https://doi.org/10.3311/PPci.22373

Periodica Polytechnica Civil Engineering, 67(3), pp. 875–889, 2023

Development of Structural Damage Detection Method Working with Contaminated Vibration Data via Autoencoder and Gradient Boosting

Viet-Hung Dang[1*]

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

Received: 11 April 2022, Accepted: 03 May 2023, Published online: 09 May 2023

Abstract

Vibration-based structural damage detection is one of the most promising venues for building smart and automated structural health monitoring applications; however, its applicability is impeded by a large amount of collected vibration data, and the performance could be undermined by degraded data. Therefore, this study develops a robust framework, dubbed AutoBoost-SDD, that can effectively handle contaminated vibration data and provide reliable monitoring results within reasonable computational resources. The proposed method consists of three key components. Firstly, multi-domain feature extraction techniques are utilized to convert high-dimensional raw data into informative feature vectors. Secondly, the auto-encoder deep learning architecture is leveraged to refine feature vectors of contaminated data. Finally, a tree-based boosting machine learning algorithm, namely LightGBM, is employed to assess the structures' operational states using learned output from the second step. The viability and performance of the proposed framework are illustrated via three case studies involving numerical data of a 5-degree of freedom system, a 2D frame structure, and experimental data of a large-scale 18-story frame structure from the literature. The results show that the AutoBoost-SDD framework is able to provide reasonable detection results despite the presence of various contaminations, including noisy, missing, and anomalous data.

Keywords

structural health monitoring, vibration, signal processing, deep learning, structural analysis

1 Introduction

An intelligent structural damage detection system [1–3] is an indispensable and critical component of smart structures because it can timely provide an accurate evaluation of the actual state of a structure with reduced manual labor and a low budget. Towards an intelligent structural damage detection (SDD) application, using vibration signal [4] for performing SDD tasks is a convenient way compared to other methods such as manual visual inspection, impact test, etc., since it can be carried out when the structure is in daily service condition and exposed to random operational and environmental loads, without requiring any interruption. A number of well-known works from various authors [5–8] have demonstrated the applicability and credibility of the vibration-based SDD methods. However, in reality, this task is challenging due to unavoidable negative perturbations such as environmental noise, device instability, human errors, transmission loss, etc., leading to noisy and missing measurement data.

A quick and simple technique for handling missing data is to assume a linear relationship between data features, then use available features to derive missing ones, as done in [9]. However, the applicability of such a method for complex structures featuring non-linear responses is limited. When working with incomplete structures' static responses, Kourehli et al. [10] proposed to employ the probabilistic simulated annealing method to evaluate damages in structures. The viability of the method was demonstrated through a simulation example and an experimental 8-degree of freedom (dof)-spring-mass system. Later, Yin et al. [11] developed a practical method based on the Finite element model reduction technique and Bayesian inference, which could provide reasonable structural damage detection results even with incomplete modal data. The proposed method was applied to detect connection stiffness reduction due to bolt loosening in an experimental two-story frame structure. Recently, deep

Cite this article as: Dang, V.-H. "Development of Structural Damage Detection Method Working with Contaminated Vibration Data via Autoencoder and Gradient Boosting", Periodica Polytechnica Civil Engineering, 67(3), pp. 875–889, 2023. https://doi.org/10.3311/PPci.22373

learning-based methods have been increasingly used to impute missing data. A common deep learning architecture is effectively used in structural health monitoring by Dang et al. [12], namely, convolutional neural networks. Fan et al. [13] proposed an innovative approach based on a convolutional neural network for recovering incomplete structural health monitoring (SHM) information with a part of the data lost during the transmission process. The accuracy and robustness of the proposed method were demonstrated via the Dowling Hall Footbridge, showing that recovered signals could provide accurate modal identification results even with a 90% data loss ratio. Jiang et al. [14] utilized the generative adversarial network for directly recovering incomplete sensor data measured from a real bridge without requiring expert knowledge. Furthermore, the method was applicable to both static structural responses and dynamic responses, including those caused by transport vehicles.

In addition to incomplete data, noise is also a negative factor commonly encountered in reality. Ibrahim et al. [15] proposed utilizing a high-pass filter for canceling low-frequency noise of signals measured from low-cost sensors. The authors stated that the higher the cutoff frequency, the more noise cancellation is achieved; however, increasing the cutoff frequency also removes signals' low-frequency contents, which potentially introduces errors. de Castro et al. [16] developed a method combining the Electromechanical impedance technique and the cross-correlation signal processing technique for structural health monitoring in noisy environments. The authors found that the imaginary part of the impedance and wavelet approximation is less sensitive to noise. The method was applied to an experimental plate structure, providing reasonable detection and quantification results of damage under low, moderate, and high noises. To detect minor damages in a structure under a noisy environment, Das and Saha [17] explored a hybrid damage detection algorithm fusing the variational mode decomposition with frequency domain decomposition. The method is applicable to various types of noise, including white noise and random-valued impulse noise. Ma et al. [18] observed that if a sensor's performance degrades, it only affects the corresponding data collected from that sensor but does not alter data from other sensors. Hence, the authors introduced a novel index, namely, the percentage of the extreme value of the largest principal components for, first, separating sensor performance degradation from structural damage, then performing damage detection with high accuracy even in the presence of noise. Recently, Dang et al. [19] demonstrated that using graph learning can effectively reduce the negative effects of missing and noisy data thanks to the ability to leverage the spatial correlation of sensor locations.

1.1 Research significance

In the authors' opinion, there are two drawbacks in reviewed works that could impede their applicability in practice. The first drawback is the difficulty of transmission, analysis, and storage of a large number of raw vibration signals. The second drawback is the need to improve the robustness against several types of perturbations with different degrees. Therefore, this study proposes a novel SDD method, termed AutoBoost-SDD (Autoencoder-Gradient Boosting-Structural Damage Detection), to cope effectively with multiple contaminated signals. The main idea of the method is three folds: first, vibration signals are transformed into a concise yet meaningful feature vector [20] encoding information from statistical, temporal, and spectral domains. Second, a denoising masked auto-encoder network is designed to adjust feature vectors extracted from perturbed signals to new vectors that are close to those from intact signals. Third, a tree-based boosting machine learning algorithm is leveraged to perform SDD tasks using the adjusted feature vector. To sum up, the key contributions of this study are summarized below:

- A robust SDD method able to work with contaminated vibration signals is designed and fully implemented. The proposed method can provide accurate detection results with the presence of various types and different degrees of perturbations.
- The credibility and feasibility of the proposed AutoBoost-SDD are demonstrated through three structural databases with increasing complexity: a 5-degree of freedoms system, a 2D numerical frame structure, and a full-scale experimental 18-story structure from the literature.
- In addition, comparison, robustness, and parametric studies are carried out, providing comprehensive insights into the mechanism of the AutoBoost-SDD method and helping increase its applicability to other problems.

In the remainder of this paper, Section 2 presents the overall working flow and details of the main components of the AutoBoost-SDD framework, including the feature extractor, denoising masked autoencoder and the tree-based

boosting algorithm. In section 3, the proposed method is validated via a number of case studies. Finally, Section 4 draws conclusions and outlines some future perspectives.

2 Self-supervised learning SDD framework using masked denoising auto-encoder

The overall working flow of the proposed framework is graphically illustrated in Fig. 1, which consists of two stages: unsupervised learning and supervised classifier. The first stage is decomposed further into the feature extraction step using signal preprocessing techniques and the reconstruction step using a denoising masked auto-encoder deep learning architecture. Meanwhile, in the second stage, SDD problems are recast into equivalent classification problems, and then, a highly effective and efficient boosting LightGBM is leveraged to perform SDD tasks using reconstructed data obtained from the first stage. The details of each stage of the framework will be described in the following subsections.

