Integrating Data-driven and Model-based Algorithms for Structural Health Monitoring

Roberto Boccagna
Research scholar, Dept. of Engineering and Geology, Univ. "G. D'Annunzio", Pescara, Italy

Maurizio Bottini
PhD student, Dept. of Engineering and Geology, Univ. "G. D'Annunzio", Pescara, Italy

Massimo Petracca
Research Engineer, Asdea Software, Pescara, Italy

Mirko Rossi
Research Engineer, Asdea Hardware, Pescara, Italy

Alessia Amelio
Researcher, Dept. of Engineering and Geology, Univ. "G. D'Annunzio", Pescara, Italy

Guido Camata
Professor, Dept. of Engineering and Geology, Univ. "G. D'Annunzio", Pescara, Italy

ABSTRACT: Artificial Intelligence has rapidly supplanted classical statistical methods for data analysis and prediction-making in many scientific areas in the last few years due to substantial technological advances in hardware that enable the robust dedicated algorithms to produce reliable results in reasonable software execution times. Looking specifically at applications in the field of Structural Health Monitoring, automated algorithms aim to replace visual inspections for structural condition assessments, as they can reduce maintenance costs and identify damage not otherwise detectable. In this paper, we propose a well-structured approach covering the entire monitoring process, overcoming the critical issues that available algorithms suffer from, including the availability of suitable training data and compatibilities among the solutions used for different tasks throughout the entire anomaly detection process. This is possible through the adoption of a standard data storage format and the numerical construction of a digital twin of the structure under inspection. This way, ad hoc baseline patterns may be generated to feed artificial neural networks that are truly supervised, and any alarms created can, therefore, be checked by running dynamic simulations of the corresponding FEM model.

1. INTRODUCTION

In the last few decades, Structural Health Monitoring (SHM) has benefited greatly from the outstanding progress made in the field of outlier detection-oriented algorithmic theory and from the considerable increase in CPU velocity and GPU performance in parallel computing. Such progress paved the way for anomaly detection in the context of Civil Engineering, making available the possibility of fast data collection, transmission and processing methods for a near real-time response of the overall structural state. Over the years, new, automated techniques stood alongside traditional, i.e., in-person, facility surveillance, and gradually supplanted visual inspection and monitoring as soon as networks of synchronized sensors were integrated in "smart structures". This paradigm change resulted in a substantial cost reduction for maintenance and, above all, in increased reliability when assessing structural conditions.

Data-driven methods, i.e., detection techniques entirely based on the analysis of real data, enable the recognition of hidden patterns embedded in the data flow, and report any anomalous streams of information that indicate possible damage that would be otherwise undetectable. This reflects the underlying, foundational assumption of automated monitoring methods: damage, whether cracks, collapses, local mechanisms, or gradual deteriorations, are present as data anomalies that can be recognized. Whenever data analysis detects possible outliers, dedicated algorithms may be employed for damage localization and for evaluating damage type and extent. Then, in the decision-making phase, the remaining lifetime of the structure is assessed, and expert engineers can promptly establish whether or not to restrict access to the facility as a precautionary measure.

Artificial Intelligence (AI) has a prominent role in this context since it provides various solutions to the problems posed by SHM. In particular, Machine Learning (ML) methods are capable of recognizing atypical patterns in the datasets via statistical comparison of near real-time signals and an adequately large set of training data that the algorithms use to infer the processes that generate "healthy" and, if available, "damage" scenarios.

In the SHM arena, model-based algorithms are often adopted on their own or combined with data-driven algorithms for damage detection. Such methods provide for the construction of a Digital Twin (DT) of the examined structure, so that data collected by sensors can be used as forcing terms for Finite Element Method (FEM) dynamics numerical simulations. The more accurate the model in terms of geometrical, environmental and physical properties, the more reliable the predictions made by the algorithm.

Although the idea of combining data-driven and model-based approaches is definitely not new, such hybridization still represents a challenging issue for Civil and Computer Engineering as, without a unified framework, there is nothing to act as a mortar between the disciplines.

