [6] specified, SHM main tasks are the identification, localization, and characterization of damage or deterioration. This can be done by exploiting several types of sensors, such as accelerometers [7], humidity sensors, ultrasonic wave reflection sensors [8], and cameras [9], that have heterogeneous characteristics in terms of power consumption and data generation.

In SHM, anomaly detection is not a trivial task because it faces two major challenges; namely, i) the variety of anomalous behavior, and ii) the lack of large-scale labeled dataset from real (or realistic) anomalies to train the models [10]. Effective anomaly detection in SHM is still a debating field among researchers [11]. According to [12], machine learning-based techniques to detect anomalies can be categorized into four sections: i) clustering-based approaches, ii) classification-based approaches, iii) dimension-reduction-based approaches, and eventually, iv) hybrid approaches that combine multiple technologies. Authors in [13], [14], [6], [15], [16] highlighted the need for techniques in SHM that uses low memory size and can be ported directly on the sensors or the gateways. For instance, [6] uses reconstruction error measured by PCA method to identify anomalies for structural damage detection on an oil platform. It concludes that PCA can be an affordable approach to eliminate the influence of varying wave condition, and provides a technique on damage indices to improve the accuracy of detection. [17] uses fiber optic sensors to record longitudinal displacements over a bridge in Mexico. It exploits PCA to find Q-statistics of collected data to detect damaged features of the bridge, and concludes the work by constructing a threshold value for anomalies detection.

PCA is used for correlated data and the residual deep neural networks for more non-linear correlations data. This is further experimented by [15], where linear PCA or MSD-based approaches do not suit long-term monitoring purposes with highly non-linear patterns. [13] demonstrated that the reconstruction error is a good indicator to find outliers and detect anomalies. Both PCA and autoencoders are trained to construct an outliner detector for heavy-ion collisions. [3] performed unsupervised learning with the purpose of vehicle identification using vibration data from MEMS.

Apart from the aforementioned work on the vibrations acquired by MEMS, other works [18], [15], [19] focus more on the thermal and humidity data to identify damages on the given bridges. Therefore, a study of lightweight unsupervised anomaly detection techniques for acceleration data acquired in real bridges is necessary to fill the gap.

The research is funded by the Sacertis Srl, and partially supported by the EU H2020-ECSEL project AI4DI (g.a. 826060) and by the EU H2020-ECSEL project Arrowhead Tools (g.a. 826452).

Final verison accessible at *A. Moallemi, A. Burrello, D. Brunelli and L. Benini, "Model-based vs. Data-driven Approaches for Anomaly Detection in Structural Health Monitoring: a Case Study," 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2021*

DOI: 10.1109/I2MTC50364.2021.9459999.

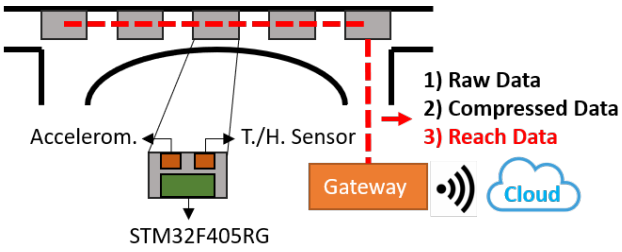


Fig. 1. Overview of the SHM installation.

In this work, we present the following contributions:

- We illustrate the acquisition scenario, and we detail the possibility of embedding an anomaly detection algorithm on the sensors to modulate the amount of data to transmit from remote sensing;
- We use a real bridge scenario, with data samples acquired both from the bridge in normal conditions and from the bridge suffering from structural damage;
- We illustrate a full pipeline, from raw data to the anomaly prediction; with this pipeline, we show that on a 25-days dataset consisting of five days taken with a damaged bridge, and twenty days data from a normal bridge condition, we can achieve up to 98.20% accuracy, with 96.60% sensitivity and 100% specificity;
- We compare model-based and data-driven models as main elements of this pipeline. Further, we explore different pipeline choices, namely i) raw data vs. frequency trace as input, ii) input time dimension, iii) the presence of an additional energy check, and iv) the final post-processing dimension. Finally, we assess the capability of our pipeline to predict more/less severe anomalies using PCA.