2.1 Extracting damage-sensitive features from vibration signals

High-dimensional raw vibration signals contain redundant information leading to unnecessary large storage requirements and additional computation times. Therefore, it is desirable to extract meaningful features that are sensitive to structural states [21–22] before performing subsequent calculation and storage actions. Vibration signals can be classified as time-series data that consist of sequences of values recorded at consecutive time instants. One can compute some statistical characteristics from time-series data, such as max/min values, standard deviation, and skewness. For example, if damage occurs, the structural stiffness will be reduced, and the structure will vibrate stronger; thus, the maximum amplitude will be increased. Therefore, these statistics can be used as damage indicators. In addition, one can include quantile values that describe the shape of the probability distribution of signal values. On the other hand, there are temporal features that describe the evolution over time of vibration values, for example, the number of peaks, zero crossing rate, the area under the curve, and the total energy of the data. The high values of the number of peaks and the zero crossing rate mean that there is significant fluctuation in data, i.e., more contribution from high-frequency components, thus indicating some changes in structural states. A more direct look at the frequency content of signals is to extract features from the spectral domain, such as the fundamental frequency, spectral centroid, Fast Fourier Transform coefficients, etc. It is commonly acknowledged that damages result in shifts in the structure's eigenfrequency values; thus, these spectral features, which can be obtained with the help of the Fourier Transformation, are useful indicators in detecting damage. In short, the features employed

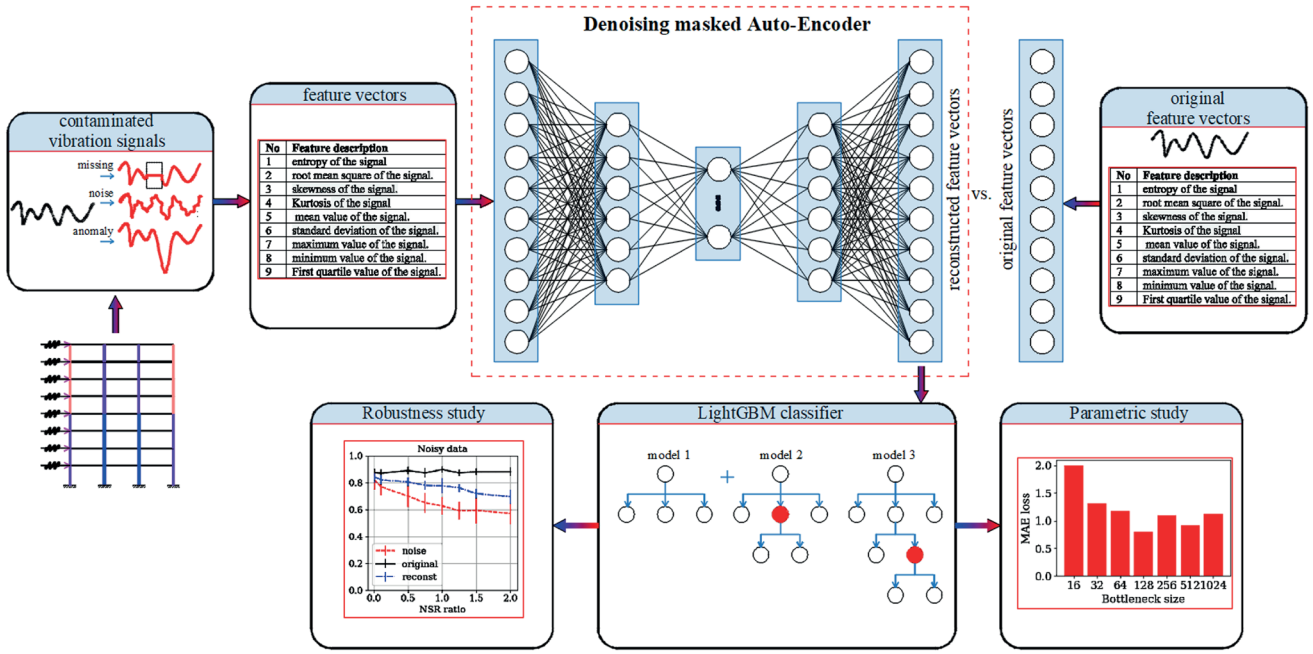

Fig. 1 Architecture of the proposed AutoBoost-SDD framework including a symmetric denoising masker Auto-Encoder and the Boosting LightGBM model

in this study are enumerated in Table 1 with the help of the TFEL library [23]. It is possible to include more features in the data to improve the detection accuracy; on the other hand, a feature importance study could be carried out to assess the contribution of every feature, and irrelevant ones could be discarded accordingly. A more in-depth study about feature extraction could be found in [24].

2.2 Contaminated vibration signals

In practice, there are numerous unfavorable factors that could contaminate the signal quality, such as transmission loss, human errors, device instability, and environmental noise. One of the most common perturbations is the presence of noisy signals, which can be modeled by adding an amount of white noise to the original signals. The noise level is characterized by a Noise-to-Signal ratio (NSR), which is the ratio between the noise amplitude and the root mean square value of the original vibration signals. Another popular perturbation is missing data, where some parts are not available, resulting in NaN (not a number) symbols in the database. Such a problem will lead to errors in calculations and stop the detection application since most equations and programming functions only take numerical values as inputs. In order to address this problem, imputation techniques need to be applied. Examples of these techniques are zero imputation, mean imputation, neighboring imputation, etc. After imputing, traditional detection algorithms can be performed, but the signal contents are considerably altered, leading to inaccurate detection results. The third type of perturbation widely considered in assessing the method robustness refers to the anomaly problem where certain signal values differ significantly from others (either larger or smaller) or where a portion of signals exhibits abnormal patterns. Such a phenomenon could modify the statistic characteristics, the temporal shape, and the frequency contents of signals, leading to detection errors and misinterpretation. There are also other types of perturbations that have been addressed in the literature [25], such as scaling problems, artificial trends, magnitude warping, time warping, and so forth. Representative examples of contaminated signals are illustrated in the leftmost panel of Fig. 1, as well as later in Section 3.4. In short, it is desirable to design a robust detection framework that could provide acceptable detection results even with signals contaminated by unknown perturbation.

Table 1 List of features extracted from vibration signals

No	Feature description	Function
1	Signal entropy	tfea.entropy
2	Maximum value of the signal.	np.max
3	Minimum value of the signal.	np.min
4	Mean value of the signal.	np.mean
5	Root mean square of the signal.	tfea.rms
6	Standard deviation of the signal.	np.std
7	Skewness of the signal.	tfea.skewness
8	Kurtosis of the signal	tfea.kurtosis
9	1st quartile value of the signal.	np.quantile(0.25)
10	Median value of the signal.	np.quantile(0.5)
11	3rd quartile value of the signal.	np.quantile(0.75)
12	Area under the curve of the signal	tfea.auc
13	Centroid along the time axis	tfea.calc_centroid
14	Total energy of the signal	tfea.total_energy
15	Zero-crossing rate of the signal	tfea.zero_cross
16	fundamental frequency of the signal.	tfea.fundamental_ frequency
17	Centroid of the signal spectrum.	tfea.spectral_centroid
18	Variation of the signal spectrum	tfea.spectral_variation
19	Skewness of the signal spectrum	tfea.spectral_skewness
20	Kurtosis of the signal spectrum	tfea.spectral_kurtosis

