Vibration Monitoring in the Compressed Domain with Energy-Efficient Sensor Networks

Published Version:
Ragusa, E., Zonzini, F., De Marchi, L., Gastaldo, P. (2023). Vibration Monitoring in the Compressed Domain with Energy-Efficient Sensor Networks. IEEE SENSORS LETTERS, 7(8), 1-4 [10.1109/LSENS.2023.3300804].

Availability:
This version is available at: https://hdl.handle.net/11585/937834 since: 2023-10-23

Published:
DOI: http://doi.org/10.1109/LSENS.2023.3300804

(Article begins on next page)

Data compression, Feature Extraction, Machine Learning, Structural Health Monitoring

Vibration Monitoring in the Compressed Domain with Energy-Efficient Sensor Networks

Edoardo Ragusa[1*§], Federica Zonzini[2*§], Luca De Marchi[2*], and Paolo Gastaldo[1]

[1] Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture (DITEN), University of Genoa, Genoa, Italy
[2] Department of Electrical, Electronic, and Information Engineering (DEI), University of Bologna, Bologna, Italy
[*] Member, IEEE

Manuscript received ; revised ; accepted . Date of publication ; date of current version .

Abstract—Structural Health Monitoring (SHM) is crucial for the development of safe infrastructures. Onboard vibration diagnostics implemented by means of smart embedded sensors is a suitable approach to achieve accurate prediction supported by low-cost systems. Networks of sensors can be installed in isolated infrastructures allowing periodic monitoring even in the absence of stable power sources and connections. To fulfill this goal, the present paper proposes an effective solution based on intelligent extreme edge nodes that can sense and compress vibration data onboard, and extract from it a reduced set of statistical descriptors that serve as input features for a machine learning classifier, hosted by a central aggregating unit. Accordingly, only a small batch of meaningful scalars needs to be outsourced in place of long time series, hence paving the way to a considerable decrement in terms of transmission time and energy expenditure. The proposed approach has been validated using a real-world SHM dataset for the task of damage identification from vibration signals. Results demonstrate that the proposed sensing scheme combining data compression and feature estimation at the sensor level can attain classification scores always above 94%, with a sensor life cycle extension up to 350x and 1510x if compared with compression-only and processing-free implementations, respectively.

Index Terms—Compressed Sensing, On-sensor Feature Extraction, Vibration Monitoring

I. INTRODUCTION

Structural Health Monitoring (SHM) systems supported by embedded smart devices allow for the automatic inspection of technical facilities, providing a suitable solution to control their health status with minimal cost and invasiveness. This result can be reached by designing a novel generation of intelligent sensor systems, equipped with data processing and mining capabilities [1]. On the one hand, the literature confirms that combining information from all the sensing nodes is crucial to obtain accurate structural predictions [2]. On the other, sharing information within the network requires a large amount of energy and is the primary source of power consumption in wireless sensor networks built on the sensor-to-cloud continuum [3]. Therefore, the current challenge in sensor network design is to maximize the accuracy of the structural health assessment process while minimizing the amount of information transmitted inside the monitoring network as an indirect means to optimize the overall sensor power budget.

To address this goal, this paper presents a solution based on smart sensing nodes, placed on different areas of the facility under analysis. Figure 1 schematizes the proposed sensing system. Extreme edge sensors collect and compress data; then, they extract a set of statistical descriptors which are forwarded to a low-end centralizing node featuring resource-constrained computing units. The latter is in charge of data aggregation and damage identification directly from the reduced set of compressed features by resorting to a low-complexity predictor built on a Machine Learning (ML) classifier. We investigate this problem in the specific context of vibration-based diagnostics, i.e., scenarios in which the structural conditions are typically judged by analysing vibration (e.g., acceleration) signals induced by the dynamic response of the target asset. As a major advantage, communication inside the network of sensors can be handled using extremely low-power protocols (e.g., LoRaWan), that fail to support cloud-based alternatives involving a continuous and burdensome flow of data.