In this work, we propose a comprehensive solution for SHM, which has the advantage of being conceived as a chain of tools and algorithms designed and optimized specifically to perform their unique tasks. The Scientific ToolKit for Opensees (STKO) GUI for OpenSees [Petracca et al. (2017)] is employed for the DT modeling and dynamics, as it offers a Python interface for customization, which conveniently connects the model-based and data-driven approaches, of which the data-driven approach is entirely developed in Python. Moreover, a unique data format (the HDF5 format) is adopted for I/O operations to avoid data incompatibilities.

2. THE MONITORING PROCESS

Currently, the main issues that arise in the automated monitoring process regard the establishment of a robust framework that covers all the intermediate steps from data acquisition to output production and interpretation. The Civil Engineering community offers many suggestions that may fit very specific cases [Tibaduiza Burgos et al. (2020)], but an objective standard for SHM is still lacking. In particular, the pre-processing sequence of manipulations that transform rough data time-series into a suitably small set of features to be processed via AI tools represents the most controversial part of the entire monitoring problem. Differences among the various solutions may already emerge during data collection, as different devices register different physical quantities at a given sampling rate.

The information embedded in the time series recorded for a fixed time window is typically extracted as a reduced set of representative features, as the whole time series would be intractable from the AI point of view. Such features can be divided into global, if extracted through multivariate analysis by considering data of a given time window as a whole, with the typical aim of estimating modal characteristics, or local, if data are unpacked before calculating temporal and/or spectral indicators for each time series. Further pre-processing steps, including data standardization and projection onto a subspace of lower dimension, may eventually be performed to eliminate redundant information, taking into account various evironmental effects and preparing the input for the ML algorithm.

Conversely, there is a general agreement about the ML techniques recommended for supervised and unsupervised learning; we note that we refer to unsupervised learning whenever the set used for AI training contains only data collected from structures in full, healthy, operational conditions, while supervised learning requires the training set to contain both regular and anomalous data.

We present a broad, automated solution that self adapts in a smart way to the widest set of structures equipped with sensors, covering all possible environmental and external conditions and providing unsupervised as well as supervised solutions. In our scheme, data-driven and model-based algorithms are designed to coexist; moreover, the two strategies are mutually reinforcing, meaning that each of them compensates for the limitations of the other if considered alone for monitoring purposes. Nevertheless, the data-driven part keeps its supremacy within the proposed framework, as it can be run regardless of the realization of a digital twin of the monitored structure.

As we intend for our algorithm to catch all possible damages according to the broadest definition given by Farrar and Worden (2007), we constructed two different AI solutions that work in parallel, to independently detect both local and global damages, if any. In the following sections, we outline our monitoring solutions. Our very first input consists of 3-dimensional time series of linear accelerations, angular velocities and inclinations, all sampled at the frequency of 1 KHz[1], while the temperature value is appended to the features list at a later stage in order to mitigate its effects on the data. Optimal sensor placement is taken for granted in this context; optionally, filters for noise removal can be applied to time series before any other data processing method.

[1]The unit adopted for data acquisition is the MonStr device provided by ASDEA Hardware.

3. THE DATA-DRIVEN ALGORITHM

Here we introduce the portion of the framework dedicated to data analysis and anomaly detection. In what follows, when referring to data we intend the time series of monitored quantities collected within a fixed time window. The length of a single time window is a user-defined parameter, although it is recommended to adopt a value greater than or equal to 100 times the fundamental period of the reference structure for optimal modal parameter estimation [Ubertini et al. (2013)]. Time windows can be immediately consecutive or partially overlapping.