(*)tfea: tsfel.feature_extraction.features

2.3 Denoising masked auto-encoder

Among various neural networks that have been successfully applied in structural engineering [26–30], an auto-encoder is a specialized deep-learning architecture designed to learn a compact representation of data that encodes the most meaningful information [31]. The authors postulate that the learned compact data representation of an auto-encoder architecture will filter out noise, anomalies, redundant information, and other spurious artifacts. Furthermore, it is expected that the reconstructed feature vector resembles that of a vibration signal in good condition. It is worth noting that the auto-encoder does not require labeling data with associated structural states. Specifically, a vibration signal is polluted with perturbations described earlier; then, feature vectors are extracted before entering the auto-encoder. At the output layer, the reconstructed feature vector will be compared with that distilled from the original data. By doing so, the auto-encoder becomes practical and can be trained with a large volume of unlabeled vibration data. A typical auto-encoder network is composed of three parts: an encoder, a bottleneck, and a decoder. The encoder gradually compresses input data into latent representations with significantly lower dimensionality compared to the input data. This can be done by using multiple layers of

perceptron where the size of the subsequent layers progressively decreases. The bottleneck is a neural layer that has the smallest size in the network and represents the compressed data representation. Although it is possible to store and utilize learned representation at this step for performing damage detection, this representation does not have any physical or statistical significance, making it a complete black box to the users. Therefore, a decoder is employed to reconstruct vibration features back to their original form. Usually, the architecture of the decoder is symmetric to that of the encoder, consisting of multiple layers where the size of the layers gradually increases from one layer to the next. Different from the traditional autoencoder, in this study, one does not directly feed signals into the encoder. Instead, one extracts vibration features at first and compares reconstructed feature vectors with those of the original signal.

The similarity between reconstructed feature vectors and original ones is measured via the mean absolute error (MAE). Note that it is necessary to apply the standardization to avoid the scaling problem, especially when computing the MAE loss. Otherwise, features of large values will dominate other features with small values regardless of their sensitivity to damage. Afterward, the autoencoder is trained by using the backpropagation algorithm [32]. The gradient of the loss function with respect to the model's weights is computed and used to simultaneously update the weights of the encoder and decoder in the direction that minimizes the loss function. The essential hyperparameters of the autoencoder are the size of the bottleneck layer, the number of layers in the Encoder/Decoder, the activation functions applied to the output of each layer, the learning rate, and batch size. The appropriate values for these hyperparameters are problem and data-specific; they will be determined through a hyperparameter tuning process with the aid of specialized techniques such as grid search, random search, or Bayesian optimization.

The Auto-encoder is implemented from scratch by the authors with the help of the machine learning library Pytorch [33] and trained on a computation server equipped with an Intel Xeon CPU E5-2650, Nvidia 3080 GPU and 64 Gb RAM.

2.4 Tree-based boosting machine learning algorithm LightGBM

Decision trees. The decision tree model refers to a hierarchical binary splitting model that divides data into two groups based on data features in a top-down fashion [34]. Each splitting is performed such that the homogeneity of samples within groups is maximized. The procedure is iterated until one of the stopping criteria is satisfied, e.g., reaching the maximum number of iterations or the minimum number of samples in a group. After that, the predicted output of each final group, a.k.a leaves, is obtained by averaging the values of its data samples. When a new sample enters the trained Decision Tree model, this sample will be classified into one of the final leaves, and the predicted value is the output value of the corresponding leaf.

LightGBM. Although the decision tree is fast and practical, its performance is not stable; in other words, it is very sensitive to many parameters, e.g., the data size, the tree depth, the used criteria, etc. Based on the decision-tree algorithm, Ke et al. [35] developed a distributed gradient-boosting framework that can handle large datasets with reduced computation time and does not require as much memory. LightGBM consists of a sequence of decision tree models, which are successively trained such that the later tree learns to correct the mistakes of the previous trees. At each iteration, the tree models are grown in a leaf-wise fashion, i.e., the new tree is obtained by including a new leaf to its predecessor. Note that the added leaf usually corresponds to the most important features that could significantly reduce the errors. The final predictions are calculated by summing the predictions of all the individual tree models. A graphical explanation of LightGBM is illustrated in Fig. 2. It can be seen that starting with model 1, one will create a model 2 based on a critical feature (in red), and the results at iteration 2 are obtained by aggregating outputs of both models 1 and 2. Next, at iteration 3, model 3 is devised to improve the performance of model 2, and the results at iteration 3 are calculated based on the outputs of all three models. The above procedure is

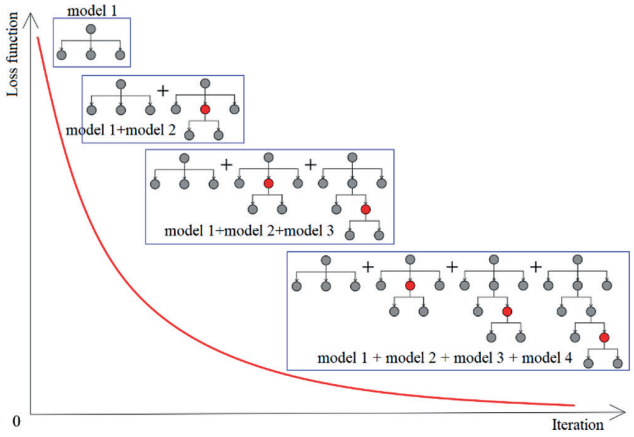

Fig. 2 Graphical representation of the LightGBM algorithm

repeated until one of the convergence criteria is reached. On the other aspect, some main hyperparameters of LightGBM are the number of decision trees to include in the ensemble, the maximum depth of each tree, the maximum number of leaves in a tree, etc.

3 Application examples

3.1 Problem descriptions

In this section, the proposed AutoBoost-SDD model is applied to three case studies with increasing complexity. The first case study is a 5-degree-of-freedom system modeling a 5-story structure with lumped masses subjected to a time-varying horizontal load at the top story, as displayed in Fig. 3(a). Damage is artificially introduced into the structure by randomly reducing one of five story stiffnesses by an amount in the range of [90%, 60%, 30%, 10%]. It is desirable to detect whether the damage exists or not and localize the story where it occurs based on vibration signals recorded at lumped masses. The system is numerically modeled by using the open-source Finite element software OpenSees by the Pacific Earthquake Engineering Research Center [36]. In the model, the discrete masses are connected via zero-length elements with stiffness and viscous damping values described above. The structural responses are obtained by dynamic analysis using the Newmark integration scheme and Newton-Raphson method. The sampling frequency is 100 Hz, and the time duration of each simulation is 30 s. To create the structural database of this 5-dof system, 5000 simulations were carried out with random variables including the mass, the story stiffness, the damping coefficient, the loading amplitude, following normal distributions with mean values of m = 100 kg, k = 30 kN/m and c = 200 Ns/m, respectively. The database of the first example has a shape of [5000, 5, 3000], with 5000 being the total number of samples, 5 being the number of sensors, and 3000 being the signal length.

The second example involves a three-span eight-story steel frame subjected to time-varying horizontal loads acting at story levels, as shown in Fig. 3(b). The typical story height is 4.0 m, resulting in a total height of 32 m; the spans have the same width of 6 m. The cross-sections of the structural members are W12X87, W27X129, W24X62, and W24X55, as indicated in the figure. The structure's dynamic responses to time-varying excitations are obtained by using dynamic analysis, similar to the first example. However, the CPU time of each simulation is significantly longer than that of the 5-dof system due to the increased complexity and number of degrees of freedom, requiring a smaller time step (0.001 s) for convergence. The damage is introduced to the structure by randomly decreasing the moment of inertia of one or multiple columns (up to 5 columns) by an amount of [10%, 30%, 60%, 90%]. The purpose of the example is to use vibration signals recorded from beam-column connections to inversely detect the presence of damage. There are, in total, 5000 simulations with different damage locations, time-varying loads, and different structural parameters, including Young's modulus, structural damping ratio, and time-varying loads drawn from the distributions enumerated in Table 2. The database of the second example has a shape of [5000, 32, 1000], with 5000 being the total number of samples, 32 being the number of sensors, and 1000 being the signal length.