Three main contributions are fulfilled: 1) we offer one of the very first attempts for structural inference directly in the compressed domain, overcoming the need to reconstruct the original time series; 2) the combination of on-sensor data compression and feature extraction leads to a sensor network with minimal data transfer among nodes (up to 2500x lower payload); 3) we perform a cost-energy analysis demonstrating that sensor-near analytics introduces a negligible data pre-processing overhead with respect to full data transfer solutions; in turn, we prove that this can extend the energy autonomy of battery-powered sensors at a large extent.

II. BASICS

Compressed Sensing (CS) approaches are data reduction strategies capable to encode long time series into a preset number of coefficients, under the hypothesis that the class of processed signals is *sparse* in a specific representation domain [4]. Such condition perfectly applies to vibration signals, since they can be completely described by a small batch of peak spectral features (also known as modal frequencies), once transformed in the Fourier domain [5].

One of the main advantages of CS lies in its low-cost implementation via simple multiply and accumulate operations [6]. Assuming that a sensing node S_i records an N-long vibration signal $x_i \in R^{N\times 1}$, the latter can be transformed into a reduced vector $\hat{x}_i \in R^{M\times 1}$ ($M << N$) according with:

Corresponding author: Federica Zonzini (federica.zonzini@unibo.it).

[§] These authors equally contributed to this work.

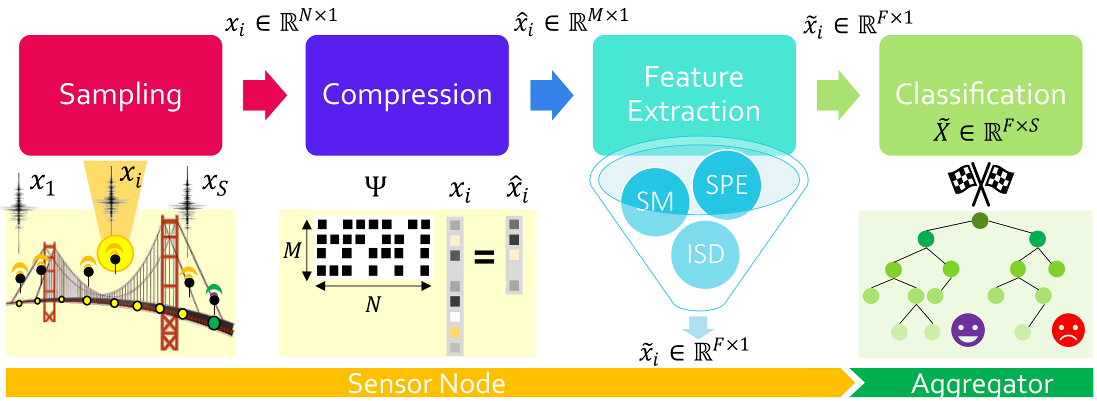

Fig. 1: Schematic description of the processing flow described

$$\hat{x}_i = \Psi x_i \tag{1}$$

$\Psi \in R^{M\times N}$ is the so-called compression (or sensing) matrix and has to be appropriately selected to maximize the information content which is retained at the reduction stage.

After compression, vibration data are still organized as ordered sequences carrying information about the structure's health state. A variety of approaches proves effective in mining information from time series and, more in general, sequences [7]. Deep Learning [8] and dedicated similarity metrics for sequence comparison, like Dynamic time warping (DTW) [9], prove state-of-the-art in modeling dependencies and recurrent patterns. When limited resources are available, recent works demonstrated that statistical features with low-cost computing requirements can equate the accuracy of more demanding procedures while reducing the computing requirements [10], [11]. Such solutions avoid facing the multitude of issues related to the deployment of deep data mining architectures on resource-constrained devices [12], that is still a very active line of research [13], [14]. A similar observation holds for DTW and affine metrics that involve solving an optimization problem in a streaming fashion.

