 https://doi.org/10.3311/PPci.21756

Periodica Polytechnica Civil Engineering, 67(2), pp. 416–430, 2023

Robust Vibration Output-only Structural Health Monitoring Framework Based on Multi-modal Feature Fusion and Self-learning

Viet-Hung Dang[1*], Truong-Thang Nguyen[1]

[1] Faculty of Building and Industrial Construction, Hanoi University of Civil Engineering (HUCE), No.55 Giai Phong Rd., Hai Ba Trung Dist., Hanoi, Vietnam
[*] Corresponding author, e-mail: hungdv@huce.edu.vn

Received: 26 December 2022, Accepted: 13 January 2023, Published online: 26 January 2023

Abstract

Output-only structural health monitoring is a highly active research direction because it is a promising methodology for building digital twin applications providing near-real-time monitoring results of the structure. However, one of the technical bottlenecks is how to work effectively with multiple high-dimensional vibration signals. To address this question, this study develops a two-stage data-driven framework based on various advanced techniques, such as time-series feature extractions, self-learning, graph neural network, and machine learning algorithms. At first, multiple features in statistical, time, and spectral domains, are extracted from raw vibration data; then, they subsequently enter a graph convolution network to account for the spatial correlation of sensor locations. After that, the high-performance adaptive boosting machine learning algorithm is leveraged to assess structures' health states. This method allows for learning a lower-dimensional yet informative representation of vibration data; thus, the subsequent monitoring tasks could be performed with reduced time complexity and economical computational resources. The performance of the proposed method is qualitatively and quantitatively demonstrated through two examples involving both numerical and experimental structural data. Furthermore, comparison and robustness studies are carried out, showing that the proposed approach outperforms various machine learning/deep learning-based methods in terms of accuracy and noise/missing-robustness.

Keywords

structural health monitoring, vibration, signal processing, machine learning, graph neural network

1 Introduction

Vibration output-only structural health monitoring (SHM) is an appealing method for effectively monitoring the operation state of civil structures. This method does not require engineers to always be on the field or interrupt the structures' services; thus, it is possible to perform SHM continuously and in the long term. However, there are two major obstacles that may hinder the applicability of this method which are: i) a massive amount of collected vibration signals, ii) unwanted noises, and missing data. In practice, with dozen accelerometers, one can collect gigabytes of acceleration data for a few hours. Hence, directly using such data will require huge storage devices and complex data analytic models; however, prediction results may be already known information, such as a healthy state without damage occurrence. That is why using feature extraction to convert high-dimensional time series data to low-dimension feature vectors is more amenable and economical in terms of budget and effort. Gui et al. [1] used sixteen statistical features, such as maximum, minimum, mean, skewness, kurtosis, etc., to build a rapid detecting damage application, which was successfully validated through experimental data from a laboratory three-story frame structure. When working with rolling element bearing, Mathew and Alfredson [2] found that among more than twenty investigated statistical features, kurtosis is a damage-sensitive one that can provide a good indication of failure states. Yanez-Borjas et al. [3] proposed to combine a set of statistical features, including high-order moments (up to 6^{th}) and shape-related features, with the Principal Component Analysis algorithm and the Mahalanobis distance metric to perform SHM of load-bearing cables in a cable-stayed bridge. For SHM of wind turbines, root mean square is widely used as an effective parameter in identifying damage existence.

Cite this article as: Dang, V.-H., Nguyen, T.-T. "Robust Vibration Output-only Structural Health Monitoring Framework Based on Multi-modal Feature Fusion and Self-learning", Periodica Polytechnica Civil Engineering, 67(2), pp. 416–430, 2023. https://doi.org/10.3311/PPci.21756

Other shape-related statistical parameters, such as shape factor, crest factor, range, impulse factor, etc., also have significant contributions to SHM results [4].

Statistical feature-based methods are practical and fast; however, they cannot take into account chronological relationships, such as autocorrelations, trends, or seasonal variations, within time series data. To improve this shortcoming, various authors have proposed to use time-series modeling techniques to simulate the structures' time response histories, then estimate the actual structures' states. Autoregressive model (AR) is one of the most popular time-series fitting techniques with clear and easy-to-understand theory backgrounds. Figueiredo et al. [5] proved that the AR model was successfully applied to real-world structures under operational and environmental conditions. For structures under random excitation, Carden and Brownjohn [6] developed a SHM method based on the Autoregressive Moving Average model, which was successfully validated through a number of examples, including a 4-story 3D frame structure, the Z24 bridge and the Singapore–Malaysia Second Link bridge. Zheng and Mita [7] also employed ARMA to identify damage locations and assess the severities of a five-story building structure subjected to various excitations, i.e., earthquake and ambient forces.

Although time-domain features are good damage indicators, however, they are easily altered by environmental noise and other unfavorable factors such as device instability, unequally sampled discrete data, etc. Hence, frequency-domain feature-based methods have been investigated as promising alternatives. Pehlivan et al. [8] utilized multiple signal analysis techniques in the frequency domain to monitor the behavior of a 240 m high television tower. The techniques included low-pass filtering for removing low-frequency noise components and a high-pass filter for separating instantaneous high-frequency components. The most common technique to transform data from the time domain to the frequency domain is to use the Fourier Transformation (FT); particularly, the Fast Fourier Transformation algorithm is highly effective when working with large time-series data. A detailed description of the SHM method based on the FT was demonstrated in [9], which involved computing the FT of output responses, and FT of input excitation, then deriving frequency response function (FRF) parameters. The FRF data were compared with those of healthy reference structures to derive the structure's actual state. Another effective method in the frequency domain for detecting damage existence is to assess the change of power spectral densities (PSD). Examples of methods utilizing PSD are the works of Pedram et al. [10] for concrete beam structure.

Furthermore, to combine the advantages of time- and frequency-domain features, various time-frequency approaches have been developed to investigate the evolution of frequency features over time. Specifically, the Short-Time Fourier Transform method (STFT) divides signals into shorter segments of equal length; then FT is applied to each segment; thus, a 1D time series is converted to a 2D image of time-frequency-amplitude representations [11]. However, a shortfall of the STFT technique is only using a fixed-width window to extract all frequency contents within signals. Meanwhile, for low-range frequency contents, a long window should be used, whereas for high-range frequency contents, a shorter window is more convenient. Hence, the Wavelet Transform (WT) technique is widely adopted [12], using varying-length windows to extract frequency information at different resolutions.

Other damage-sensitive indicators are modal features, including eigenfrequencies and mode shapes that can be extracted from vibration signals. These features can then be utilized along with meta-heuristic optimization algorithms to assess structures' health, as done in the work of Kaveh et al. [13]. For high-order indeterminate truss structures, Kaveh and Mahdavi [14] proposed a method using natural frequencies for identified damaged members. Subsequently, the authors demonstrated that using mode shapes as input provided better structural damage detection results [15]. Furthermore, features derived from mode shapes also considerably contributed to the model performance, for example, the modal assurance criterion [16] and modal strain energy-based indexes [17]. It is noted that to extract modal features, complex theories such as modal operation analysis and specialized software, e.g., MACEC [18], are needed.

2 Research significance

Although using features from a specialized domain could provide promising results for some specific structures in some specific conditions, the SHM performance may not be generalized and can be lowered for other types of structures or impeded by unwanted factors. A natural intuition is to fuse features from different domains into one single input vector to leverage the advantages of different strategies. Therefore, this study develops a multi-modal feature fusion framework that first distills features from statistical, temporal, and spectral domains. Secondly, meaningful

information in the spatial domain, e.g., the structure's connectivity, sensor placements [19, 20], is incorporated into data via the graph convolution network. Third, the interaction between features could be hallucinated via a self-supervised learning (SSL) technique. Finally, an adaptive boosting machine learning model is leveraged to assess the corresponding structure's health states. Concretely, the main contributions of this study are summarized as follows:

- A novel framework for structural health monitoring is proposed that can convert high-dimension multivariate signals to significantly lower-dimension feature vectors while still providing accurate SHM results.
- The viability of the mFF-SHM method is quantitatively and qualitatively highlighted via two case studies, including a numerical 2D asymmetric semi-rigid frame structure and an experimental database of a 3D steel frame structure. The comparison results show that the proposed method outperforms various ML/DL-based counterparts.
- Thanks to the domain diversity and self-supervised learning, mFF-SHM is robust against noisy or incomplete data, i.e., providing reliable SHM results with contaminated vibration signals.

The rest of the paper is organized as follows: Section 2 presents the key components of the proposed framework involving multi-modal feature extraction, graph convolution network, self-supervised learning with triplet loss function, and adaptive boosting machine learning. In Section 3, the viability of the proposed approach is demonstrated through two examples. Finally, in Section 4, one draws conclusions and proposes perspectives for future work.