The third example is a high-rise steel frame structure subjected to ground motions that simulate earthquake loads, carried out at the Hyogo Earthquake Engineering

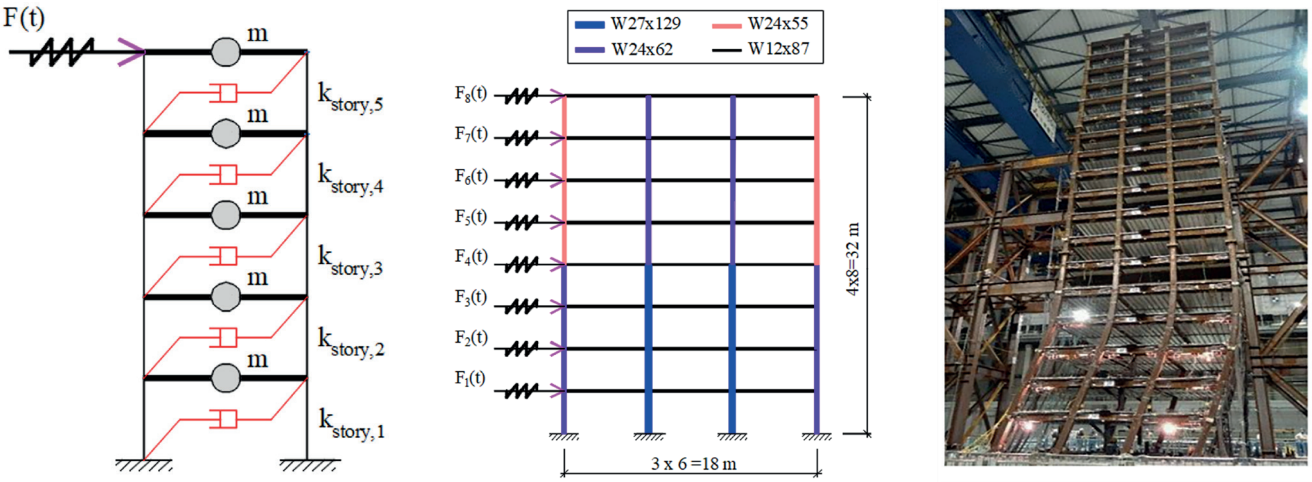

Fig. 3 Images of three case-studies including two numerical and one experimental structures

Table 2 Random variables of the numerical 2D steel structure

Variable	Mean	CoV	Distribution
E (GPa)	220	0.01	Normal
ζ (%)	5	0.05	Normal
F_1 (N)	1000	0.1	Normal
F_2 (N)	1000	0.1	Normal
F_3 (N)	1000	0.1	Normal
F_4 (N)	1000	0.1	Normal
F_5 (N)	1000	0.1	Normal
F_6 (N)	1000	0.1	Normal
F_7 (N)	1000	0.1	Normal
F_8 (N)	1000	0.1	Normal

Table 3 Damage scenarios of the experimental structure

State	pSv (cm/s)	Description
1	40	Elasticity
2	81	Partial plasticity
3	110	Beam plasticity, yielding occurred at column bases Cracking
4	180	Beam plasticity, Cracking
5	220	Beam fracture
6	250	Further beam fracture
7	300	Further Beam fractures
8	340	Local buckling
9	420	Observation of collapse

Research Center [37]. The frame has 18 stories with a total height of 25.35 m, three equal spans of 2 m width in the long horizontal direction, and one 5 m span in the other direction. An image of the structure is shown in Fig. 3(c). The columns are made from built-up hollow sections, while the beams are I-shaped and welded to the columns. The total weight of the structure is approximately 4200 kN. The structure was subjected to ground motions having characteristics of earthquake waves recorded at the Tokyo Shiba Elementary school by MeSOnet in 2011. Furthermore, nine levels of amplitudes were used, corresponding to pseudo spectral velocities (PSV) lying in the range of [40, 81, 110, 180, 220, 250, 300, 340, 420] cm/s. These excitations caused various damages to the structure, such as yielding at beam ends, fracture, local buckling of columns, global buckling at lower stories, and, eventually, collapse mechanism (Table 3). The floors' accelerations in the excitation direction were measured by accelerometer sensors. Furthermore, each vibration signal was divided into multiple 500-length sub-time-series, then combined with signals from different floors to form a multivariate time-series database. To be specific, the shape of the input data is (684, 18, 500), where 684 is the number of samples of the input database, 18 is the number of sensors, and 500 is the signal length.

Of note, the datasets of all three examples are then split into two non-overlapping training and validation datasets with a ratio of 80:20.

3.2 Self-supervised learning process

Next, self-learning is performed to train the auto-encoder component such that it can reconstruct sensitive-damage features from contaminated vibration signals. The datasets are polluted by adding random noise, arbitrarily introducing anomaly values, and replacing original values with NaN for simulating missing scenarios. The contaminated data's underlying features will be extracted and passed through the Auto-encoder for reconstruction; next, the output will be compared with the original data's features using the MAE loss. It is noted that NaN values are filled with the average of other signal values. During this stage, 50% of datasets will be used to train, and the remaining 50% will be hidden from the Auto-Encoder to validate its reconstruction ability. The learning curves of self-learning for the 5-dof systems, 2D numerical frame, and 3D experimental structure are highlighted in Fig. 4(a). It can be seen that, initially, MAEs are very high (more than 10^3), then the loss functions decrease before plateauing around low values. Here, the exponential decay schedule is applied to adaptively adjust the learning rate (lr) during the training process. In the beginning, the learning rate is set to lr = 0.005; then, it is decayed by a ratio of 0.9 after every 50 epochs without improvement. From the figure, it is reasonable to stop the self-learning process at epoch 500 as no clear improvement is observed.

3.3 Supervised learning process

In terms of the model configuration, it is acknowledged that there is no one-size-fits-all optimal configuration. Herein, the authors performed a parametric study and selected a configuration that achieves a good balance between detection accuracy, time complexity, and computation resources. Specifically, the architecture details of the AutoBoost-SDD framework in this study are enumerated in Table 4. With the datasets and detection model in place, one will carry out several studies to assess the performance, robustness, and efficiency of the model as follows.

At first, the detection results with clean and complete data are presented, which serve as reference results when assessing the negative effect of unwanted factors such as