3.1. Univariate analysis. Local detectors

For a given time window:

- for any 1-*d* time series (i.e, for each physical quantity and each of the three coordinates), a set of temporal, spectral and statistical quantities are computed. Following the work of Buckley et al. (2022), we exploit the TSFEL library for Python in our setup for feature extraction [Barandas et al. (2020)], cutting all Fourier coefficients above 30 Hz
- alternatively, features are arranged as follows:
  - unpacked: to construct independent classifiers, features extracted from different devices are kept distinguished to feed separate algorithms
  - compacted together. Information coming from devices is mixed to feed a single algorithm
- temperature (and any other information about environmental conditions, if measured) is added to features in both cases

3.2. Multivariate analysis. Global detectors

Accelerometric time series are packed together to extract modal parameters. The user can run SSI-cov and SSI-data algorithms [Peeters and De Roeck (2001)] to extract natural frequencies, damping ratios and modal shapes. Multivariate statistics is also applied to get classical indicators, again using the dedicated functions the TSFEL library makes available to perform this task.

3.3. Feature reduction

Regardless of feature arrangement, feature projection on a subspace of lower dimension is required to get rid of data redundancy, and account for environmental effects. Linear Principal Component Analysis (PCA) is adopted to accomplish this task after performing feature rescaling (through simple standardization or min-max scaling). The resulting features, which retain a fixed percentage of total variance, consist of linear combinations of the original ones and are mutually independent. Once computed for the training dataset, the same mathematical objects that perform data rescaling and projection are employed for test-data pre-processing to maintain data consistency.

3.4. Unsupervised training

In this case, training data are acquired from physical structures in an entirely healthy state so that AI has no chance to learn how to directly identify possible anomalies embedded in the data. The user can choose from among various algorithms belonging to the family of artificial neural network autoencoders (AE) best suited to their needs [Bank et al. (2020)]. The logic of such deep learning methods lies in the capacity to suitably reconstruct the inputs produced by the same process which generates the training instances, and badly reconstructing any instance whose underlying, production process differs from the "healthy" one. This is realized by inferring the statistics of reconstruction error from training data, that is the normalized ℓ^2 distance between the input and its reconstructed counterpart, then introducing a threshold to distinguish regular from anomalous instances (see Figure 1). Also, anomalous trends can be identified by keeping track of reconstruction errors over time, to monitor slow parameter variations that correspond to structure deteriorations.

3.5. Supervised training

In this case, training data can be split into two or more subsets, one corresponding to data coming from the structure in the healthy state, while the others concern various damaged scenarios. In its more basic form, we only distinguish between damaged/undamaged instances; however, if more accurate information is provided, the damaged subset may be furtherly split into smaller subsets according to damage location and severity (see e.g. the approach used in Parisi et al. (2022)). The production of damaged datasets typically requires the construction of the digital twin of the structure, which enables the user to generate dynamics for regular and custom damaged settings.

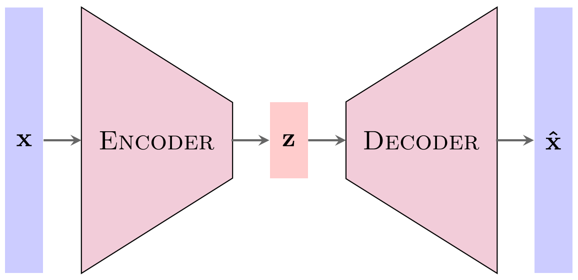

Figure 1: Schematic representation of an autoencoder. The input data $\mathbf{x}$ *is mapped to the lower dimensional vector* $\mathbf{z}$ *through an artificial neural network (the encoder); then,* $\hat{\mathbf{x}}$ *is obtained from* $\mathbf{z}$ *through the action of another artificial neural network that is symmetric with respect to the encoder (the decoder). The training phase aims to let the parameters of the network converge to the values that make* $\hat{\mathbf{x}}$ *similar to* $\mathbf{x}$ *in some suitable metrics.*

Supervised ML algorithms are preferred in this case to exploit the flagged information contained in the data, bypassing any statistical study that is needed a posteriori in the unsupervised case. Variations of the multilinear perceptron (see Figure 2) are advised with a softmax function as an activator for the output layer in order for the output to directly classify the input instance as regular or anomalous. If in the training phase the data has been further refined, i.e., the output is non-binary, damage location and severity can also be inferred [Parisi et al. (2022)].