The rest of the article is organized as follows: Sec. II introduces the bridge structure, the algorithms of anomaly detection and compression used. Sec. III presents the complete pipeline proposed in this work, detailing the pre-processing (Sec. III-A) and the anomaly detection algorithms, both data-driven and model-driven (Sec. III-B). In Sec. IV, we present general results of our pipeline. Finally, we discuss all the pipeline choices and the difficulties of the task in an ablation study in Sec. V. Sec. VI concludes the paper with final considerations and remarks.

II. Background & Real Scenario

A. Bridge structure

The structure used for the comparison in this article is an old bridge in northern Italy, on the state highway SS335. The SHM system measures accelerations and inclinations in several critical points of the structure, e.g [3]. The bridge consists of 18 sections, but only one is monitored by five nodes, as shown in Fig. 1. Recorded data are split into two groups that contain information before and after a renewing intervention of the bridge to consolidate the structure the building.

B. SHM Framework

As it is shown in Fig. 1, each SHM node uses a triaxial accelerometer, a temperature, and a humidity sensor connected to the STM32F405RG microcontroller as a computational unit. The sampling frequency of the accelerometer is 25.6 KHz, followed by a filtering and subsampling procedure that provides output at 100 Hz. As highlighted in Fig. 1, sensors are connected via a CAN-Bus connection to transfer data to the gateway. Finally, data are streamed to an IoT cloud platform for further analysis.

C. Anomaly Detection: PCA & Autoencoder

Principal Component Analysis is a method to deal with high dimensional correlated data by transforming them into a minimal correlated data [20]. Exploiting co-variance matrix of high dimensional space, lower dimension subspace can be found by taking eigen-values and eigen-vectors of the original one. PCA projects data into low dimensional subspace by preserving most of the energy of the high dimensional space [21] . Consider a $M \times N$ dimensional data matrix $x = [x_1, \quad x_2, \quad x_3, \quad \dots \quad , x_N]$ where x_k is a column vector of M features representing a sample. Its co-variance matrix centered mean is:

$$\Sigma = \frac{1}{n-1} \sum_{k=0}^{N} (x_k - \tilde{x})(x_k - \tilde{x})^T \tag{1}$$

where Σ is a square $M \times M$ matrix. Its diagonal holds variance of each individual sample, and off-diagonal values are co-variances of sample combinations. Using eigen-values decomposition, we can write:

$$\Sigma = V \Lambda V^{-1} \tag{2}$$

V columns are the eigen-vectors, whereas Λ diagonal contains corresponding eigen-values of the co-variance matrix. It can be proven that $\mathbf{V}^k \in \mathbb{R}^{M \times k}$ is a basis of the sub-space of dimension $\mathbb{R}^{M \times k}$ which retains the highest similarity with the original one.

Autoencoders are exploited in several applications for dimensionality reduction, and/or anomaly detection/prediction [22], [23]. These architectures can be a good candidate to be used in SHM systems to develop an accurate anomaly detector. Autoencoders are a sub-group of neural networks, which encompass a two-level structure: the first part, the Encoder, reduce the dimension of the input data $\mathbb{R}^M$, projecting it through one or multiple layer in a latent space $\mathbb{R}^N$, $N < M$. The second part, the Decoder, projects back the signal to its original space. The objective function is the reconstruction of the original signal, to create a latent space which is suitable for the compression of the training data. Compressing and decompressing a signal different from the training ones causes an higher error in the reconstruction. This error is used as a metric for anomaly detection.

Encoder maps the input layer $\bar{x} \in \mathbb{R}^M$ into the latent space $\bar{h} \in \mathbb{R}^k$ through:

$$\bar{h} = f(\bar{x}) = \Phi(W\bar{x} + b) \tag{3}$$

where W is the weight matrix and Φ the non-linearity. Mean square error (MSE) is usually employed as loss [24].