III. DAMAGE DETECTION FROM COMPRESSED FEATURES

The proposed sensing system (as per Fig. 1) resorts to a network of vibration sensors, each of them equipped with an ultra-low-power module that records, compresses, elaborates, and sends feature data. The novelty of our scheme is that, rather than transmitting the entire $\hat{x}$ as it is conveniently done in standard CS-based settings, we further reduce the payload dimension by extracting, directly at the sensor node level, a set of hardware-friendly statistical features. The extreme compression of the data on the sensing node limits, by construction, the cross-sensor information used in the learning process. However, this risk is unavoidable when aiming to minimize the transmission rate. The latter are organized in a vector $\tilde{x} \in R^F$, F being the number of features; accordingly, $F << M << N$ holds, imposing a compression ratio $CR_{CS+Feat} = N/F$ which is M/F times higher than the one attainable by CS-only alternatives, i.e., $CR_{CS} = N/M$.

Among the wide variety of statistical indicators, the following quantities were computed, since they are suited for time series analysis using low-power embedded systems [11]: statistical moments (SM), i.e., median, variance, skewness, and kurtosis; stationary points and energy (SPE), namely maximum, minimum, maximum-minimum distance, and energy; inter-samples differences (ISD), i.e., mean absolute deviation, median absolute deviation, mean of differences between adjacent samples, and mean of absolute differences between adjacent samples.

The reason for extracting statistical features in the compressed domain rather than from raw vibrations is that, together with the cost of transmission, the number of computations performed by the embedded device to extract them must be considered. Obviously, the latter is proportional to the number of samples in the time series. Amidst SPE, ISD and SM, the computation of skewness requires the largest number of operations and it is proportional to $10l$ (l being the generic dimension of the vector to be processed), while the computational complexity reduces to $4l$ for median, and $2l$ for the remaining and less demanding features [11]. The number of total floating point operations (FLOPS) necessary for the extraction of the whole feature set grows linearly with the length of the input sequence, making the approach suitable for constrained devices.

The aggregator unit hosts the instance of a standard ML classifier and performs structural inference upon aggregation of S different feature sets $\tilde{x}_i$, with S equal to the number of sensors. In this work, a Random Forest (RF) is selected because it offers excellent generalization performance with modest computing requirements [15]. In fact, it consists of an ensemble of classification trees and performs prediction by means of simple comparisons with preset threshold values. The literature proposes a plethora of efficient implementations, including highly optimized C libraries suitable for microprocessor-based computing units. Importantly, it is worth highlighting that the choice of the RF as a predictor is non-binding, and it can be substituted, in principle, with any other detector capable to infer data from the extracted feature set. However, RF proves robust to outliers and missing input values thanks to the regularization properties of ensemble learning. This becomes useful when considering that the extracted feature set has not been designed to handle input nodes' malfunctions.

IV. EXPERIMENTS

A. Materials and methods

1) Dataset: The Z24 bridge dataset represents a benchmark use case for vibration-based SHM [16]. The entire data collection consists of 5651 time series acquired by eight force-balance-type FBA-11 uni-axial accelerometers installed at different locations on the bridge deck and pillars: 4922 instances were taken in healthy configurations, the remaining 729 in damaged ones. Each sensor was configured to collect 32768 vibration data per hour at a sampling frequency of 50 Hz, over a measurement period of one year.

2) Processing framework: Among the manifold CS techniques, the Model-assisted Rakeness-based (MRak-CS) strategy in [5] has been chosen, which belongs to the class of adaptive CS mechanisms, i.e., methods exploiting statistical priors about the properties of the processed signals to design the sensing matrix. MRak-CS offers a conservative approach, meaning that it prevents overadaptation to the spectral distributions used for the design of the compression matrix itself. This working principle is crucial to capture structural variations over time; more importantly, it permits to store Ψ statically in the sensor memory, hence avoiding the need to compute it in a streaming way, a choice which would imply a prohibitive expenditure in terms of power consumption and execution time.