3 Multi-modal feature fusion structural health monitoring framework

In this section, one describes the proposed multi-modal feature fusion structural health monitoring framework, which consists of two main stages, as shown in Fig. 1. The first stage is self-supervised learning, and the second stage is supervised classification. The first stage aims to learn low-dimensional, informative feature-based representations from raw vibration data which involve information from statistical, temporal, spectral, and spatial domains. The second stage will leverage learned representation from the first stage to assess actual structural states. The details of each component in the working flow will be presented in the following.

3.1 Vibration signal feature extraction

Feature extraction aims to transform high-dimensional vibration data into lower-dimensional vectors that still preserve as much as possible underlying characteristics helpful in detecting the structures' actual state. Towards a generalized SHM framework, one will consider features in multiple domains, including statistical, temporal, and spectral domains, rather than relying on only one domain

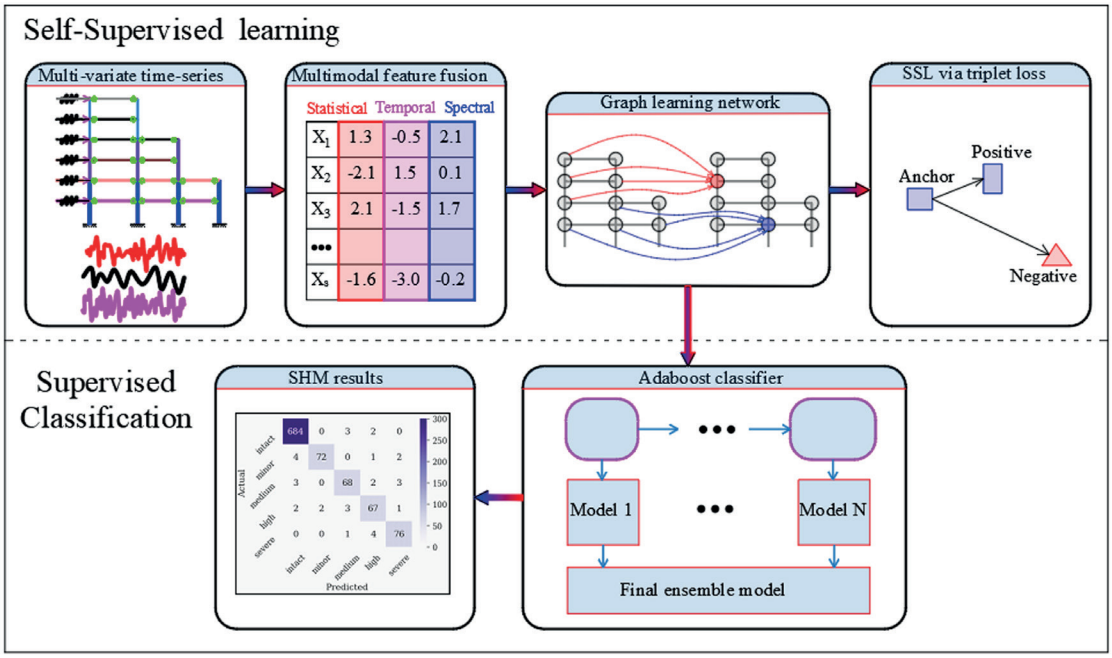


Fig. 1 Working flow of the proposed mFF-SHM framework

that may work well with a specific class of structures but is unsuitable for others. Before describing in detail the adopted features, some notations are introduced. Let's denote a time-series data, L is its length, t is the time instant, f_s stands for sampling frequency, and s' represent the discrete derivative of s with respect to t.

Statistical features are the most common and relatively simpler features, which can be derived using some straightforward mathematical equations and based on only data amplitude values but not time instants. The first features of interest are maximum, minimum, and mean values. The mean value is known as the first-order moment of data; in addition, one also considers higher moments such as the second moment (variance), third moment (skewness), and fourth moment (kurtosis). Furthermore, features showing information about the probability distribution of data values, such as third, second, and third quartile values, are also accounted for. These three values signify that 25%, 50%, and 75% of the number of values are less than them, respectively. Other features used by various authors for statistically characterizing the signal amplitude are also taken into account, such as mean absolute deviation, root mean square, and entropy. Hence, there are, in total, 12 statistical features of . Besides, one also calculates these features for the time derivative of signals and appends them to the feature vector.

The second group of features is the temporal features, which describe some underlying patterns over time within time-series data. Unlike statistical features, in order to compute temporal features, one needs to account for both amplitude values and corresponding time instants. For example, the centroid is calculated by:

$$centroid = \frac{\sum_{i=1}^{L} t_i \times s_i^2}{\sum_{i=1}^{L} s_i^2}. \tag{1}$$

The total energy of a signal is computed by:

$$energy = \frac{\sum_{i=1}^{L} s_i^2}{t_L - t_1}. \tag{2}$$

The area under the curve is given by:

$$AUC = \sum_{i=1}^{L} \left(t_i - t_{i-1}\right) \times \frac{s_i + s_{i-1}}{2} \tag{3}$$

Some temporal features for which programming algorithms are needed are the number of peaks, zero crossing rate, and autocorrelation.

The third feature group is the spectral features, which require conducting some transformation in advances, such as Fourier transformation or wavelet transformation. The Fourier transformation will result in vectors of frequency values and corresponding magnitude values based on which spectral features are derived, such as the fundamental frequency, FFT mean coefficient, spectral centroid, and maximum power. Furthermore, one can calculate the statistical characteristics of these spectral values as done with temporal features such as kurtosis, skewness, variance, and entropy. Another way to extract spectral features is to use the wavelet transformation, resulting in a set of wavelet coefficients. Similar to Fourier transformation, some statistical characteristics of wavelet coefficients are derived, such as wavelet absolute mean, wavelet variance, wavelet energy, and wavelet entropy. These features are extracted with the help of a number library such as Tsfel [21], Sktime [22], and Tsfresh [23]. To sum up, the list of extracted features is enumerated in Table 1.

3.2 Graph convolution network

Conventionally, a graph is a type of data consisting of nodes and edges between them, which could present the relationship between nodes in an explicit way. Let's denote a graph by $G = (N, E)$, with N, E being its set of nodes and edges, respectively. The relationship between nodes can be mathematically described via a binary adjacency matrix A whose component $A_{i,j} = 1$ if there is an edge connecting node i with j; in contrast, $A_{i,j} = 0$. The group of nodes j with $A_{i,j} = 1$ is called the neighbor of node i and is denoted by $ne(i)$. For a civil structure, nodes are usually joints or connections, and edges are structural members. Each node is characterized by a vector of features extracted from vibration signals, as described above. In order to effectively incorporate the structure's spatial information into the structural database, on top of statistical, temporal, and spectral features, the graph convolution network [24] is resorted to. This is a special type of neural network that perform neighborhood aggregation operations written as follows:

$$\boldsymbol{h}_i = \sigma\left(\boldsymbol{W}_g \times \frac{\sum_{u \in ne(i)} x_u}{\left|ne(i)\right|} + \boldsymbol{B}_g x_i\right), \tag{4}$$

where $\boldsymbol{h}_i$ is hidden states of node i, $\boldsymbol{W}_g$ and $\boldsymbol{B}_g$ are trainable weights and biases. σ is a non-linear activation function. It is noteworthy that the equation involves only $ne(i)$, not the entire graph; thus, the operation is highly computationally efficient, especially with sparse adjacency matrices.