III. Anomaly Detection Pipeline

This section highlights the main contribution of this work, namely a framework to detect anomalies in a SHM system and consequently reducing data traffic to cloud. First, we divide the data into windows and we extract their energy. Windows that do not reach a sufficient energy level are discarded prior to further analysis. After, the windows of vibration data are processed by an anomaly detection algorithm, which includes signal compression

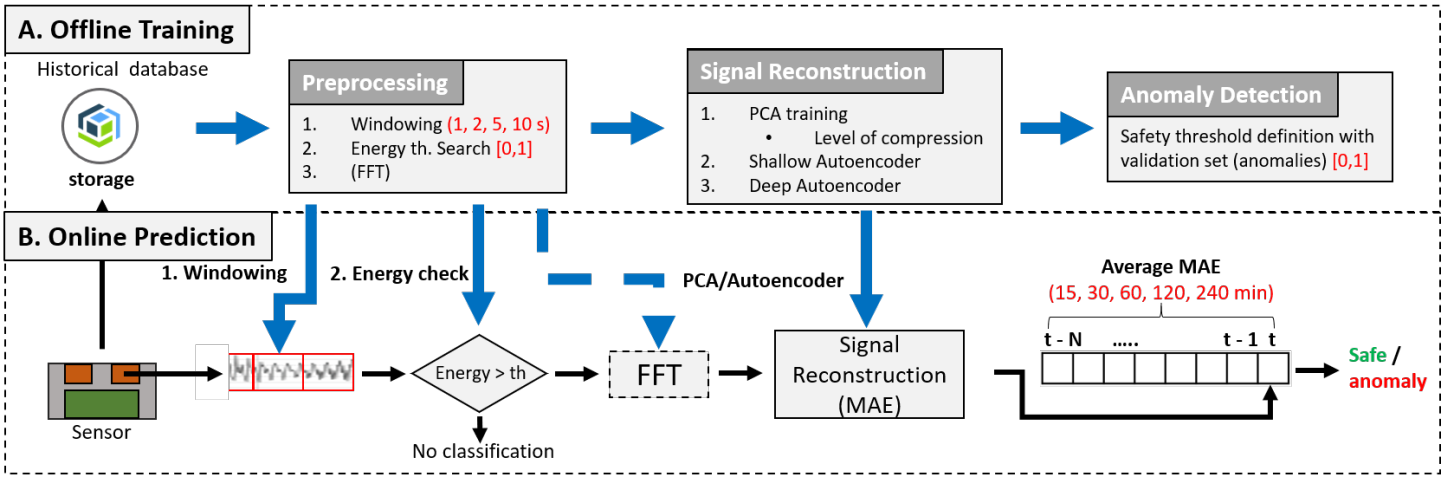


Fig. 2. Description of the anomaly detection pipeline, from acceleration signal to final outcome.

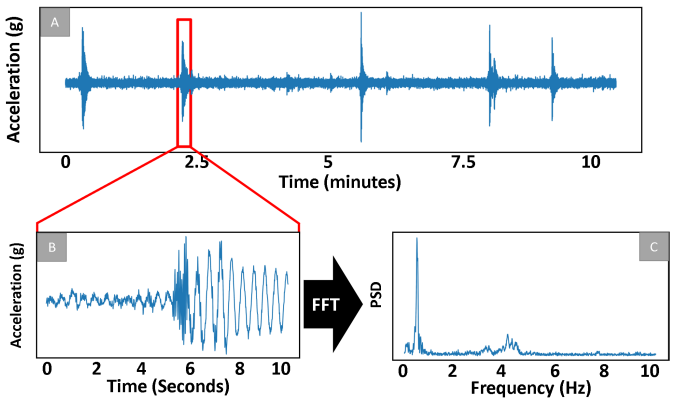


Fig. 3. Window of 10 minutes of input acceleration. In subfigure B and C, the zoom of a peak is represented in time and frequency domain.

and reconstruction to detect anomalies. Fig. 2 illustrates the details of our framework, with an additional offline training vs. online prediction separation.

A. Pre-processing

The first step of the processing pipeline includes the windowing of the signal, the extraction of the energy of the windows, and the frequency extraction. Fig. 3 shows the raw acceleration signal and the zoom on a vibration peak in both time and frequency domains. FFT of the data is used in experiments to compare with anomaly detection made on time domain data. Data are mean centered and scaled down between -1 and 1. The peaks are viaduct vibrations inducted by vehicle passages.