Alongside Deep learning algorithms, ensamble learning methods like random forests [Ho (1995)] are successfully used in multiclass classification problems with improved accuracy with respect to artificial neural networks [Breiman (2001)], although the training phase is usually costly from a computational point of view (see Figure 3). However, the computational time taken by the two strategies for the inference phase is generally comparable. The use of random forests may also prevent algorithm overfitting [Zhou et al. (2013)].

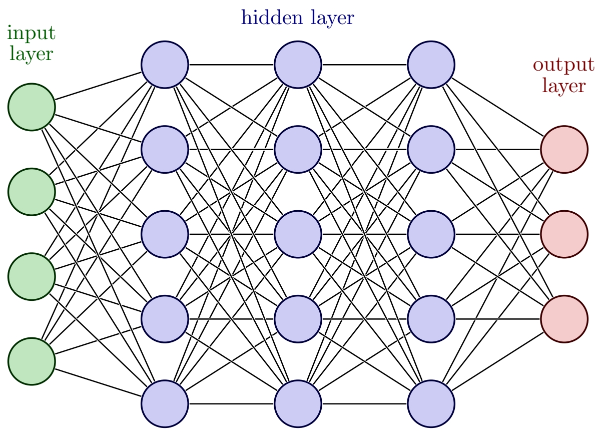

Figure 2: Schematic representation of a multilayer perceptron. The input instance is mapped to a binary output for a classifier that distinguishes between regular and anomalous inputs, and has more than two neuronal components for a finer anomaly classification.

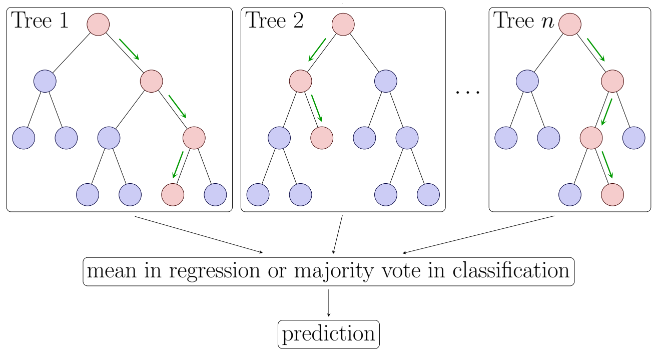

Figure 3: Schematic representation of a random forest of n decisional trees. Each tree is independently trained using a different portion of the training set and works as a multiclass classifier. In the inference phase, the input instance is mapped to the mean class in regression or to the class which gets the majority of votes.

4. THE MODEL-BASED ALGORITHM

When considering the model-based situation alone, we refer to the model-based solution as the algorithm that uses dynamic data acquired by sensors as input forces for the discrete-time evolution of the FEM model corresponding to the structure under examination. The idea is to let the model evolve according to the input data, periodically checking its overall conditions. This approach has the advantage of providing a direct, clear response compared to a data-driven solution that needs statistical interpretation of the received output, particularly when additional information beyond a rough reporting of anomalies is requested. The user can also visually inspect the behavior of the structure under given loads, and interpretation of results is thus immediate. The potential and limitations of this method, i.e., the reliability of the outputs, are totally related to the accuracy with which the numerical model reproduces the physical structure in terms of geometry, material, physical properties, and meshing. Moreover, as such properties are mutable over time and affect structural dynamics, a periodic updating of the model is requested in order for the actual facility and its digital twin to dynamically match. Our solution uses genetic algorithms to minimize the appropriately chosen metric that account for differences among natural frequencies and modal shapes (see Levin and Lieven (1998)). The tunable parameters for minimization are the elastic moduli of FEM elements.

We adopted STKO for the model-based portion as it is a straightforward solution to meet our need to maximize software and I/O interaction, as STKO's Python scripting interface permits the user to customize and program the pre and postprocessors as needed. Moreover, STKO includes all materials, elements, conditions and interactions offered in OpenSees, and also hosts all the codes needed for the AI software to perform.