Table 1 List of extracted features used in the mFF-SHM framework

No	Feature description	Abbreviation	No	Feature description	Abbreviation
1	Entropy of the signal	entropy	25	Area under the curve of the signal	AUC
2	Root mean square of the signal	rms	26	Sentroid along the time axis	Centroid
3	Skewness of the signal	skewness	27	Total energy of the signal	Energy
4	Kurtosis of the signal	kurtosis	28	Autocorrelation of the signal	Autocorr
5	Mean value of the signal	mean	29	Zero-crossing rate of the signal	zero-cross
6	Standard deviation of the signal	std	30	Number of peaks from a defined neighbourhood of the signal.	Peaks
7	Maximum value of the signal	max	31	Maximum power spectrum density of the signal	max_PS
8	Minimum value of the signal	min	32	Fundamental frequency of the signal	1st-freq
9	First quartile value of the signal	Q1	33	Centroid of the signal spectrum	centroid-S
10	Median value of the signal	median	34	Entropy of the signal spectrum	entropy-S
11	Third quartile value of the signal	Q3	35	Skewness of the signal spectrum	skewness-S
12	Mean absolute deviation value of the signal	MAD	36	Kurtosis of the signal spectrum	kurtosis-S
13	Entropy of the time derivative of the signal	entropy-diff	37	Variation of the signal spectrum	variation-S
14	Root mean square of the time derivative of the signal	rms-diff	38	CWT absolute mean value of the first wavelet scale	mean-W-lv1
15	Skewness of the time derivative of the signal	skewness-diff	39	CWT absolute mean value of the second wavelet scale	mean-W-lv2
16	Kurtosis of the time derivative of the signal	kurtosis-diff	40	CWT absolute mean value of the third wavelet scale	mean-W-lv3
17	Mean value of the time derivative of the signal	mean-diff	41	CWT energy of the first wavelet scale	energy-W-lv1
18	Standard deviation of the time derivative of the signal	std-diff	42	CWT energy of the second wavelet scale	energy-W-lv2
19	Maximum value of the time derivative of the signal	max-diff	43	CWT energy of the third wavelet scale	energy-W-lv3
20	Minimum value of the time derivative of the signal	min-diff	44	CWT entropy of the signal	entropy-W
21	First quartile value of the time derivative of the signal	Q1-diff	45	CWT standard deviation of the first wavelet scale	std-W-lv1
22	Median value of the time derivative of the signal	median-diff	46	CWT standard deviation of the second wavelet scale	std-W-lv2
23	Third quartile value of the time derivative of the signal	Q3-diff	47	CWT standard deviation of the third wavelet scale	std-W-lv3
24	Mean absolute deviation value of the time derivative of the signal	MAD-diff			

3.3 Self-supervised learning using triplet loss

The triplet loss technique is a convenient way to perform SSL that has been successfully applied in various domains since its first introduction [25]. The key idea of this technique follows a natural intuition that samples belonging to the same class will exhibit similar characteristics. Thus, if one introduces a metric for measuring the similarity distance, distances between similar samples should be small; otherwise, those between dissimilar samples should be significantly large. As described in the previous subsection, each data sample is characterized by a vector of multi-modal features. Thus, some commonly used similarity distances for real-valued feature vectors are Euclidean distance, a.k.a L2 norm, or cosine distance. Cosine distance is more utilized when the magnitude of the vectors has less significance than their orientation. Meanwhile, for vibration, the signal amplitude is also an important indicator; hence, the Euclidean distance is adopted.

Next, the formula of triplet loss can be written as follows:

$$L = \sum_i^N \left[f\left(\boldsymbol{x}_i^a\right) - f\left(\boldsymbol{x}_i^p\right)_2^2 - f\left(\boldsymbol{x}_i^a\right) - f\left(\boldsymbol{x}_i^n\right)_2^2 + \alpha \right], \quad (5)$$

where $\boldsymbol{x}_i^a$ represents an anchor sample, and $\boldsymbol{x}_i^p$ denotes a positive sample with the same label as $\boldsymbol{x}_i^a$. A positive sample is obtained by applying data augmentation techniques to the original sample. For time series, some commonly adopted data augmentation techniques are jittering (noise injection), scaling, permutation, flipping, and zeroing. Representative examples of these techniques are illustrated in Fig. 2. On the other hand, $\boldsymbol{x}_i^n$ is a negative sample that is labeled differently from $\boldsymbol{x}_i^a$, $\|\cdot\|_2^2$ stands for Euclidean distance. $f(\boldsymbol{x})$ is the feature vector of sample $\boldsymbol{x}$. In Eq. (6), the first term signifies the distance between the anchor and positive samples that needed to be minimized. The second term with a negative sign is the distance between the anchor and negative samples that needs to be maximized. Moreover, a small value α is introduced as a threshold to improve the calculation stability of the loss function gradient. Fig. 3 graphically represents the learning process with the triplet loss function. For example, in the beginning, the negative sample is closer to the anchor than the positive one. However, after finishing the learning process, the positive sample approaches the anchor more than the negative one by at least a distance of α.

In terms of model hyperparameters, the batch size is set to 256, the adopted optimizer is the Adam optimization algorithm, and the learning rate is 0.001. Note that for a batch of 256 samples, one applies first augmentation techniques to form 3 × 256 augmented samples. After that, one randomly selects 256 triplets from these 3 × 256 samples, each triplet consists of 3 samples ($\boldsymbol{x}_a$, $\boldsymbol{x}_p$, $\boldsymbol{x}_n$) where $\boldsymbol{x}_p$ is an augmented variant of $\boldsymbol{x}_a$, and $\boldsymbol{x}_n$ and $\boldsymbol{x}_a$ belong to different classes. The mFF-SHM framework is developed by the authors with the help of the Python programming language and machine learning libraries such as Pytorch, Pytorch-geometric [26], and PyCaret [27]. Computations were conducted on an Intel Xeon CPU E5-2650 server equipped with 2 Nvidia 3080 GPUs and 64 Gb RAM.

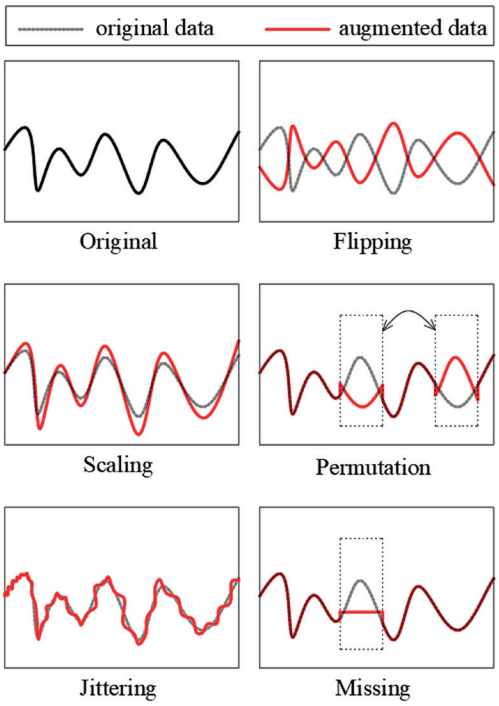


Fig. 2 Data augmentation techniques for time-series data

3.4 Adaptive boosting machine learning algorithm

Among machine learning algorithms, boosting model is one of the highest-performance models that combine multiple base ML models into a strong one. Each individual model focuses on a different set of features and utilizes different strategies to perform prediction, and it has its own drawbacks. By using multiple models, one increases the model diversity, reducing errors and improving the performance. Specifically, in this study, one uses decision tree (DT) models as weak models, each DT using different subsets of features, and the final model is obtained as a weighted sum of all DTs (Fig. 4). The weights assigned to DT models are adaptively updated in a sequential manner. Such an approach is known as the adaptive boosting technique (Adaboost) [28]. On the other hand, one also weights data samples to pay more attention to difficult samples that the DT models wrongly classified. This idea is graphically demonstrated in Fig. 5. Given 10 data samples belong to two classes, "rectangle" and "triangle". Initially, one assigns the same weight for these data samples and puts them through the first DT model. The DT will classify these data via a horizontal line and misclassify two sample points on the left. In the second step, the weights of the two wrongly classified samples are increased, as visually shown by the increased size of these rectangles. Next, these updated samples are entered into a second DT, which in turn, misclassifies three triangle samples. Hence, their

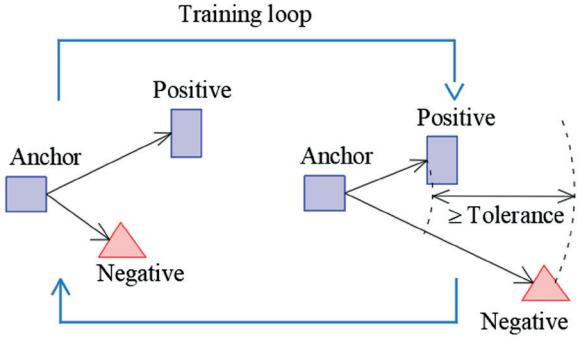


Fig. 3 Graphical representation of the triplet loss function

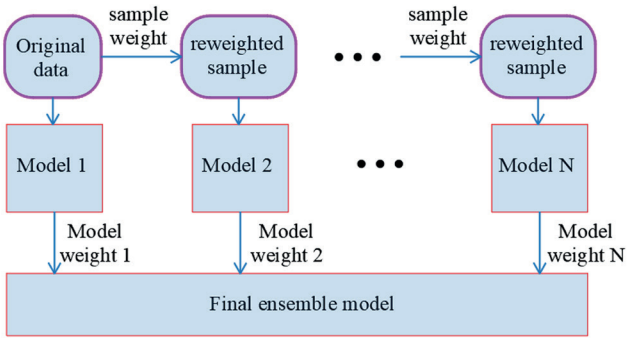


Fig. 4 Graphical illustration of the adaptive boosting algorithm

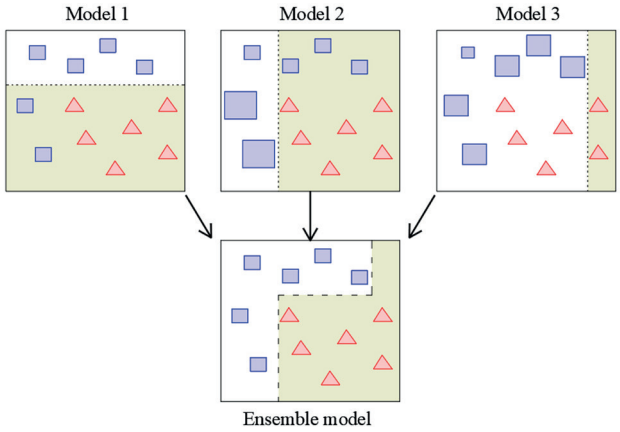


Fig. 5 A representative explanation of the Adaboost algorithm

three weights are adjusted correspondingly. The whole updated database then goes through a third DT model. Finally, all three models are combined into a strong fourth model, which correctly classifies data samples, as shown in the subfigure at the bottom.