1) Windowing: We divide the data into not overlapping windows for processing, similarly to [6]. We explore a window dimension in the range of 1-10 seconds in an ablation study, demonstrating best performance with 5 seconds dimension. Therefore, during in-field execution of the algorithm, data are gathered for 5 seconds before a new instance of the pipeline is executed to distinguish normal windows from anomalies.

2) Energy Computing & Tresholding: Since the analysed bridge experiences a low level of traffic, hence having many windows without vibrations, we decided to filter out non-informative

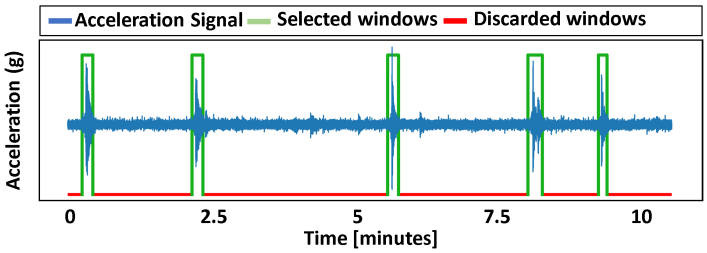


Fig. 4. Input signal filtered with energy. Windows that are above the energy threshold are further processed by the rest of the pipeline.

Algorithm 1 Energy Filtering

Input: $\mathbf{X_{train}}, \mathbf{X_{val}}$
$th = 10^{-10}$
do
  $th+ = 2^{-8}$
  $\mathbf{X_{train}}, \mathbf{X_{val}} \leftarrow$ filter $(\mathbf{X_{train}}, \mathbf{X_{val}}, th)$
  $W \leftarrow pca(\mathbf{X_{train}})$
  $X_r \leftarrow \mathbf{X_{val}} \mathbf{W} \mathbf{W}^\intercal$
  $S \leftarrow RSNR(\mathbf{X}, \mathbf{X_r})$
while $S < 16$ dB
Output: th

windows (white noise) to not impair the anomaly detection. Note that anomaly detection algorithms based on signal compression and reconstruction can not be applied to a signal without correlation, e.g. white noise. Therefore, we extract the energy of each window and remove the ones with energy lower than a trained threshold. In Sec. V-B, we describe the performance gain of this step in the whole application.

We compute energy of each window as follows:

$$E = \sum_{i=1}^{W_d} S_i \tag{4}$$

To determine the energy threshold, we decide to remove an increasing percentage of window, until the reconstruction of the signal does not reach a sufficient mean quality. According to [6],

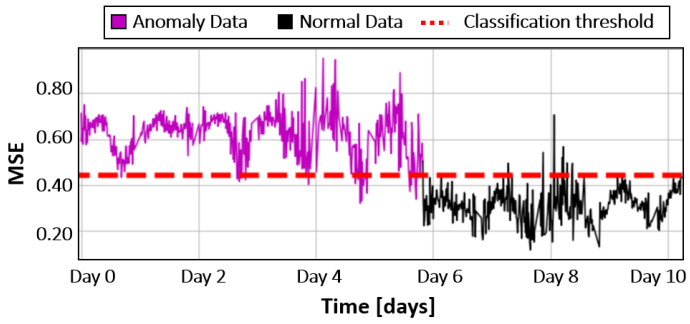


Fig. 5. Classification of the PCA algorithm with 5 input seconds windows and 30 minutes postprocessing. The threshold is automatically defined as described in section III-B

a Reconstructed Signal to Noise Ratio (RSNR) = 16 dB (we refer to [6] for RSNR definition) allows subsequent structure analysis, thus implying that the compressed signal contains enough information of the starting one. Therefore, we increase the percentage of windows removal until the mean RSNR of compression-decompression pipeline over a validation set is higher than 16 dB (using a compression ratio factor of $15\times$ as in [6], [3]). Refer to Alg. 1 for the pseudocode of this window filtering. Fig. 4 shows selected and discarded windows over a 10 minutes interval.

B. Signal Reconstruction & Anomaly Detection

After preprocessing, we feed the data to a compression-decompression algorithm, used in our pipeline as an anomaly detector. Beside the normal task of compression, these algorithms can indeed be used to detect anomaly by looking at the difference between original signal and compressed-reconstructed one. Higher is the difference, farther is the new sample from the training (normal) data.