Additionally, it is worth recalling that severe memory constraints characterize low-end microprocessors, as those typically equipped by smart low-cost sensors, where the flash space hardly exceeds 1 MBytes (e.g., cut-of-the-shelf STM32 or ESP32 microcontroller units). To fulfill this limitation, each time series has been split into subsequent windows of dimension $N = 512$ samples, while the compression ratio CR_{CS} has been swept from 4 to 256, doubling it at each iteration. By doing so, the maximum memory slot necessary for the allocation of Ψ amounts to 512 kBytes.[1]

Beside, the code for the RF classifier was implemented in Python by considering a standard 5-fold cross-validation: for each fold, a validation set was extracted from the training set selecting a random subset of 20% of the training data.

B. Results

The performance of the RF classifier versus the impact of increasing CR_{CS} has been evaluated first: results are summarized in Table 1 and are provided in terms of standard classification metrics (i.e., accuracy, precision, recall, and F1), following the same validation setup proposed in [17]. As can be observed, all the indicators remain stably above 94% even in the most severe setting, with negligible drop when moving from the lowest to the deepest CR_{CS}. This outcome can be justified by the adopted MRak-CS scheme, which is designed to preserve the statistical properties of the signals after compression. Hence, it perfectly combines with the selected feature set.

Additionally, scores have been compared with three alternative data-driven solutions already proposed in the literature for the same benchmark. Gaussian Mixture Model (GMM) has firstly been introduced in [18] as a non-supervised model for anomaly detection from vibration data, while the One Class Classifier Neural Network (OCCNN) and the Autoassociative Neural Network (ANN) in [17] address the same task by means of small-size ML architectures. The CS+Feature extraction workflow presented in this work differs remarkably from the ones envisioned in the previous attempts. Indeed, in [18], data compression procedures are totally neglected, hence sensing nodes are meant to transfer to a central unit the entire raw time series (32768 time samples). Conversely, in [17], the compression stage has been introduced at the sensor level to maximize the overall

Table 1: Classification performances of the implemented RF compared with existing ML models dealing with the same benchmark: analysis in terms of increasing CR and network payload.

Classifier	CR_{CS}	**Accuracy** [%]	**Precision** [%]	**Recall** [%]	**F1** [%]	**Payload** [# samples]
RF	4	94.00	96.17	96.97	96.57	13
RF	8	94.01	96.27	96.89	96.56	13
RF	16	93.79	96.11	96.79	96.45	13
RF	32	94.07	96.36	96.85	96.60	13
RF	64	94.35	96.41	97.14	96.77	13
RF	128	93.65	96.00	96.75	96.37	13
RF	256	94.04	96.27	96.91	96.59	13
OCCNN [17]	6	93.00	94.00	95.00	91.00	5462
ANN [17]	6	95.00	99.00	94.00	97.00	5462
GMM [18]	1	95.00	98.00	93.00	95.00	32768

energy consumption, but signal decompression is encompassed at the centralizing unit before extracting damage sensitive features. In both cases, the latter coincide with modal frequencies, whose reconstruction and identification passes through computationally expensive structural identification techniques.

On top of that, the last column of Table 1 quantifies the number of floating point elements that the edge sensor should send to the aggregator unit to perform the inference of a single datum. As can be seen, our solution can achieve comparable or even better results with respect to state-of-the-art alternatives (only GMM can obtain a slight improvement in accuracy and precision, whereas OCCNN and ANN score always worst), moreover allowing for a huge limitation of the data dimension: indeed, in our implementation, we only require 13 feature values to be transmitted over the sensor network, leading to a data payload which is 420x and 2500x lower with respect to that involved by [17] and [18], respectively.

V. COST-BENEFIT ANALYSIS

A cost-benefit analysis has been performed to effectively evaluate the superiority of the proposed workflow from an energy budget point of view. To this end, the tool in [19] has been exploited to simulate the power expenditure of a target device when featuring different wireless communication protocols typically employed for the deployment of battery-operated monitoring networks in modern sensor installations [20]: the BLE 5.0 technology, the LoRaWan connectivity, WiFi HaLoW based on the 802.11ah and the IEEE 802.15.4 standard[2].