Formally, let's denote $DT_k(X_i)$ as base models, with $k = 1, \ldots, N_{model}$, N_{model} is the total number of models, α_k is the weight assigned to $DT_k(X_i)$, X_i is a data sample with $i = 1, \ldots, N_{data}$, N_{data} is the total number of samples. Considering model k, error made by DT_k on X_i is denoted by $e_{i,k}$. Thus, the total error by DT_k is:

$$e_k = \sum_{i=1}^{N_{data}} e_{ki} . \tag{6}$$

Next, the weight of DT_k is computed as follows:

$$\alpha_k = \frac{e_k}{1-e_k} . \tag{7}$$

On the other hand, weight of each data sample i is updated for the next model $k + 1$ as below:

$$w_{k+1,i} = \frac{w_{k,i}\alpha_k}{\sum_{i=1}^{N_{data}} w_{k,i}\alpha_k} . \tag{8}$$

4 Case studies

4.1 Case study 1: Numerical 2D semi-rigid structure

The first example is a six-story 2D asymmetric semi-rigid steel frame structure. The frame has a bay width of 6 m and a total height of 25 m, as shown in Fig. 6. The cross-sections of the beams and columns are also detailed in the figure. The structure is excited by six independent time-varying concentrated loads acting on every story floor. In reality, steel members such as columns and beams are connected through bolt or welded connections with finite rotational stiffness and are hence usually considered semi-rigid connections. The semi-rigid connection is modeled via a rotational spring. More details about using graph theory for the description of semi-rigid skeletal structures can be found in [29, 30]. Other main structural parameters involving Young modulus, damping ratio, and loading amplitudes are considered random variables whose values are described via a predefined probability distribution and statistical characteristics as enumerated in Table 2. Specifically, the loading amplitudes are sampled from an uniform distribution with lower and upper bounds being 1000 and 2000 N. Young modulus value is drawn from a normal distribution with a mean of 200 GPa and a coefficient of variation (CoV) of 0.1. The dynamic analysis of the structure is realized by using the open-source finite element software OpenSees developed by the Pacific Earthquake Engineering Research Center [31]. The rotational spring is modeled through zero-length elements. For numerically

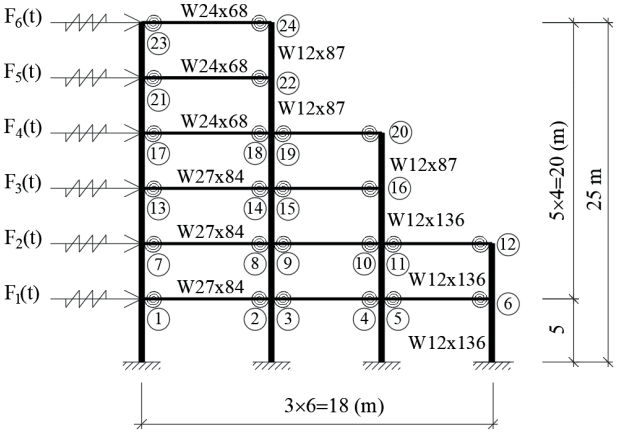


Fig. 6 Schematic representation of the semi-rigid frame structure

Table 2 Random variables of the numerical 2D semi-rigid structure

Variable	E (GPa)	ζ (%)	F (N)
mean	200	5	
CoV	0.1	0.05	
Dist.	Normal	Normal	Uniform [1000–2000]

solving the dynamic equilibrium equations, the Newton-Raphson algorithm and Newmark numerical integration scheme with $\gamma = 0.5$ and $\alpha = 0.25$ are utilized.

In this case study, the damage was introduced by randomly diminishing one or many rotational spring stiffnesses (up to 5). The remaining rotational stiffness can be in the range of (10^5, 10^6, 10^7, 10^8,), corresponding to four damage levels, i.e., severe, high, moderate, and minor, apart from the original intact state. Numerical simulations with these damages are carried out in OpenSees, and outputs of interest are vibration signals recorded at all the structure's column-beam connections with a frequency sampling of 100 Hz for 30 s. The results of each simulation are stored in a matrix of size [38 × 30000]. There are, in total, 5000 simulations with different structural parameters and different damage scenarios that were performed, resulting in vibration-based data of size [5000 × 18 ×30000]. It took around 31 s CPU time for each FEM simulation; hence, the total computational time required for data generation is around 43 hours.

With the structural database in place, the SSL training process with the settings detailed above was carried out. The resulting learning curve highlighting the evolution of the triplet loss versus the number of training epochs is shown in Fig. 7. Apparently, the loss function quickly decreased for the first 300 epochs; after that, it improved with a less noticeable rate and achieved the lowest value at epoch around 480. For the subsequent 50 epochs, no improvement was observed; hence, the training process was terminated per the early stopping criteria.

Once the SSL learning is finished, it is used to assess the structure's state with the help of an Adaboost-based classifier. Actually, for each semi-rigid connection, one train a specific classifier header with corresponding labels.

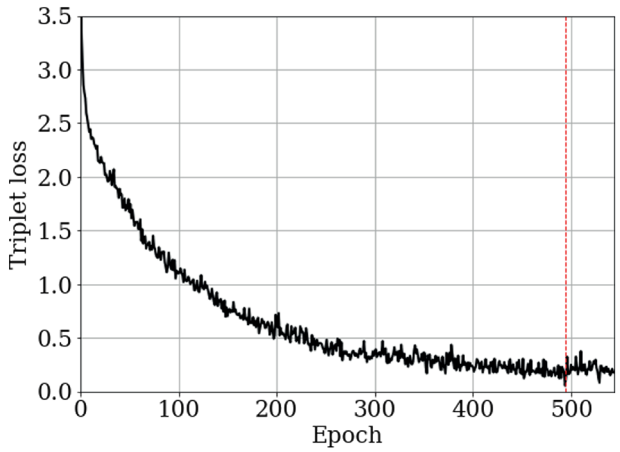


Fig. 7 Evolution of triplet loss function in function of training epochs

Note that a data sample could be labeled in different ways depending on the semi-rigid connections of interest. It is possible that many damages occur at the same time; thus, a data sample could be labeled as severe for connection i, but can be minor for connection j or even undamaged for connection k, with $i \neq j \neq k$. Considering connection 6 for example, the SHM results obtained by the proposed mSS-SHM are graphically presented in Fig. 8 via a confusion matrix.

Detailed errors made by the proposed method for every damage scenario can be derived from the off-diagonal cells of the confusion matrices. For example, the second row of the confusion matrix in Fig. 8 reveals that mFF-SHM misclassified 4, 1, and 2 samples belonging to minor damage cases into intact, high, and severe damages, respectively. In total, there are 33 wrongly identified samples, i.e., the error is 3.3%. It can be seen that the data are unbalanced as the number of samples corresponding to the intact state is clearly greater than those of other states. For such unbalanced data, the metric F1 is usually used to assess the model performance. Precisely, the F1-score for results in Fig. 8 is up to 93%, with a large part of wrongly classified samples related to the dominated "intact" class.

In order to clarifier the performance of the proposed method, a comparison study is conducted to compare mFF-SHM with a number of popular Machine Learning/Deep learning (ML/DL) based methods involving: Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor, Decision Tree, Naïve Bayes, Logistic Regression, Ridge classifier, Multiple Layer Perceptron (MLP), 1D-Convolution Neural Network (1DCNN), Long Short Time Memory (LSTM) and AdaBoost without SSL.