This step is divided in 3 phases, i.e., the compression, the reconstruction and the computation of the error to identify anomalies. For the first 2 steps, we explored three different algorithms, namely a PCA, with level of compression of $15\times$, a fully-connected autoencoder, with a single hidden layer and without non-linear activation, which mimics the behaviour of the PCA and a convolutional autoencoder with 8 hidden layers followed by sigmoid activation. We used the Adam optimizer and 80 epochs to train our models. Note that PCA and the fully-connected autoencoder are identical from the point of view of the computation, but while the PCA is based on the data-modeling, the second is a data driven approach. With all the algorithm we produce the mean square error (MSE) between original and reconstructed signals.

Finally, a threshold is applied to distinguish normal from anomaly data. As the threshold of the energy, also this threshold is automatically derived from normal (not anomaly) validation data, simulating a scenario with not available abnormal data. To compute it, the compression algorithm is applied to the normal validation set and the threshold is computed as the mean of the MSE over all the data in the validation plus three time their standard deviation. In this way we have a statistical false negative ratio of 0.01%, which is later further reduced with the post-processing. Fig.5 depicts the computation of MSE over normal

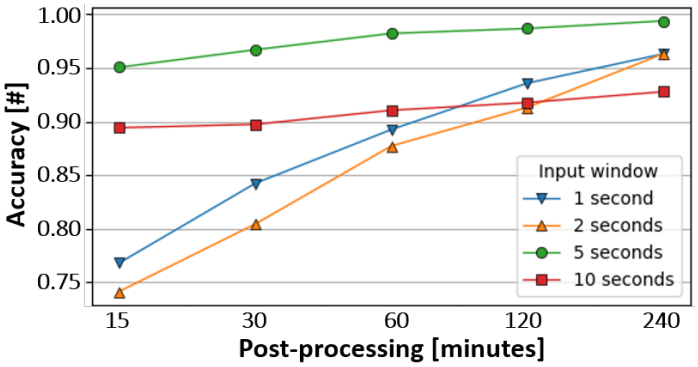


Fig. 6. Performance exploration with input window and post-processing dimension sweeping.

and abnormal data using PCA and the corresponding separating threshold, reaching an accuracy of 0.98 percent.

C. Post-processing: Time Average

To reduce isolated false alarms, which would be not possible in the SHM scenario (the structure degradates over time, and a damage can not be suddenly recovered), we try different averaging post-processing. As presented in Fig. 2, before comparing with the threshold and classify a window as safe or anomaly the MAE is averaged over windows of 15 to 240 minutes. In the next section, we will explore the impact of this step on the overall results.

IV. Experimental Results

A. Setup

This section assesses the performance of the proposed pipeline on a dataset extracted from a real SHM system installed on a viaduct presented in Sec. II-A. We use sensitivity, the percentage of correctly detected anomalies, specificity, the percentage of correctly normal classification and accuracy, the total correct classification, as metrics to validate our pipeline.

B. SHM Dataset

The dataset is composed by five days of anomaly data and twenty days of normal data, continuously sampled at 100 Hz from 5 tri-axial accelerometers. For this analysis, we consider only the the z-axis (i.e., the vertical axis) of the middle sensor of the chain. We consider "anomalies" the data coming from the bridge before a scheduled renewing intervention, made on the bridge that was suffering from structural issues. Given the unique case of this viaduct, to the best of our knowledge this is the first labeled dataset with real anomalies from a viaduct. We divide the data in four sets: i) five days anomaly data test set, ii) five days normal data test set, iii) ten days normal data for training, and iv) five days normal data for validation.

C. Input & Post-processing Dimension Exploration

Beside the two thresholds described in the previous section, Fig.2 shows other 2 hyperparameters (in red) that can be explored, namely the input and the post-processing window dimension. Fig. 6 showcases the space exploration, using four values for input window and an increasing dimension for post-processing. The input size affects the accuracy in a non-deterministic way; hence

TABLE I

COMPARISON OF THE THREE ALGORITHMS OF ANOMALY DETECTION PROPOSED IN OUR WORK USING 5 SECONDS INPUT AND 60 MINUTES FOR POST-PROCESSING.