A sensor life cycle estimation has been performed under the following assumptions: i) data are sampled on a hourly basis, which is a common duty-cycle for this kind of structures where degradation phenomena undergo a slow inertia; ii) power is drawn from a battery with a capacity of 3600 mAh and reference voltage of 3.3 V, iii) each device is equipped with a low-end microprocessor clocked at 80 MHz and featuring 15 mA and 7 μA current consumption in normal and idle operating mode, respectively. Starting from these electrical parameters, the energy spent in one hour has been estimated first as

$$E_{1h} = E_{sensing} + E_{DSP} + E_{idle} \tag{2}$$

namely by summing together the contribution due to data sensing ($E_{sensing}$), on-board data processing (E_{DSP}) and the one consumed in absence of any operation (E_{idle}). While $E_{sensing}$ is a constant

[1] It is assumed that each piece of information is represented in `float32` format, i.e., as a word of 4 Bytes.

[2] A communication distance equal to 50 m has been chosen to perform a realistic study and be compatible with the considered protocols.

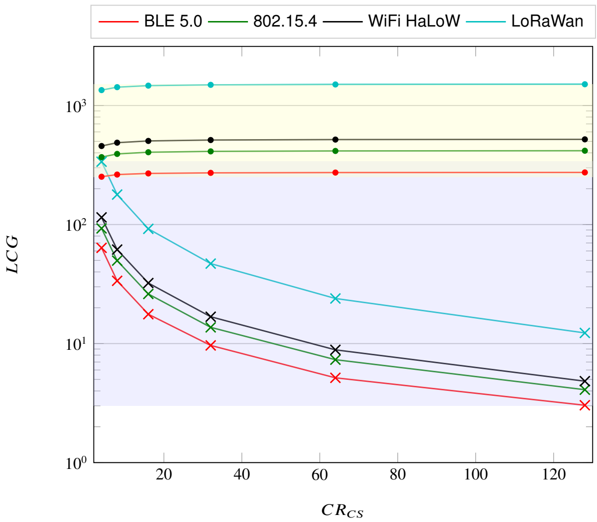

Fig. 2: Gain in the sensor life cycle of the CS+Feature extraction framework proposed in this work with respect to the absence of on-sensor analytics (yellow background and round markers) and purely CS-based setting (blue background and cross markers) for various CR_{CS} and communication protocols.

quantity, E_{DSP} varies depending on the level of sensor-near functionalities charged to the extreme edge sensor. To investigate this point, we have compared the CS+Feature extraction setting proposed in this work with two alternatives: absence of any on-board processing functionality (No_DSP label) and CS-only configurations (label CS). Then, the gain in the sensor battery life cycle has been computed as

$$LCG = \frac{E_{1h}^{\dagger}}{E_{1h}^{CS+Feat}} \tag{3}$$

in which the superscript † indicates either CS or No_DSP realizations.

Trends as a function of increasing CR_{CS} and different communication protocols are reported in Fig. 2: they show that our approach allows to extend exponentially the sensor autonomy with respect to purely CS frameworks (curves in the blue background with cross markers), from a minimum of 3x for BLE 5.0 with CR_{CS} = 128 to a maximum of 337x for LoRaWan at CR_{CS} = 4. Conversely, such advancement is almost constant at values always above 250x (reaching 1510x in case LoRaWan is used) when comparing with processing-free solutions (yellow background and round markers). Such outcomes are a direct consequence the data payload size imposed by the three different settings: while the data payload is fixed to 13 and 32768 values for our implementation and No_DSP, respectively, the amount of data to be transmitted in case of CS increases while diminishing the CR_{CS} level.

VI. CONCLUSION

This work presented a solution for low-power networks of vibration sensors: it exploits CS and feature extraction at the extreme edge sensor and a low-cost ML classifier at the centralizing side to trade-off computational cost and generalization performance of the system. As a major result, a suitable feature extraction procedure cuts down data transfer inside the network, thus leading to power-efficient aggregation of the information from different sensing nodes. Tests on a real-world dataset confirm the suitability of the proposed approach. Comparison with the existing solution highlights the effectiveness of the proposed method.