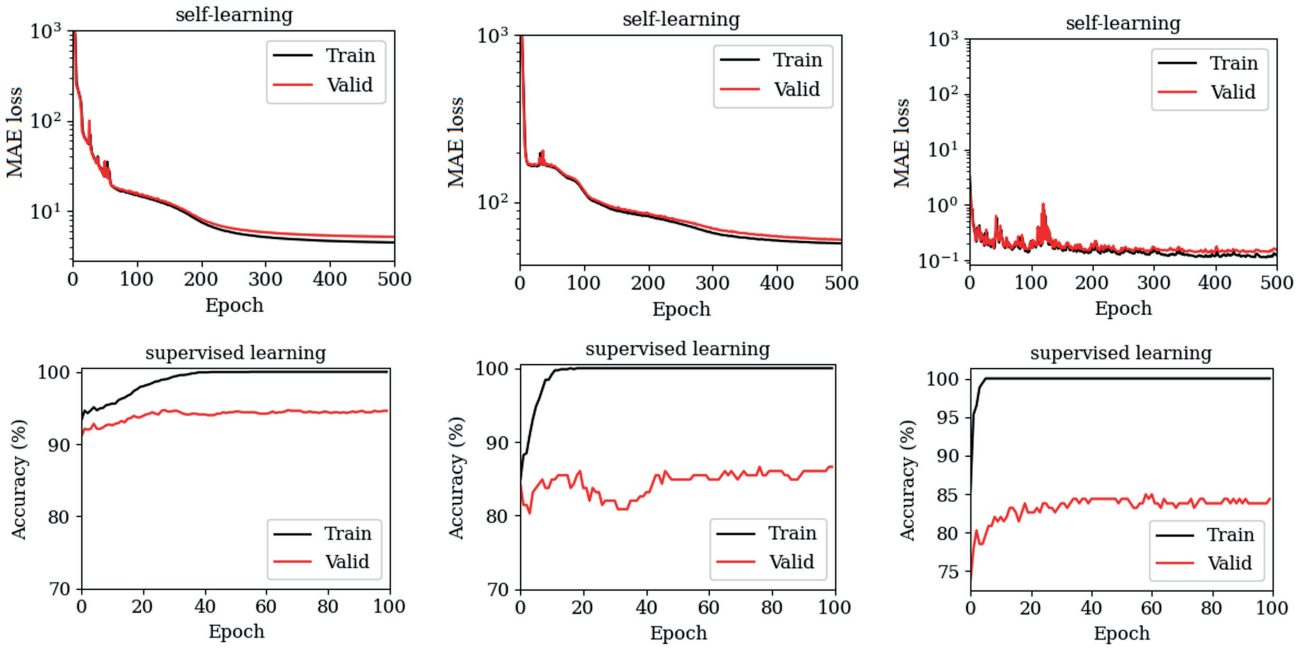

Fig. 4 Learning curves of the Auto-Encoder network

Table 4 Configuration details of AutoBoost-SDD

Model details	Value	Model details	Value
Learning rate	Exponential Decay schedule	N_layer of Encoder/ decoder	3
Loss function	MAE	Bottleneck size	64
Tree depth	15/15/8	Activation	ReLu
N_leaves	50/125/9	Batch size	256
N_tree	200/1000/90	Optimizer	Adam

noise, missing data, and anomalies. More specifically, the features extracted from clean data are fed directly into the LightGBM algorithm to inversely identify the corresponding structural states. The evolutions of the LightGBM model's accuracy during training iterations on both the training and validation datasets are highlighted in the second rows of Fig. 4. Apparently, the feature-based model can quickly reach convergence with less than 100 training iterations, and the training times are in the order of a few minutes. Detailed values of CPU times will be presented later in the comparison study.

On the other hand, these SDD results for three case studies under investigation are graphically demonstrated via confusion matrices in Fig. 5. The vertical axis of a confusion matrix refers to the actual structural states of data samples, whereas the horizontal axis denotes those predicted by the AutoBoost-SDD method. In the first and second examples, the model classifies structural states into two categories: intact and damaged, while in the third example, there are nine states. Based on the confusion matrices, it is possible to derive other statistical metrics to measure the model performance, including accuracy, false negative and false positive errors, and f1-score. Accuracy

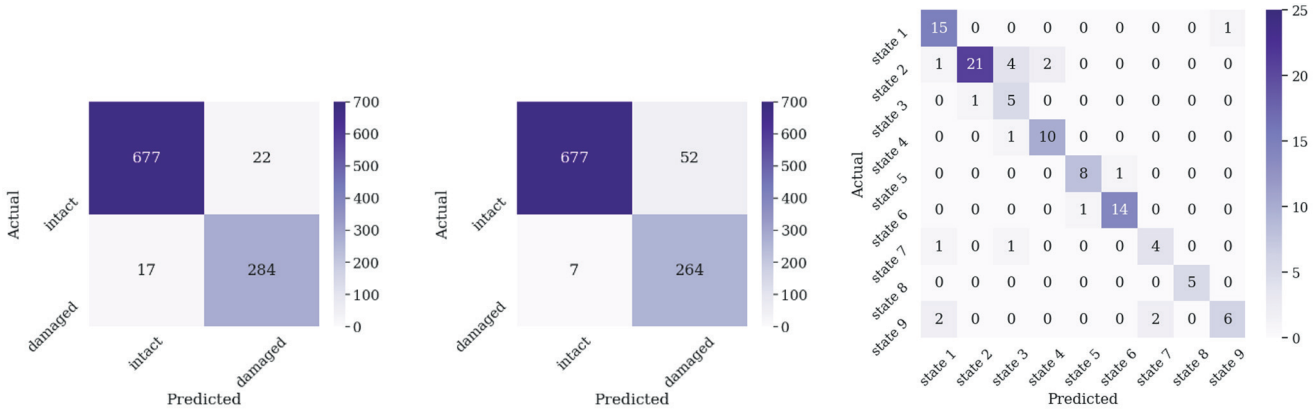

Fig. 5 Damage detection results via confusion matrices

is obtained as the ratio between the sum of diagonal cells and the total number of samples. F1-score is a metric that can be applied to both balanced and imbalanced datasets. From the figure, the accuracies of AutoBoost-SDD for three examples are 96.1%, 87.5%, and 83.0%, and the f1-score are 93.5%, 85.2%, and 81.6%, respectively. The first example has the highest SDD results, followed by the second example. Meanwhile, for the experimental dataset with significantly more structural states, the f1-score is lower than those of numerical datasets. Note that it is possible to improve the detection accuracy with a more complex model; however, such a model is more susceptible to contaminated data and overfitting problems if it relies only on supervised learning.

In this study, the LightGBM algorithm is selected as the classifier based on the results of a comparison study that evaluated nine machine learning algorithms, including XGBoost [38], CatBoost [39], Random Forest, AdaBoost, Gradient Boosting, Decision Tree, Logistic Regression, Support Vector Machine (SVM) in addition to the standard artificial neural network. Logistic regression is one of the most simple and fastest classification models that can quickly provide baseline results based on which other algorithms can improve results further. SVM is a well-established method that was widely used before the deep learning era. Decision Tree is one of the few ML models that are considered interpretable rather than a black box like the others. XGBoost, LightGBM, CatBoost, Random Forest, AdaBoost, and Gradient Boosting are tree-based algorithms that have been devised to improve the models' performance, robustness, and speed by leveraging ensemble theory and parallel computation on available hardware. The comparison results are enumerated in Table 5, including the f1 score, training time, and inference time. Overall, the four algorithm tree-based ensemble models XGBoost, LightGBM, CatBoost, and Random Forest can provide highly similar f1 scores, e.g., 0.94, 0.84, and 0.81 for the first, second, and third structures, respectively. These results are considerably higher than other methods, including the over-parameterized ANN, simple regression model, and single decision tree. However, the CatBoost model requires the longest training time and inference time which can be up to one order of magnitude longer than the others when working with multiple sensors at the same time. For instance, its training time for the third example reaches 60800 ms compared to 351 ms required by Random Forest. Overall, the LightGBM model consistently achieves the best balance between the performance and time complexity, especially at inference time; it is the fastest among tree-based ensemble models. In order to ensure a relatively fair comparison, one applied the same key parameters for tree-based methods, such as the number of trees equal to 200, the number of leaves of 50, and the maximum depth level of 15. In short, these results provide specific and solid evidence for the adoption of LightGBM in this study.

3.4 Parametric study

As presented in the previous section, the AutoBoost-SDD framework has some key hyperparameters that could have a significant impact on the final results; hence, a parametric study was conducted to estimate their contribution. One divides hyperparameters into two groups, one for self-learning and the other for supervised learning. The first group consists of the activation function, the size of the bottleneck layer, and the number of Encoder/Decoder layers.