Algorithm	Data Input	Accuracy	Spec.	Sens.
PCA	Raw	98.2 %	100 %	96.0 %
fully-connected Autoencoder	Raw	91.5 %	94.4 %	89.1 %
convolutional Autoencoder	Raw	50.6 %	67.2 %	37.1 %

TABLE II

PERFORMANCE OF OUR PIPELINE CHANGING ANOMALY DETECTION ALGORITHM WITH BOTH FREQUENCY AND TIME DATA AS INPUT.

Algorithm	Data Input	Accuracy	Spec.	Sens.
PCA	Raw	98.2 %	100 %	96.0%
	FFT	93.2 %	84.9 %	100 %
fully-connected Autoencoder	Raw	91.5 %	94.4 %	89.1 %
	FFT	82.7 %	60.7 %	99.4 %
convolutional Autoencoder	Raw	50.6 %	67.2 %	37.1 %
	FFT	43.5 %	38.2 %	47.6%

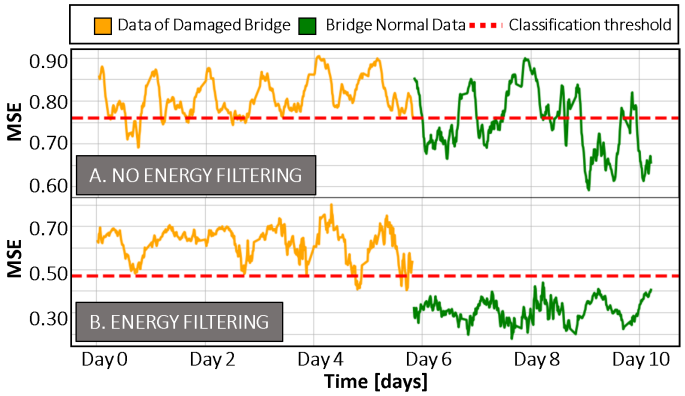
we keep 5 seconds in the subsequent analysis since it outperforms both smaller and bigger windows. On the other hand, using longer post-processing windows is positively correlated with performance. Hence, using longer windows is beneficial for the accuracy, creating a trade-off in the delay vs accuracy space. Here, we do not deeply discuss this trade-off since it is not relevant for our application, usually characterized by slow modifications of the viaduct structure. Therefore, for the rest of the work, we decided to use 60 minutes, which could be a reasonable number also to detect sudden damages and which perform almost perfectly for our use-case. Note that the same analysis holds also for higher post-processing window dimensions.

D. Model-Based vs. Data-Driven Anomaly Detectors

We compare the described models for anomaly detection, namely the PCA, and the two autoencoders, using filtered windows of 5 seconds raw data as input and 60 minutes of post-processing. Results are shown in Table I, with PCA showing the best performance, 98.2%, 100% and 96% of accuracy, specificity and sensitivity, respectively. Fully-connected autoencoder reaches similar performance (91.5% accuracy), since it uses an identical model, with different training procedure. Note that autoencoder performs slightly worse given the low amount of data in training dataset. On the other hand, convolutional autoencoder results in a very low accuracy, given the extremely low capability of recognizing anomaly events. We conclude that this result is again due to the small number of data used to train it, since the deep autoencoder has more parameters and it overfits the training data.

V. ABLATION STUDY

In the following, we examine three scenarios. First, we analyze how different segments of our pipeline affect the performance, showing the impact of use time vs. frequency data as input to our pipeline and the effect of energy filtering. After, we test the tolerance of our detector artificially changing the ”difficulty” of anomalies in the dataset.



	Accuracy	Sensitivity	Specificity
A	76.20%	89.20%	58.90%
B	98.20%	96.60%	100%

Fig. 7. Performance improvement given by the energy filtering step. For this experiment, we use input dimension of 5 s, PCA as classification algorithm and 60 minutes as post-processing average.