5. MAIN ADVANTAGES OF OUR HYBRID SOLUTION

Our proposal for completely automated monitoring represents a step towards the establishment of a well-functioning standard for SHM, without reinventing the wheel of digital-twinning. In the following bulleted list, we highlight the pros of adopting our hybrid method, considering that our machinery becomes full data-driven in the absence of a digital replica of the structure. Furthermore, as with every digital twin framework, the numeri-

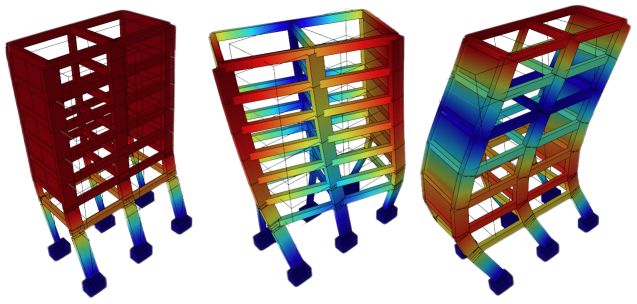
Figure 4: Example of a FEM model of a structure modeled within STKO. The figure shows different mode shapes.

cal model can be used for "what-if" analysis and, mainly, to enrich the training set with custom damaged datasets, thus unlocking access to supervised methods.

5.1. Our improvements

- Double-check anomalies detected by the data-driven solution: in order to prevent false alarms, potential anomalies are verified by running the corresponding numerical dynamics and noticing if damage is really produced.
- Damage type, localization and extension: if a binary classifier is employed as an AI detector, a numerical simulation provides missing information regarding damage characterization, suggesting possible actions to the user (closure of the facility, possible repairs).
- Software compatibility: the entire workflow that realizes the sequence of operations described in the previous sections is perfectly embedded in the STKO program. The hardware and software solutions are designed by the same developers to accomplish a precise, specific task and be integrated one with the other. Hence, there is no need to integrate software programs produced by different companies.
- Data format compatibility: in the various steps, all data are organized and exchanged in the HDF5 format. The central node itself rearranges measurements collected by distributed devices in an HDF5 file. This way, there is a complete inner coherence that prevents any data loss that often occurs in the process of conversion to different data formats, that is also time-wasting.
- Smart use of CPUs and GPUs: dedicated, powerful graphic cards perform parallel computing for the AI part as the TensorFlow Python library for deep learning distributes by default independent processes on the GPU. The TSFEL library for features extraction gives the user the opportunity to split processes on available threads for a quicker response. Concerning the model-based algorithm, STKO supports multi-processing solvers of OpenSees.

Figure 5 depicts the workflow described in this section.

6. CONCLUSIONS

Digital twin implementation stands at the frontier of research in Civil and Computer Engineering, thus far representing the more reliable solution for automating the damage detection process for physical structures. We proposed our framework for letting data-driven and model-based approaches coexist and mutually reinforce each other, avoiding typical issues encountered in the realization of the scheme and motivating our algorithmic choices. The use of specifically designated software improves our solution's stability and strength, while the adoption of a unique data format for the whole process eliminates any possible incompatibilities or data loss. In the future, we will spend more time challenging our algorithm with simulated and real data and checking the accuracy of the outputs obtained. Focus will be placed on developing more user options so that the machine can adapt perfectly even to very unique structures requiring monitoring services.

7. REFERENCES

Bank, D., Koenigstein, N., and Giryes, R. (2020). "Autoencoders." *arXiv preprint arXiv:2003.05991.*

Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T., and Gamboa, H. (2020). "Tsfel: Time series feature extraction library." *SoftwareX*, 11.

Breiman, L. (2001). "Random forests." *Machine learning*, 45.