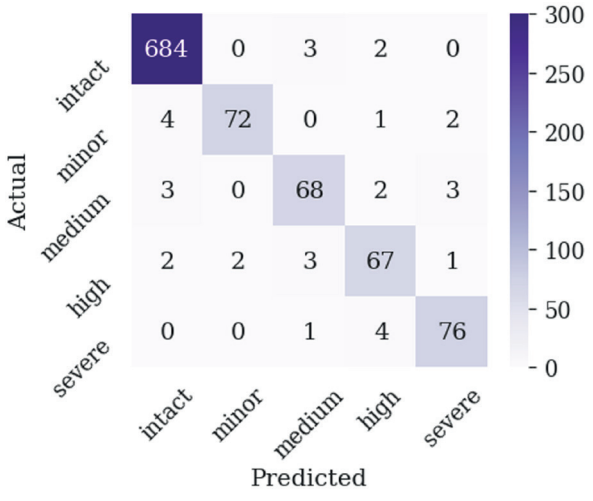


Fig. 8 SHM results via confusion matrix for semi-rigid connection 6 of the 2D frame structure

Comparison results are enumerated in Table 3, including F1-score, feature extraction time, self-learning time, training time for the classifier, and total CPU time. In addition, one can derive the percentages of errors as the difference between 100% and the accuracy values. Although accuracy and error are intuitive metrics, for imbalanced data in which a class (e.g., severe damage) may have significantly fewer samples than others, wrongly identified samples of this class could have more negative impacts on the SHM results. Thus, F1-score is a more relevant metric to assess the model's performance. Apparently, mFF-SHm achieves the highest F1 score (on the testing dataset) though when compared to other machine learning methods, mFF-SHM requires additional self-learning time of up to 18.2 min, thus making it slower than other ML feature-based methods. However, mFF-SHM is still approximately equal to or faster than deep learning models using high-dimensional raw vibration data, e.g., LSTM requires a total CPU time of 66.5 min, which is more than two times that of mFF-SHM (30.3 min).

Analogously, one repeats the computation process for other semi-rigid connections, including a variant of mFF-SHM, relabeling, training and validation. Note that all mFF-SHM models have the same architecture except for the classifier, whose parameters are determined with corresponding labeled data. In total, there are 24 variants of mFF-SHM for 24 semi-rigid connections. The performances of these models are depicted in Fig. 9, showing that F1 scores are slightly higher for connections belonging to edge columns than for those of inside columns. For example, F1-score for connections 1, 6, 7, 12, 13, 16, 17, 21, 22, 23, and 24 are about 95%, whereas, for other connections, F1-scores fluctuate around 90%. Such results can be partly explained as follows. For each joint inside the frame structure, there are two rotational springs; thus, if a spring is damaged, the other can still transfer vibrations between structural elements. Meanwhile, for joints on the edge columns, there is no alternative way for vibration propagation; thus, damages have more effect on the structure's responses. Overall, by combining these mFF-SHM

Table 3 Comparison results between the mFF-SHM framework with ML/DL counterparts for the 2D frame structure

No	Model	Accuracy (%)	F1 (%)	FE time (min)	Self-learning time (min)	Training time (min)	Total time (min)
1	mFF-SHM	0.95	0.94	6.3	18.2	5.8	30.3
2	Ada Boost Classifier	0.93	0.92	6.3	0	5.8	12.1
4	1DCCN	0.91	0.90	0	0	26.6	26.6
5	LSTM	0.89	0.87	0	0	66.5	66.5
6	SVM - Linear Kernel	0.86	0.83	6.3	0	2.2	8.5
7	K Neighbors Classifier	0.86	0.83	6.3	0	3.4	9.7
8	MLP	0.85	0.85	0	0	41.2	41.2
9	Decision Tree Classifier	0.82	0.82	6.3	0	2.1	8.4
10	Naive Bayes	0.73	0.72	6.3	0	0.9	7.2
11	Logistic Regression	0.65	0.62	6.3	0	2.2	8.5
12	Ridge Classifier	0.55	0.56	6.3	0	1.4	7.7

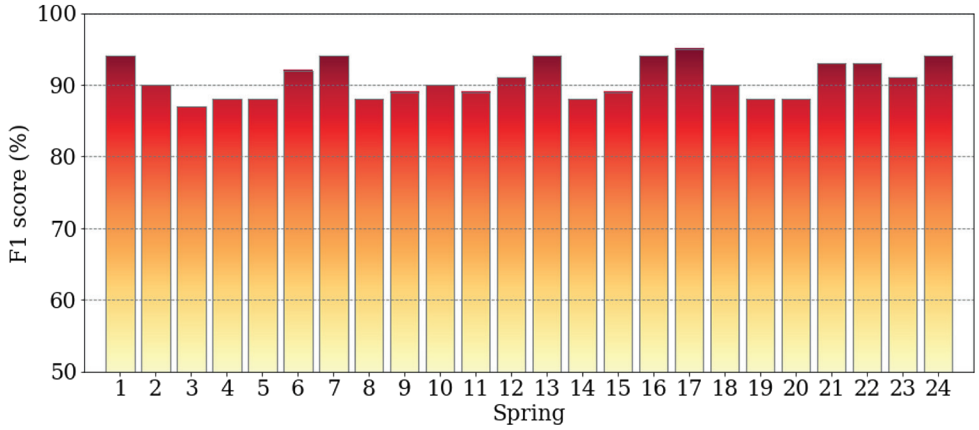


Fig. 9 SHM results for all 24 semi-rigid connections

variants, one can obtain reasonably accurate SHM results for multi-damage scenarios where different damages occur at different locations with different severities.

On the other hand, it is noteworthy to investigate the contribution of various features on the model performance, which help engineers and researchers understand better the structure's behaviors and select appropriate signal processing techniques for achieving the highest accuracy possible. One of the most popular techniques for this purpose is the permutation feature importance technique. Considering a feature, this technique randomly shuffles feature values among samples while keeping the other features unchanged. After that, the reduction in the model score is calculated; if the reduction is large, this feature is important; otherwise, it has little contribution to the final performance. The process is repeated for all features, score reductions are then normalized to derive feature importance scores, and the features will be ranked based on their scores.

The results of the feature importance study for the 2D frame example are illustrated in Fig. 10. It is interesting that the wavelet entropy is very sensitive to damage, whereas the contribution of the signal entropy is negligible. That is why wavelet transformation is one of the most effective techniques for SHM, widely used by various authors [12]. Other statistical features about the distributions of signal amplitudes, such as Q1, median (Q2), Q3, RMS, max,… are also ranked among the top important features.

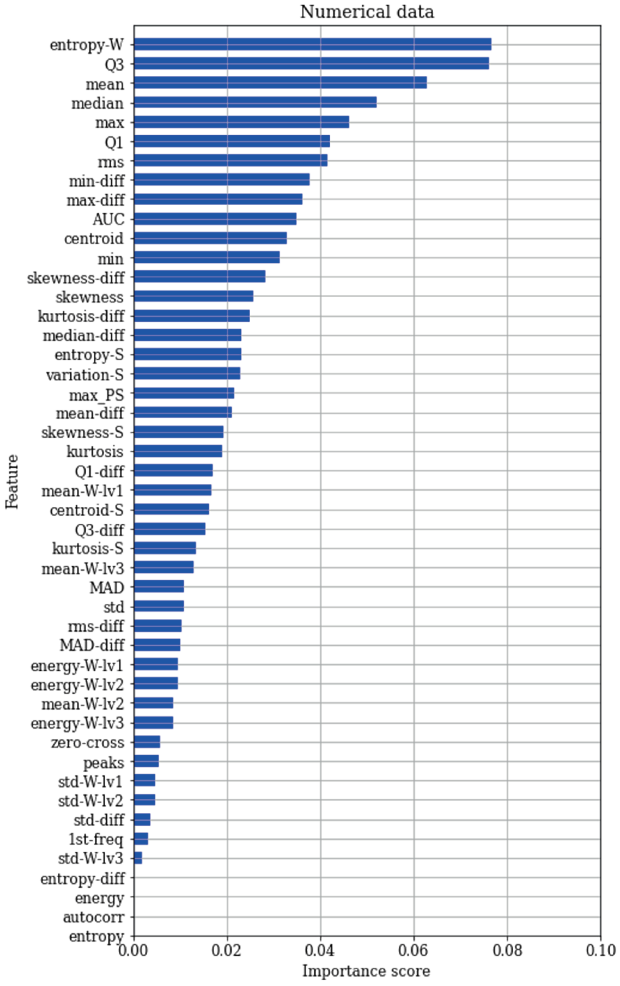


Fig. 10 Feature importance results for the 2D semi-rigid frame structure

4.2 Case study 2: Experimental 3D frame structure

In this example, the proposed feature-fusion approach is applied to an experimental three-story frame structure realized at Los Alamos National Laboratory [32], as illustrated in Fig. 11. The structure has a plan dimension of 0.868 × 0.462 (m), a total height of 1.553 m. The floors are made from aluminum plates with a thickness of 1.3 cm, which are supported by four columns via brackets and bolt connections. The structure was horizontally excited through a shaker which could generate different excitation levels with frequencies in the range [0, 200] Hz. Damages were introduced into the structure by varying the bolt connection tightness at various locations and with different torque values. Details of all damage scenarios are enumerated in Table 4. There are, in total 10 structural states, denoted by D0 to D9, where D0 represents an intact state, and D1 to D9 denote damaged states. To record the structure's responses, 24 accelerometers were installed across the structure body, 8 accelerometers per floor, and two per column-plate joints as depicted in Fig. 10 and enumerated in Table 5. The sampling frequency of the sensors is 1600 Hz.