A. Raw vs. Frequency Data

Modal analysis and in particular natural frequencies are a key factor to get insights of the dynamic characteristics of a structure [25]. Hence, Fourier analysis can be a promising direction to detect anomalies in a viaduct. Further, we observe a small variation in the power spectrum in most of the windows between normal and anomaly data, which strengthen the conclusion that frequency analysis could give interesting results. Therefore, we test our three anomaly detectors using both frequency and time inputs. For frequency, we extracted the FFT for each input window instead of using the raw signal. Given the low natural frequencies of the viaduct, we cut frequency spectrum in the 0-25 Hz band. As expected, also frequency inputs allow to reach very high accuracy (e.g. 93.2 % accuracy with PCA), but it comes out to be lower when compared to the same analysis performed on raw data. With this analysis, we conclude that we do not have only a phase shift in the natural frequencies of the viaduct in the anomalies, but also magnitude and time patterns of the signal can impact the performance, by giving information about the structural issues of the viaduct.

B. Energy Filtering Impact

In Sec. III-A, we described the filtering of windows with no-information, to avoid a detrimental effect on the anomaly detection algorithm. In Fig. 7, we quantify this benefit, showing the difference in classification between the pipeline with the energy filtering or without it. We can observe that the accuracy of the model, the PCA, is strongly affected by the removal of this part, with a loss of 41.1 % of the specificity. We can notice that this degradation is given by the higher MSE showed also in the normal windows (0.5 vs. 0.8 mean MAE). This behaviour is mainly caused by the windows that are filtered out by energy (”not informative” windows), since reconstructing windows without vibration is an impossible task and result in an high MAE (there is no correlation in white noise, and there is not a ”trainable”

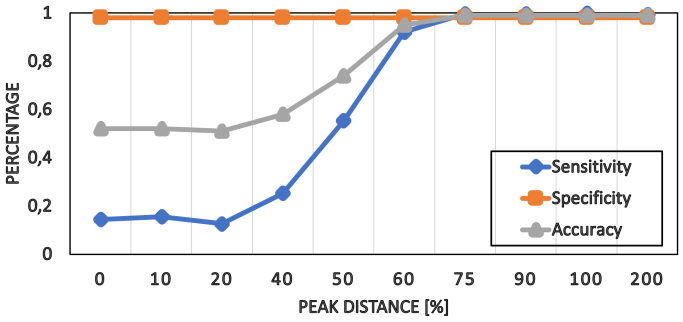


Fig. 8. Specificity on differenet distances w.r.t real anomaly case scenario of bridge

scheme). These windows are present both in anomalies and in normal windows. Therefore, being these windows the dominant ones, the difference between the "informative" ones is reduced (the MAE is the balanced mean between informative and not informative windows MAE over time) and the two classes are not more linear separable.

C. Synthetic Experiments

We finally evaluate the difficulty of the task, trying to artificially vary the difficulties of the anomalies. To realize different sets of anomalies, we decided to vary the distance between frequency peaks in normal and anomaly data, transforming the data to the frequency domain and then back to the time. In particular, we use a 15 minutes FFT to increase the resolution in frequency and we change the distance between the peaks (we have a single peak in the spectrum) of the two classes between 0 to 200% of the original distance. Closer peaks should strongly increase the difficulty of the problem. A similar approach has been used in other fields to generate synthetic anomalies [26], [27]. The results of these experiments are reported in Fig. 8, where 100 % points to unchanged data. From the graph, we can clearly see that while the specificity is constant, the sensitivity of the algorithm proportionally increases with the increment of the peak distance. With 75% of the original distance, our algorithm already perform similarly to its application on original data, achieving 99% of accuracy, whereas specificity is constant at 98%, sensitivity increased to be approximately 100%.

VI. Conclusions

Light and strong stresses can cause structures to be non-functional or dangerous to be used. Thus, continuous monitoring of such a structure with the aim of degradations over time monitoring is the main objective of many SHM systems. This work presents several approaches to improve SHM bridge installations with ML architectures to detect anomalies. We showed a real system enhanced by our proposed pipeline to monitor degradation on a unique use case of an Italian bridge. Both model-based and data-driven approaches have been explored, achieving nearly optimal accuracy with model-based perspective (i.e., PCA). Conversely, more complex data-driven algorithms suffer from the scarce amount of data.

References

[1] L. D'Errico *et al.*, "Structural health monitoring and earthquake early warning on 5g urllc network," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 783–786.