ACKNOWLEDGMENT

F.Z. wants to thank the project PNRR – M4C2 – Investimento 1.3, Partenariato Esteso PE00000013 – "FAIR – Future Artificial Intelligence Research" – Spoke 8 "Pervasive AI", funded by the European Commission under the NextGeneration EU programme.

REFERENCES

[1] L. Sun, Z. Shang, Y. Xia, S. Bhowmick, and S. Nagarajaiah, "Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection," *Journal of Structural Engineering*, vol. 146, no. 5, p. 04020073, 2020.

[2] C. Sbarufatti, A. Manes, and M. Giglio, "Application of sensor technologies for local and distributed structural health monitoring," *Structural Control and Health Monitoring*, vol. 21, no. 7, pp. 1057–1083, 2014.

[3] Y. M. Kadioglu, "Energy consumption model for data processing and transmission in energy harvesting wireless sensors," in *Computer and Information Sciences: 31st International Symposium, ISCIS 2016, Kraków, Poland, October 27–28, 2016, Proceedings*. Springer, 2016, pp. 117–125.

[4] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[5] F. Zonzini, M. Zauli, M. Mangia, N. Testoni, and L. De Marchi, "Model-assisted compressed sensing for vibration-based structural health monitoring," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7338–7347, 2021.

[6] Z. Zou, Y. Bao, H. Li, B. F. Spencer, and J. Ou, "Embedding compressive sensing-based data loss recovery algorithm into wireless smart sensors for structural health monitoring," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 797–808, 2014.

[7] Z. Han, J. Zhao, H. Leung, K. F. Ma, and W. Wang, "A review of deep learning models for time series prediction," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 7833–7848, 2019.

[8] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[9] H. Li, "Time works well: Dynamic time warping based on time weighting for time series data mining," *Information Sciences*, vol. 547, pp. 592–608, 2021.

[10] E. Ragusa, R. Zunino, V. Mastronardi, M. Moglianetti, P. P. Pompa, and P. Gastaldo, "Design of a quantitative readout in a point-of-care device for cisplatin detection," *IEEE Sensors Letters*, vol. 6, no. 11, pp. 1–4, 2022.

[11] C. Gianoglio, E. Ragusa, P. Gastaldo, and M. Valle, "A novel learning strategy for the trade-off between accuracy and computational cost: a touch modalities classification case study," *IEEE Sensors Journal*, vol. 22, no. 1, pp. 659–670, 2021.

[12] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine learning for microcontroller-class hardware-a review," *IEEE Sensors Journal*, 2022.

[13] H. Shi, H. You, Z. Wang, and Y. Lin, "Nasa +: Neural architecture search and acceleration for multiplication-reduced hybrid networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.

[14] H. Benmeziane, K. E. Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "A comprehensive survey on hardware-aware neural architecture search," *arXiv preprint arXiv:2101.09336*, 2021.

[15] J. Xie and M. Zhu, "Maneuver-based driving behavior classification based on random forest," *IEEE Sensors Letters*, vol. 3, no. 11, pp. 1–4, 2019.

[16] B. Peeters and G. De Roeck, "One-year monitoring of the z24-bridge: environmental effects versus damage events," *Earthquake engineering & structural dynamics*, vol. 30, no. 2, pp. 149–171, 2001.

[17] F. Zonzini, A. Carbone, F. Romano, M. Zauli, and L. De Marchi, "Machine learning meets compressed sensing in vibration-based monitoring," *Sensors*, vol. 22, no. 6, p. 2229, 2022.

[18] E. Favarelli and A. Giorgetti, "Machine learning for automatic processing of modal analysis in damage detection of bridges," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2020.

[19] F. Zonzini, V. Dertimanis, E. Chatzi, and L. De Marchi, "System identification at the extreme edge for network load reduction in vibration-based monitoring," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 467–20 478, 2022.

[20] A. Sabato, C. Niezrecki, and G. Fortino, "Wireless mems-based accelerometer sensor boards for structural vibration monitoring: a review," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 226–235, 2016.