To construct the adjacency matrix A, encoding the spatial correlation of the sensors, one considers that two sensors are connected if they satisfy one of the following conditions: i) two sensors at the same column-plate connections, ii) sensors on the same plate and iii) two sensors at the two ends of the same column. The resulting adjacency of the investigated structure is highlighted in Fig. 12. For example, considering the sensor 2BC (sensor 10), per condition *i*: $a_{9,10} = 1$: because sensor 2BC and 2BP (sensor 9) at the same location 2B, per condition iii, $a_{2,10} = 1$ and $a_{10,18} = 1$ because sensor 3BC (sensor 2) and 1BC (sensor 18) directly link with sensor 2BC through column 2B. For other sensors which are not connected with sensor 2BC, the associated adjacency matrix components $a_{i,10} = 0$ with $i \neq 2, 9, 18$. Next, one divides the experimental database into three non-overlapping datasets for training, validation, and testing the mFF-SHM model, with a ratio of 70:15:15. Specifically, the size of these three datasets are (1160, 24, 1024), (248, 24, 1024), and (248, 24, 1024), respectively. The first dimension is the number of samples, the second is the number of sensors, and the third stands for the sample length.

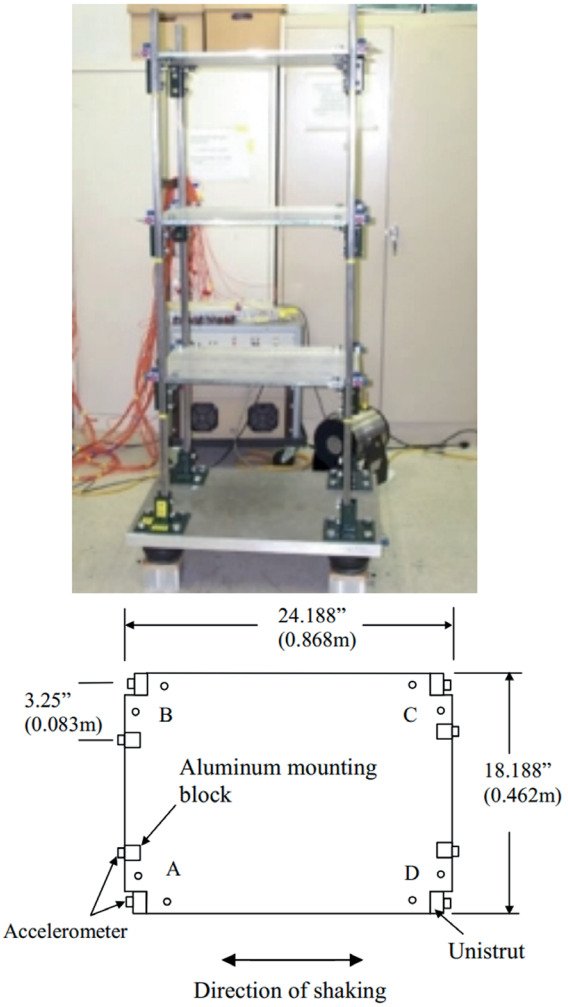


Fig. 11 Image of the experimental steel structure [32]

Table 4 List of damage scenarios

State	Location	Damage description
D0	No damage	
D1	1C and 3A	No bolt between plate and bracket
D2	1C and 3A	Bracket being completely removed
D3	1C	Remaining torque of 5 ft
D4	1C	Remaining torque of 10 ft
D5	1C	No bolt between plate and bracket
D6	1C	Bracket is completely removed
D7	1C	hand tight torque
D8	3A	No bolt between plate and bracket
D9	3A	Bracket being completely removed

For this example, one designs a variant of the mFF-SHM model for mapping input data with one of nine structural states under investigation. A similar working flow, as shown in Fig. 1 and described in the first example, is employed, including data augmentation, feature extraction, self-learning with triplet loss, and an Adaboost-based classifier.

Table 5 List of accelerometer sensors

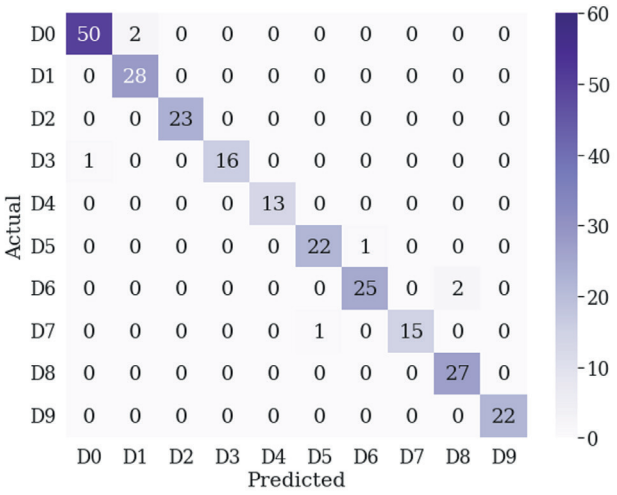
No.	1	2	3	4	5	6
Sensor	3BP	3BC	3AP	3AC	3CP	3CC
No.	7	8	9	10	11	12
Sensor	3DP	3DC	2BP	2BC	2AP	2AC
No.	13	14	15	16	17	18
Sensor	2CP	2CC	2DP	2DC	1BP	1BC
No.	19	20	21	22	23	24
Sensor	1AP	1AC	1CP	1CC	1DP	1BP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
7	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0
10	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0
12	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0
15	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0
18	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

Fig. 12 Adjacency matrix of the sensors installed on the steel frame structure

It is natural to investigate the accuracy of the model in the first place. Because there are up to 9 structural states, thus the problem can be viewed as a multi-output classification; hence, the confusion matrix and F1-score are adopted to assess the model performance. Fig. 13 illustrates the obtained confusion matrix via a heat map where the color intensity emphasizes the number of samples. In the matrix, the diagonal cells indicate samples that are correctly predicted by the proposed model, whereas the off-diagonal cells refer to misclassified ones. It can be seen that most of the samples lie on the diagonal, except for 7 out of 241 samples; this corresponds to an accuracy of 97.1% and an average F1-score of 97%. In addition, F1 scores for every class are also calculated and listed at the bottom of the figure.

Besides, a feature importance study based on the permutation technique for the experimental database is carried out, whose results are reported in Fig. 14. Unlike the first example, 4 out of 5 top features are related to the time derivative of the vibration signals, including, Rms-diff, Q3-diff, min-diff, Q1-diff. Meanwhile, the top features in the first example, such as entropy-W, Q1, Q2, and Q3-quartile values, are



State	D0	D1	D2	D3	D4	
F1_score	0.97	0.96	1.0	0.97	1.0	
State	D5	D6	D7	D8	D9	Average
F1_score	0.96	0.98	0.97	0.96	0.96	0.97

Fig. 13 SHM results for the experimental frame structure obtained by the mFF-SHM method