Table 5 Comparison results between different ML/DL-based classifiers

Model	Ex1			Ex2			Ex3		
	F1-score	Training time (ms)	Inference time (ms)	F1-score	Training time (ms)	Inference time (ms)	F1-score	Training time (ms)	Inference time (ms)
XGBoost	0.94	921	6.9	0.84	11100	22.9	0.81	1030	6.0
LightGBM	0.94	259	5.0	0.84	2990	9.0	0.81	1390	3.0
CatBoost	0.94	9500	25.9	0.83	5800	240	0.81	60800	21
RandomForest	0.94	1740	19.9	0.83	8610	38.7	0.80	351	12
AdaBoost	0.86	1770	21	0.72	19600	170	0.48	961	12
GradientBoosting	0.90	7970	3.93	0.80	95000	9.0	0.74	29500	3.0
DecisionTree	0.90	393	2	0.81	3760	6.01	0.78	90	2.0
LogisticRegressor	0.87	44	0.88	0.78	6890	4.0	0.56	123	4.2
SVM	0.86	378	157	0.74	12700	3710	0.69	31.9	16
ANN-(128/64)	0.87	4800	137	0.75	24100	172	0.80	1380	104

Recall that the Encoder and Decoder of the Auto-Encoder model are symmetric, i.e., they have the same number of neural layers. For the activation function, five different functions are considered, which are relu, sigmoid, tanh, softmax, elu. These activation functions are commonly used in deep learning models. Their corresponding MAE losses are plotted in bar charts shown on the leftmost of Fig. 6. For all three case studies, the relu and elu functions provide considerably lower MAE loss values than the others. However, the formula of the relu function is more straightforward, and does not introduce additional parameters like the elu function; hence, the relu function is selected. For the bottleneck size, associated MAE results are displayed in subfigures in the middle columns of Fig. 6. For the 5-dof systems, the 128-neuron layer yields the lowest MAE loss, while the smallest 16-neuron layer results in the highest MAE loss. For the 2D numerical frame, except the 16-neuron layer, other bottleneck layers provide approximately the same MAE loss values. For the experimental structure, the lowest MAE loss between reconstructed feature vectors and with original ones was obtained by setting the bottleneck size to 64 or 1024. Based on these results, one sets the bottleneck size to 64. Similarly, it was observed that setting the number of Encoder/Decoder layers to 3 provided good MAE loss values for different problems and maintained a reasonable model complexity.

For supervised learning with the lightLGBM model, typical key hyperparameters of tree-based models are the number of trees (n_estimators), the maximum number of leaves per tree (n_leaves), and the tree depth (max_depth) [35]. The corresponding parametric results for all three examples of interest are demonstrated in Fig. 7. We found that it could not apply the same set of hyperparameters to different problems because their effects vary significantly across problems. For example, setting a max_depth between [15, 20]

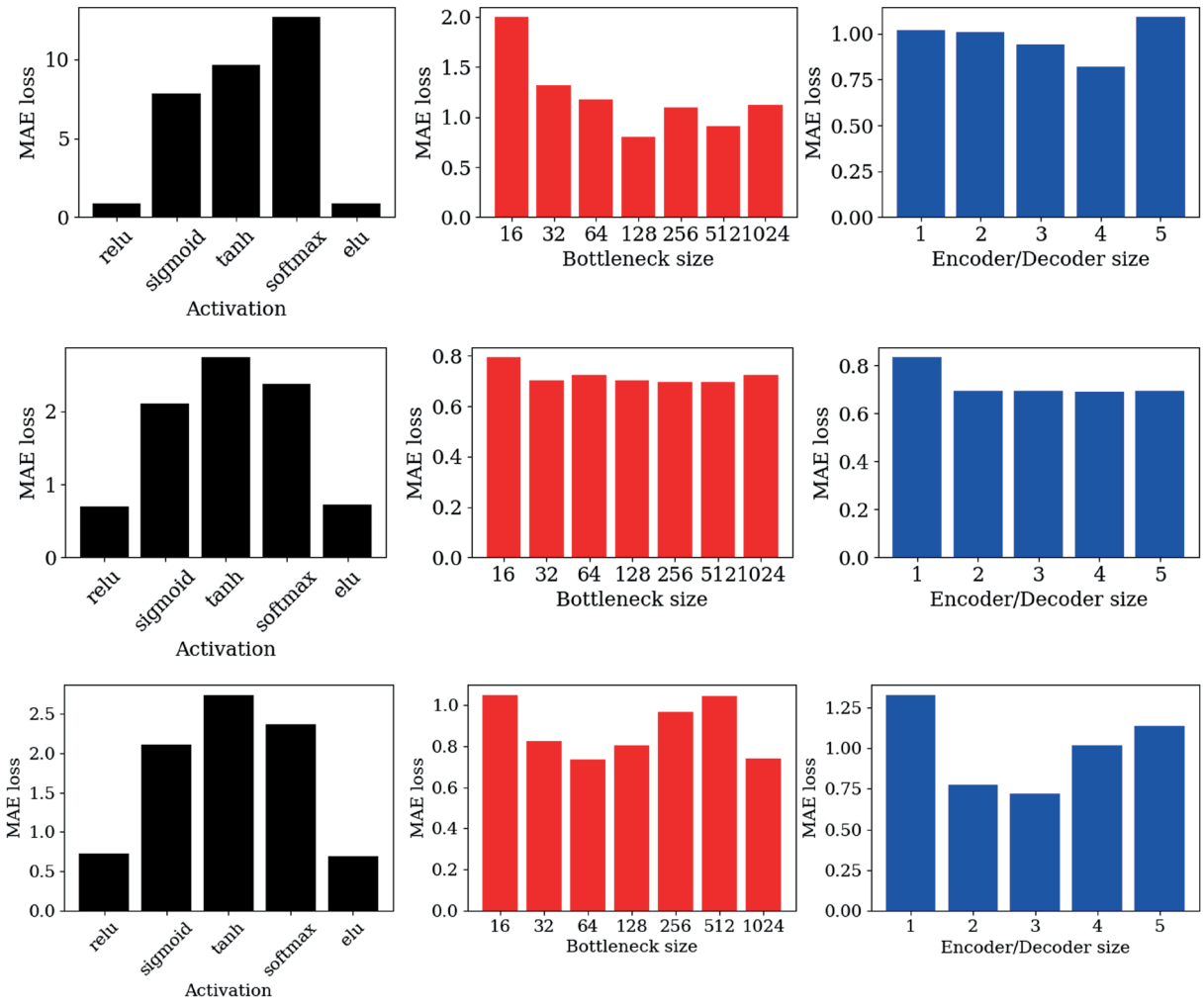

Fig. 6 Parametric results showing the effects of activation functions, bottleneck size and the number of layers in the Encoder/Decoder on the self-learning Auto-Encoder model