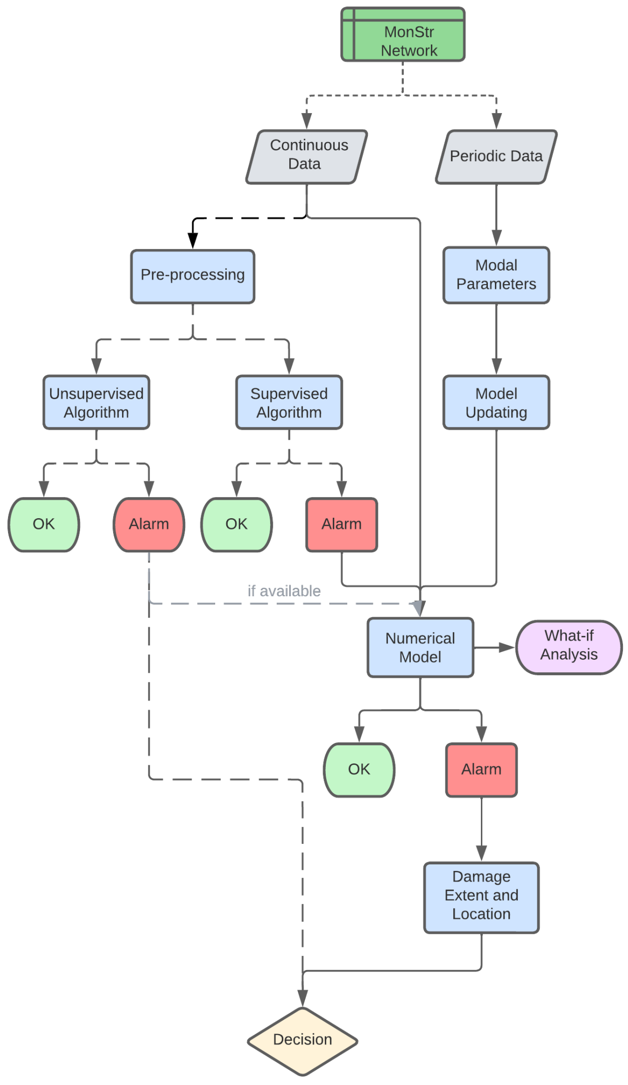

Figure 5: Flowchart of the process from data acquisition to output emission and decision making. Dotted lines refer to the flow of data acquired from the sensor network, dashed lines refer to connections among algorithms related to the data-driven portion, continuous lines to model-based processes. Continuously recorded data feeds data-driven methods, and numerical dynamics come into play whenever an alarm is emitted. If damage signaled by the alarm is confirmed, its extent and location is also evaluated. Periodically, modal parameters are extracted and employed to update the digital twin, which can be also used to perform a "what-if" analysis.

Buckley, T., Bidisha, G., and Pakrashi, V. (2022). “A feature extraction & selection benchmark for structural health monitoring.” *Structural Health Monitoring*.

Farrar, C. R. and Worden, K. (2007). “An introduction to structural health monitoring.” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365.

Ho, T. K. (1995). “Random decision forests.” *Proceedings of 3rd international conference on document analysis and recognition*, 1.

Levin, R. I. and Lieven, N. A. J. (1998). “Dynamic finite element model updating using simulated annealing and genetic algorithms.” *Mechanical systems and signal processing*, 12(1).

Parisi, F., Mangini, A. M., Fanti, M. P., and Adam, J. M. (2022). “Automated location of steel truss bridge damage using machine learning and raw strain sensor data.” *Automation in Construction*, 138.

Peeters, B. and De Roeck, G. (2001). “Stochastic system identification for operational modal analysis: a review.” *J. Dyn. Sys., Meas., Control*, 123(4).

Petracca, M., Candeloro, F., and Camata, G. (2017). “Stko user manual.” *ASDEA Software Technology, Pescara, Italy*.

Tibaduiza Burgos, D. A., Gomez Vargas, R. C., Pedraza, C., Agis, D., and Pozo, F. (2020). “Damage identification in structural health monitoring: A brief review from its implementation to the use of data-driven applications.” *Sensors*, 20(3).

Ubertini, F., Gentile, C., and Materazzi, A. L. (2013). “Automated modal identification in operational conditions and its application to bridges.” *Engineering Structures*, 46.

Zhou, Q., Ning, Y., Zhou, Q., Luo, L., and Lei, J. (2013). “Structural damage detection method based on random forests and data fusion.” *Structural Health Monitoring*, 12(1).