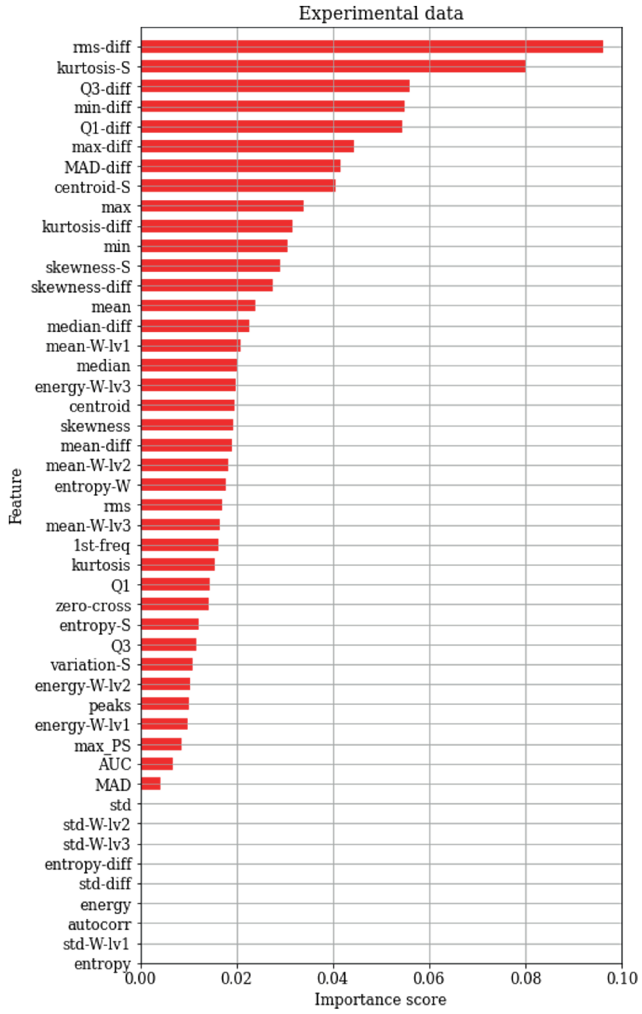


Fig. 14 Feature importance results for the experimental frame structure

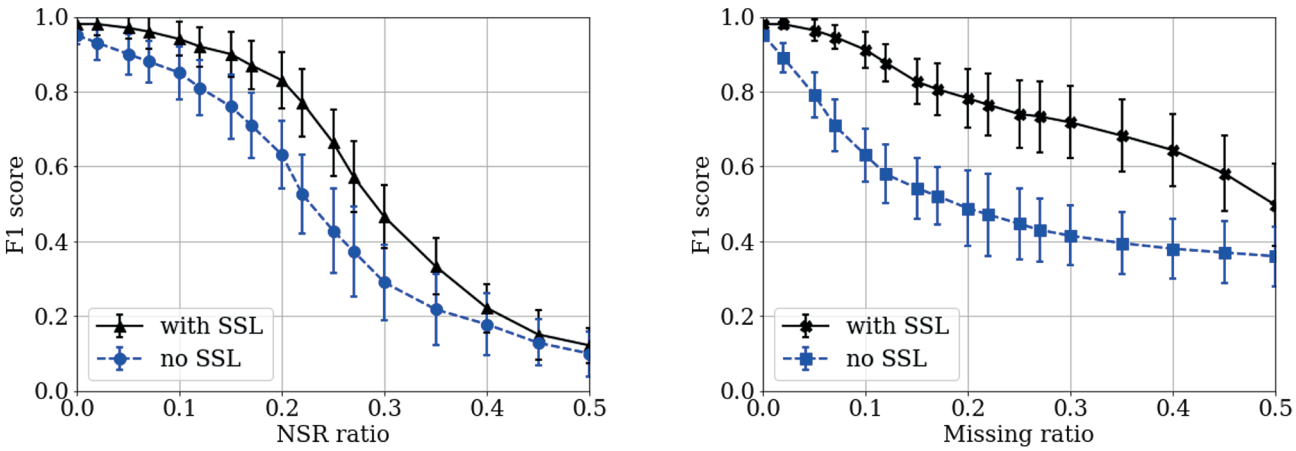
ranked at the bottom half. These results demonstrate that no unique feature can be effectively applied to all SHM problems; however, fusing features in different domains could be useful for different structures.

Next, the mFF-SHM model is compared with other DL/ML-based counterparts in terms of both performance and efficiency. ML-based models directly use extracted features as inputs, whereas mFF-SHM utilizes latent representations of extracted features obtained by self-learning. On the other hand, the DL-based methods take original vibration signals as input. The implementation details of DL models can be found in [33], and those of ML models are set to default values recommended in the machine learning libraries Pycaret and Sci-kit learn. The comparison results are presented in detail in Table 6. It can be seen that the DL-based method can provide relatively good SHM results without using any signal preprocessing; however, their models and computational complexities are significantly higher than feature-based methods, e.g., the training time of the LSTM method is nearly 9 times higher than that of the Decision Tree classifier (36.9 min vs. 4.3 min). The mFF-SHM model has the best balance between the performance and the computational time, as it achieves higher F1-score and accuracy than ML-based methods, with less total CPU time than DL-based methods. It is noted that the total time already includes all time required for feature extraction, self-learning, and training time of the classifier head.

One of the initial motivations for the incorporation of self-learning into SHM applications is to improve the robustness against unfavorable random noise and missing data. To demonstrate this intuition, a robustness study is carried out in which the testing dataset is contaminated by different noise levels. The noise level is characterized by a Noise-to-signal ratio (NSR), being a ratio between the noise amplitude and the root mean square value of original clean signals. On the other aspect, to simulate missing data, a random part of the vibration signal will be zeroed out. The degree of missing data is characterized by the ratio between the number of values being zeroed out with the signal length. After that, the mFF-SHM model and an Adaboost-classifier model without SSL will be tested with contaminated data. The robustness study results are demonstrated in Figs. 15(a) and (b), showing the evolution curves of F1-score against different of NSR and missing ratio values. In the figures, the solid and dashed lines denote results obtained by the models with and without SSL, respectively. To obtain these results, one repeated the

Table 6 Comparison results between the mFF-SHM framework with ML/DL counterparts for the experimental frame structure

No	model	Accuracy (%)	F1 (%)	FE time (min)	Self-learning time (min)	Training time (min)	Total time (min)
1	mFF-SHM	0.98	0.97	2.7	11.6	2.39	16.7
3	1DCCN	0.96	0.94	0	0	23.08	23.1
4	Ada Boost Classifier	0.92	0.91	2.7	0	2.42	5.1
5	LSTM	0.94	0.91	0	0	36.85	36.9
6	Decision Tree Classifier	0.92	0.89	2.7	0	1.56	4.3
7	MLP	0.92	0.89	0	0	23.16	23.2
8	SVM - Linear Kernel	0.81	0.81	2.7	0	1.11	3.8
9	K Neighbors Classifier	0.80	0.78	2.7	0	1.57	4.3
10	Logistic Regression	0.73	0.68	2.7	0	1.96	4.7
11	Naive Bayes	0.72	0.77	2.7	0	1.50	4.2
12	Ridge Classifier	0.65	0.58	2.7	0	1.10	3.8



(a) (b)

Fig. 15 Robustness study results for the experimental frame structure; (a) F1 score vs. NSR ratio, (b) F1 score vs. missing ratio

calculations 100 times and reported the mean and standard deviation values. It can be seen that SSL clearly improves the robustness of the SHM applications, as the gap between the two lines is noticeable. Specifically, SSL helps maintain the F1-score over 90% for NSR and missing ratio ≤ 0.1. The effect of SSL is even more obvious for incomplete data, where, without SSL, the F1-score quickly reduces to about 60% for a missing ratio of 0.1. Besides the discriminative capability of triplet loss, the authors postulate that graph learning is also an important factor for robustness improvement. If a sensor signal is contaminated, adverse effects can be mitigated by information from its neighboring sensors via appropriate aggregation operations.

5 Conclusions

In this study, one has developed a novel multi-modal data-fusion framework for structural health monitoring able to accurately assess the actual structural operational state using only-vibration signals. The main idea of the proposed approach is to extract features from multiple domains, including statistical, temporal, spectral, and spatial domains. Moreover, the interactions between features are also learned via self-supervised learning with a triplet loss function to synthesize latent data representation that better describes the structures' states. After that, the adaptive boosting machine learning algorithm is leveraged to perform SHM tasks using learned data representations.

The credibility and applicability of the mFF-SHM method are consistently demonstrated through two examples with both numerical and experimental databases. The results show that the mFF-SHM method consistently provides SHM results with higher accuracy than various ML/DL-based counterparts while requiring less CPU time than DL models. Moreover, feature extraction reduces the size of a lengthy vibration signal to a 47-length vector, thus significantly alleviating the data storage requirement, which is clearly advantageous for continuous SHM applications working in cloud computing environments.

Moreover, the mFF-SHM framework can still provide reliable monitoring results with noisy data/data with missing values. Specifically, it yields results with more than 90% accuracy for NSR/missing ratio going up to 10%. The feature importance was also performed, pointing out that there is no optimal set of features applicable for all SHM problems, i.e., a feature which is damage-sensitive for a specific structure may be less informative for others; thus, multimodal feature-fusion is an appealing intuition for improving the generality and capability of SHM applications.

In the next step of the study, it is desirable to extend the proposed method to an end-to-end online framework that continuously receives data from measurement devices, extracts and stores meaningful features in cloud storage, and provides SHM results in a near-real-time fashion via an interactive web application. Another exciting research direction is to improve the robustness of mFF-SHM when working with incomplete data by leveraging advanced imputation techniques such as generative adversarial or diffusion models. These models first fill in missing values and then use imputed data to assess corresponding structures' states.