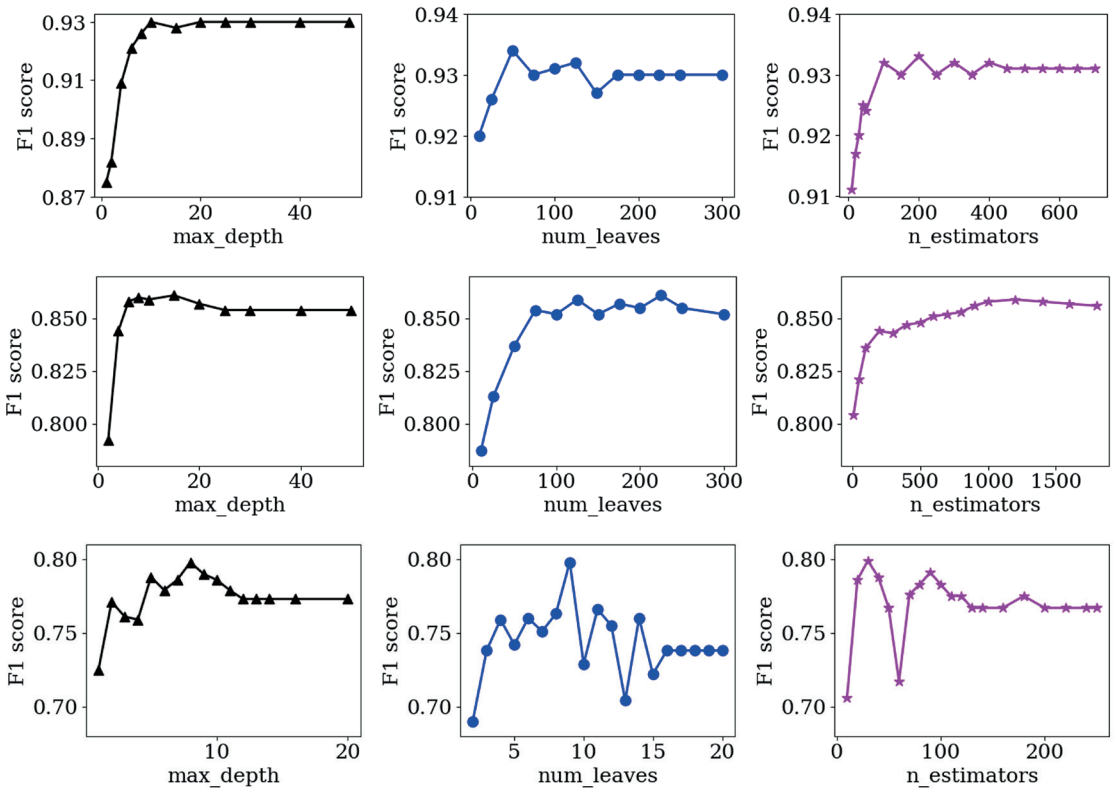

Fig. 7 Parametric results showing the effects of the tree depth, number of leaves and the number of trees on the LightGBM classifier performance

provide high F1 scores for the two first examples but leads to markedly low scores for the third example. On the other hand, using a n_estimator of around 100 results in a relatively high F1-score of 0.8, but the recommended values for the first and second examples are around 200 and more than 1000, respectively. The same observation about the fluctuation of the hyperparameter effect also holds for n_leaves. Hence, the adopted sets (max_depth, num_leaves, n_estimators) for the 5-dof system, 2D numerical frame, and 3D experimental structure are (15, 50, 200); (15, 125, 1000) and (8, 9, 90), respectively.

3.5 Robustness study

Finally, the designed AutoBoost-SDD method is tested with various vibration signals contaminated with different types and degrees of perturbations. Fig. 8 shows an example of a vibration signal and its contaminated variants, along with the extracted and reconstructed features using the AutoBoost-SDD framework. Qualitatively, extracted features from the contaminated signals are different from those from the original vibration signal, whereas reconstructed features resemble the original ones to some extent. Quantitatively, the robustness study results are demonstrated in Fig. 9, where the subfigures in the first, second, and third rows correspond to the 5 dof system, 2D frame structure, and 3D experimental structure, respectively. The results for missing data are placed on the left, noisy data in the middle, and anomalous data on the right. Since perturbations are introduced randomly, calculations for each perturbation case are repeated ten times, and the mean and standard deviation values are reportedIn each subfigure, there are three curves where the solid black curve corresponds to SDD results with clean data, the dash-dotted line refers to the SDD results with reconstructed data, and the dashed line presents results for perturbed data. For missing data, the missing rate ranges from 1% to 15%; for noisy data, the NSR ratio is from 0.01 to 2.0; and for anomalous data, the number of anomaly points is within the range [1, 100]. Clearly, the greater the level of perturbation in the signals, the lower the f1 scores and the poorer the detection performance. However, the effects of different types of perturbations are not equal. Noisy and anomalous data cause more degradation in the model performance compared to missing data. The reduction in f1 score is approximately proportional to the noise ratio, whereas only a few anomaly points could cause a considerable decrease in the score. On the other hand, the reconstructed data clearly improve the SDD results, as the corresponding curves lie close to those of the original data. For example, in the case of perturbed data with

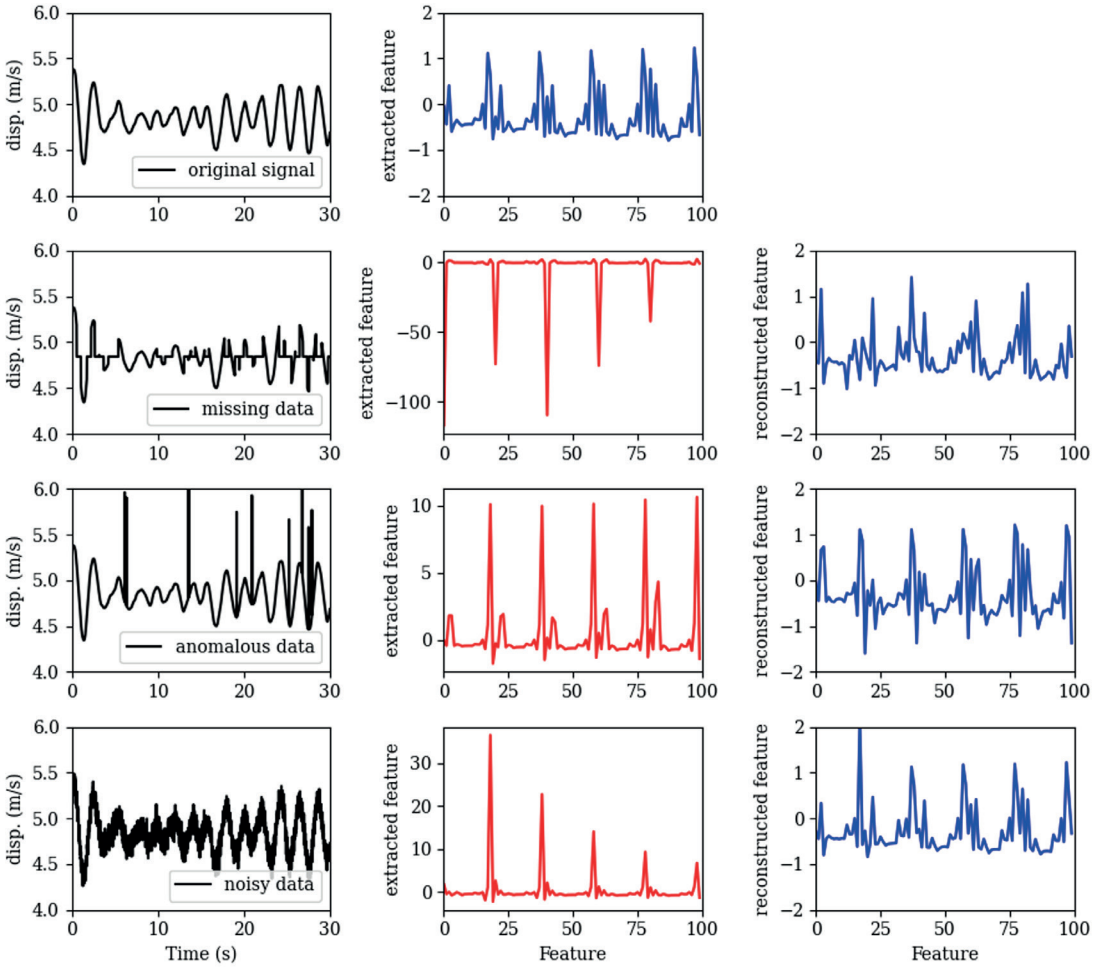

Fig. 8 Representative examples of a vibration signal and its contaminated variants (leftmost), extracted features (middle) and reconstructed feature (rightmost)

100 anomaly points, the f1-score reduces from around 80% to 47%, but using reconstructed data, the f1-score remains around 64%. It is also observed that the vertical bars denoting the standard deviation of reconstructed data are shorter than those of perturbed data, which means results obtained with reconstructed data are more reliable. These results qualitatively and quantitatively reaffirm the robustness of the proposed AutoBoost-SDD framework against vibration signals polluted by various unfavorable issues with different contamination levels.