[2] H.-F. Chang *et al.*, "Real-time structural health monitoring system using internet of things and cloud computing," 2019.

[3] A. Burrello *et al.*, "Enhancing structural health monitoring with vehicle identification and tracking," in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2020, pp. 1–6.

[4] A. Abdelgawad *et al.*, "Structural health monitoring: Internet of things application," in *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2016, pp. 1–4.

[5] F. Lamonaca *et al.*, "Internet of things for structural health monitoring," in *2018 Workshop on Metrology for Industry 4.0 and IoT*, 2018, pp. 95–100.

[6] A. Burrello *et al.*, "Embedded streaming principal components analysis for network load reduction in structural health monitoring," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[7] A. Girolami *et al.*, "Low-cost and distributed health monitoring system for critical buildings," in *2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, 2017, pp. 1–6.

[8] D. Dai *et al.*, "Structure damage localization with ultrasonic guided waves based on a time–frequency method," *Signal Processing*, vol. 96, pp. 21 – 28, 2014, time-frequency methods for condition based maintenance and modal analysis.

[9] A. Basharat *et al.*, "A framework for intelligent sensor network with video camera for structural health monitoring of bridges," in *Third IEEE International Conference on Pervasive Computing and Communications Workshops*, 2005, pp. 385–389.

[10] G. Pang *et al.*, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 353–362.

[11] J. Chen *et al.*, *Outlier Detection with Autoencoder Ensembles*. SIAM, 2017, pp. 90–98. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611974973.11

[12] J. Jiang *et al.*, "Outlier detection approaches based on machine learning in the internet-of-things," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 53–59, 2020.

[13] P. Thaprasop *et al.*, "Unsupervised outlier detection in heavy-ion collisions," 2020.

[14] F. Siddiqui *et al.*, "The use of pca and signal processing techniques for processing time-based construction settlement data of road embankments," *Advanced Engineering Informatics*, vol. 46, p. 101181, 2020.

[15] E. Figueiredo *et al.*, "Linear approaches to modeling nonlinearities in long-term monitoring of bridges," *Journal of Civil Structural Health Monitoring*, vol. 3, no. 3, pp. 187–194, 2013.

[16] F. Zonzini *et al.*, "Cluster-based vibration analysis of structures with gsp," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3465–3474, 2021.

[17] D. Garcia-Sanchez *et al.*, "Bearing assessment tool for longitudinal bridge performance," *Journal of Civil Structural Health Monitoring*, vol. 10, 2020.

[18] A.-M. Yan *et al.*, "Structural damage diagnosis under varying environmental conditions—part ii: local pca for non-linear cases," *Mechanical Systems and Signal Processing*, vol. 19, no. 4, pp. 865 – 880, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327004001797

[19] E. Cross *et al.*, "Long-term monitoring and data analysis of the tamar bridge," *Mechanical Systems and Signal Processing*, vol. 35, no. 1, pp. 16 – 34, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327012003573

[20] Z. Cui *et al.*, "Bat algorithm with principal component analysis," *International Journal of Machine Learning and Cybernetics*, vol. 10, p. 603–622, 2019.

[21] X. Zhao *et al.*, "Enhanced particle swarm optimization based on principal component analysis and line search," *Applied Mathematics and Computation*, vol. 229, pp. 440 – 456, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300313013519

[22] M. Kan *et al.*, "Stacked progressive auto-encoders (spae) for face recognition across poses," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[23] C. S. N. Pathirage *et al.*, "Stacked face de-noising auto encoders for expression-robust face recognition," in *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2015, pp. 1–8.

[24] C. S. N. Pathirage *et al.*, "Structural damage identification based on autoencoder neural networks and deep learning," *Engineering Structures*, vol. 172, pp. 13 – 28, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0141029618302062

[25] J. Su *et al.*, "Review on field monitoring of high-rise structures," *Structural Control and Health Monitoring*, vol. n/a, no. n/a,

p. e2629, 2020, e2629 STC-19-0500.R2. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2629

[26] C. Wang *et al.*, "Detection of false data injection attacks using the autoencoder approach," 2020.

[27] X. Niu *et al.*, "Dynamic detection of false data injection attack in smart grid using deep learning," in *2019 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2019, pp. 1–6.