Acknowledgement

This research is supported by Hanoi University of Civil Engineering (HUCE), Vietnam.

References

[1] Gui, G., Pan, H., Lin, Z., Li, Y., Yuan, Z. "Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection", KSCE Journal of Civil Engineering, 21(2), pp. 523–534, 2017. https://doi.org/10.1007/s12205-017-1518-5

[2] Mathew, J., Alfredson, R. J. "The Condition Monitoring of Rolling Element Bearings Using Vibration Analysis", Journal of Vibration, Acoustics, Stress, and Reliability in Design, 106, pp. 447–453, 1984. https://doi.org/10.1115/1.3269216

[3] Yanez-Borjas, J. J., Machorro-Lopez, J. M., Camarena-Martinez, D., Valtierra-Rodriguez, M., Amezquita-Sanchez, J. P., Carrion-Viramontes, F. J., Quintana-Rodriguez, J. A. "A new damage index based on statistical features, PCA, and Mahalanobis distance for detecting and locating cables loss in a cable-stayed bridge", International Journal of Structural Stability and Dynamics, 21(09), 2150127, 2021. https://doi.org/10.1142/S0219455421501273

[4] Hu, C., Albertani, R. "Wind turbine event detection by support vector machine", Wind Energy, 24(7), pp.672–685, 2021. https://doi.org/10.1002/we.2596

[5] Figueiredo, E., Figueiras, J., Park, G., Farrar, C. R., Worden, K. "Influence of the autoregressive model order on damage detection", Computer-Aided Civil and Infrastructure Engineering, 26(3), pp. 225–238, 2011. https://doi.org/10.1111/j.1467-8667.2010.00685.x

[6] Carden, E. P., Brownjohn, J. M. W. "ARMA modelled time-series classification for structural health monitoring of civil infrastructure", Mechanical Systems and Signal Processing, 22(2), pp. 295–314, 2008. https://doi.org/10.1016/j.ymssp.2007.07.003

[7] Zheng, H., Mita, A. "Damage indicator defined as the distance between ARMA models for structural health monitoring", Structural Control and Health Monitoring, 15(7), pp. 992–1005, 2008. https://doi.org/10.1002/stc.235

[8] Pehlivan, H., Aydin, Ö., Gülal, E., Bilgili, E. "Determining the behaviour of high-rise structures with geodetic hybrid sensors", Geomatics, Natural Hazards and Risk, 6(8), pp. 702–717, 2015. https://doi.org/10.1080/19475705.2013.854280

[9] Gordan, M., Ismail, Z., Razak, H. A., Ibrahim, Z. "Vibration-based structural damage identification using data mining", In: 24th International Congress on Sound and Vibration (ICSV24), London, UK, 2017, pp. 6494–6501. ISBN 978-1-5108-4585-5 http://doi.org/10.13140/RG.2.2.35401.03686

[10] Pedram, M., Esfandiari, A., Khedmati, M. R. "Damage detection by a FE model updating method using power spectral density: Numerical and experimental investigation", Journal of Sound and Vibration, 397, pp. 51–76, 2017. https://doi.org/10.1016/j.jsv.2017.02.052

[11] Dang, H. V., Tran-Ngoc, H., Nguyen, T. V., Bui-Tien, T., De Roeck, G., Nguyen, H. X. "Data-driven structural health monitoring using feature fusion and hybrid deep learning", IEEE Transactions on Automation Science and Engineering, 18(4), pp. 2087–2103, 2021. https://doi.org/10.1109/TASE.2020.3034401

[12] Kankanamge, Y., Hu, Y., Shao, X. "Application of wavelet transform in structural health monitoring", Earthquake Engineering and Engineering Vibration, 19(2), pp. 515–532, 2020. https://doi.org/10.1007/s11803-020-0576-8

[13] Kaveh, A., Hosseini, S. M., Akbari, H. "Efficiency of plasma generation optimization for structural damage identification of skeletal structures based on a hybrid cost function", Iranian Journal of Science and Technology, Transactions of Civil Engineering, 45, pp. 2069–2090, 2021. https://doi.org/10.1007/s40996-020-00504-8

[14] Kaveh, A., Mahdavi, V. R. "Damage identification of truss structures using CBO and ECBO algorithms", Asian Journal of Civil Engineering, 17(1), pp. 75–89, 2016. [online] Available at: https://magiran.com/p1437666

[15] Kaveh, A., Zolghadr, A. "An improved CSS for damage detection of truss structures using changes in natural frequencies and mode shapes", Advances in Engineering Software, 80, pp. 93–100, 2015. https://doi.org/10.1016/j.advengsoft.2014.09.010

[16] Kaveh, A., Javadi, S. M., Maniat, M. "Damage assessment via modal data with a mixed particle swarm strategy, ray optimizer, and harmony search", Asian Journal of Civil Engineering, 15(1), pp. 95–106, 2014. https://www.sid.ir/fileserver/je/103820140107

[17] Kaveh, A., Zolghadr, A. "Cyclical parthenogenesis algorithm for guided modal strain energy based structural damage detection", Applied Soft Computing, 57, pp. 250–264, 2017. https://doi.org/10.1016/j.asoc.2017.04.010

[18] Van den Branden, B., Peeters, B., De Roeck, G. "Introduction to MACEC v2. 0: Modal analysis on civil engineering constructions", User Guide and Case Studies, Katholieke Universiteit Leuven, Leuven, Belgium, 1999.

[19] Kaveh, A., Dadras Eslamlou, A. "An efficient two-stage method for optimal sensor placement using graph-theoretical partitioning and evolutionary algorithms", Structural Control and Health Monitoring, 26(4), e2325, 2019. https://doi.org/10.1002/stc.2325

[20] Kaveh, A., Dadras Eslamlou, A., Rahmani, P., Amirsoleimani, P. "Optimal sensor placement in large-scale dome trusses via Q-learning-based water strider algorithm", Structural Control and Health Monitoring, 29(7), e2949, 2022. https://doi.org/10.1002/stc.2949

[21] Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T., Gamboa, H. "TSFEL: Time series feature extraction library", SoftwareX, 11, 100456, 2020. https://doi.org/10.1016/j.softx.2020.100456

[22] Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., Király, F. J. "sktime: A unified interface for machine learning with time series", [preprint] arXiv:1909.07872, 17 Sep. 2019. https://doi.org/10.48550/arXiv.1909.07872

[23] Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A. W. "Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package)", Neurocomputing, 307, pp. 72–77, 2018. https://doi.org/10.1016/j.neucom.2018.03.067

[24] Kipf, T. N., Welling, M. "Semi-supervised classification with graph convolutional networks", In: International Conference on Learning Representations (ICLR 2017), Toulon, France, arXiv:1609.02907v4, 22 Feb. 2017. https://doi.org/10.48550/arXiv.1609.02907

[25] Schroff, F., Kalenichenko, D., Philbin, J. "FaceNet: A unified embedding for face recognition and clustering", In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 815–823. ISBN 978-1-4673-6963-3 https://doi.org/10.1109/CVPR.2015.7298682

[26] Fey, M., Lenssen, J. E. "Fast graph representation learning with PyTorch Geometric", In: International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, arXiv preprint arXiv:1903.02428, 25 Apr. 2019. https://doi.org/10.48550/arXiv.1903.02428

[27] Ali, M., Moreno, P. "PyCaret: An open source, low-code machine learning library in Python", [online] Available at: https://www.pycaret.org

[28] Hastie, T., Rosset, S., Zhu, J., Zou, H. "Multi-class AdaBoost", Statistics and its Interface, 2, p. 349–360, 2009. https://doi.org/10.4310/SII.2009.v2.n3.a8

[29] Kaveh, A., Moez, H. "Minimal cycle bases for analysis of frames with semi-rigid joints", Computers & structures, 86(6), pp. 503–510, 2008. https://doi.org/10.1016/j.compstruc.2007.05.024

[30] Kaveh, A., Shahryari, L. "Eigenfrequencies of symmetric planar frames with semi-rigid joints using weighted graphs", Finite Elements in Analysis and Design, 43(15), pp. 1135–1154, 2007. https://doi.org/10.1016/j.finel.2007.08.001

[31] McKenna, F. "OpenSees: a framework for earthquake engineering simulation", Computing in Science & Engineering, 13(4), pp. 58–66, 2011. https://doi.org/10.1109/MCSE.2011.66

[32] Los Alamos National Laboratory, [online] Available at: https://www.lanl.gov/

[33] Dang, H. V., Raza, M., Nguyen, T. V., Bui-Tien, T., Nguyen, H. X. "Deep learning-based detection of structural damage using time-series data", Structure and Infrastructure Engineering, 17(11), pp. 1474–1493, 2021. https://doi.org/10.1080/15732479.2020.1815225