4 Conclusions

The present study develops a robust two-stage method for structural damage detection that can provide reliable detection results even with contaminated data and reduced computational resources. Over the course of the paper, ones described the main idea, the overall working flow, and the details of main components such as the vibration feature extractor, unsupervised auto-encoder, and supervised LightGBM have been described. Furthermore, specific values of hyperparameters, implementation details, and the credibility of the proposed method are demonstrated throughout three case studies, including numerical data, and experimental data. The obtained results reaffirm that the AutoBoost-SDD significantly improves the SDD performance in the presence of various types of perturbation, i.e., it achieves higher detection accuracy compared to supervised counterparts by a clear margin. Furthermore, one investigates the AutoBoost-SDD framework from different perspectives through different studies, including comparison, parametric and robustness studies, to gain more understanding of its underlying mechanism and the impact of each hyperparameter on the model's performance. Such understanding is useful for fine-tuning the framework to achieve better performance and facilitate its adaptability to tackle new problems with varying complexities.

For the next step of the study, one will include a probabilistic component into the AutoBoost-SDD framework so

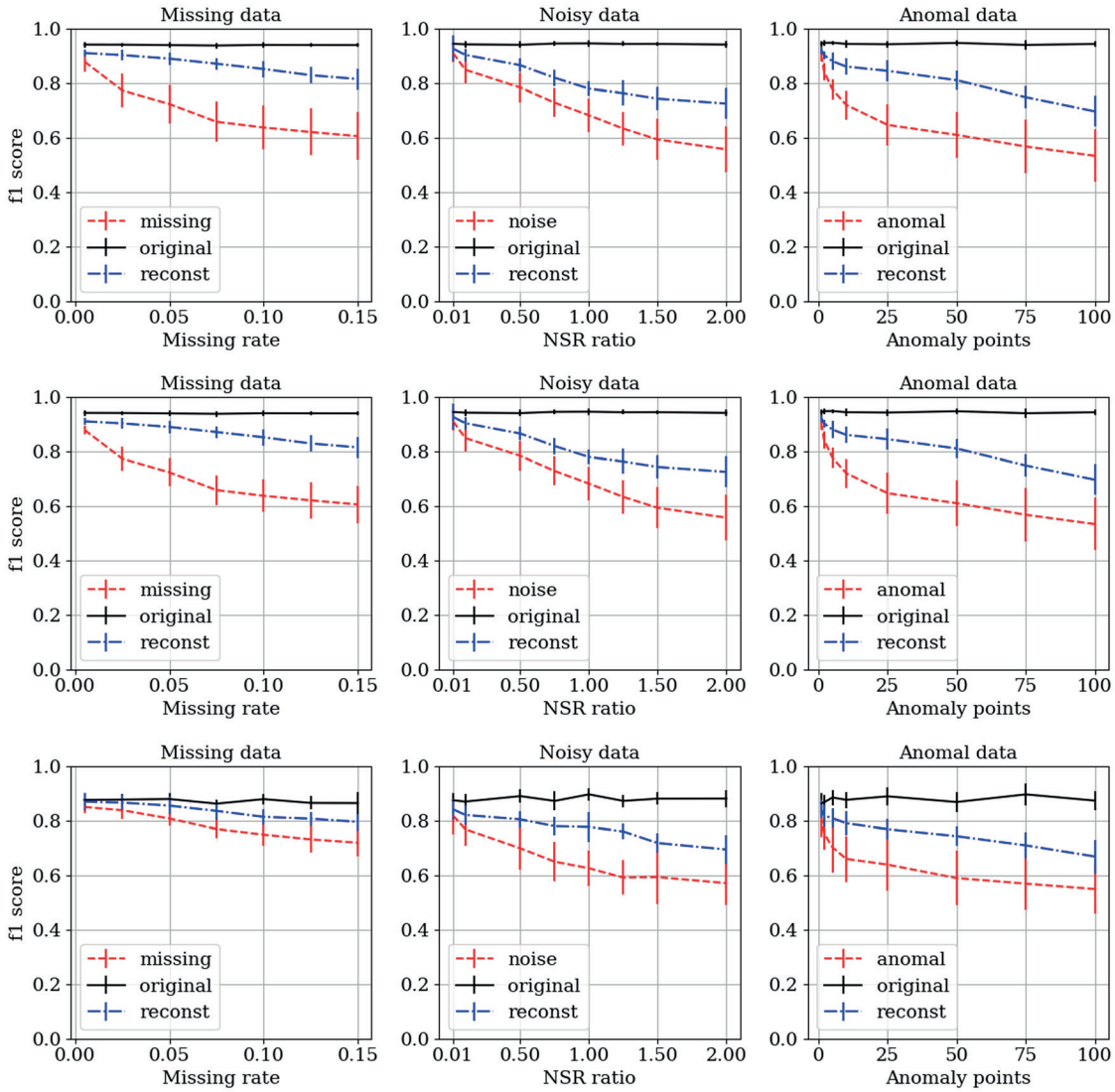

Fig. 9 Robustness results showing the performance of the proposed AutoBoost-SDD framework against various signal contaminations

that the method can estimate associated confidence intervals of detection results, making the decision more conservative when accounting for uncertainty in the model behaviors and input data. Another interesting direction is to extend the proposed method to an online learning framework that can evolve with human feedback and new vibration data corresponding to new types of damage. This could be realized by combining the AutoBoost-SDD framework with reinforcement learning or game theory algorithms.

Acknowledgement

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

References

[1] Kaveh, A., Hosseini, S. M., Zaerreza, A. "Boundary Strategy for Optimization-based Structural Damage Detection Problem using Metaheuristic Algorithms", Periodica Polytechnica Civil Engineering, 65(1), pp. 150–167, 2021. https://doi.org/10.3311/PPci.16924

[2] Kaveh, A., Dadras, A. "Structural damage identification using enhanced thermal exchange optimization algorithm", Engineering Optimization, 50(3), pp. 430–451, 2018. https://doi.org/10.1080/0305215X.2017.1318872

[3] Kaveh, A., Maniat, M. "Damage detection based on MCSS and PSO using modal data", Smart Structures and Systems, 5(15), pp. 1253–1270, 2015. https://doi.org/10.12989/sss.2015.15.5.1253

[4] Deraemaeker, A., Reynders, E., De Roeck, G., Kullaa, J. "Vibration-based structural health monitoring using output-only measurements under changing environment", Mechanical Systems and Signal Processing, 22(1), pp. 34–56, 2008. https://doi.org/10.1016/j.ymssp.2007.07.004

[5] 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

[6] 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

[7] Dang, V.-H., Vu, T.-C., Nguyen, B.-D., Nguyen, Q.-H., Nguyen, T.-D. "Structural damage detection framework based on graph convolutional network directly using vibration data", Structures, 38, pp. 40–51, 2022. https://doi.org/10.1016/j.istruc.2022.01.066

[8] Dang, V.-H., Raza, M., Tran-Ngoc, H., Bui-Tien, T., Nguyen, H. X. "Connection stiffness reduction analysis in steel bridge via deep CNN and modal experimental data", Structural Engineering and Mechanics, 77(4), pp. 495–508, 2021. https://doi.org/10.12989/sem.2021.77.4.495

[9] Kullaa, J. "Eliminating environmental or operational influences in structural health monitoring using the missing data analysis", Journal of Intelligent Material Systems and Structures, 20(11), pp. 1381–1390, 2009. https://doi.org/10.1177/1045389X08096050

[10] Kourehli, S. S., Bagheri, A., Amiri, G. G., Ghafory-Ashtiany, M. "Structural damage detection using incomplete modal data and incomplete static response", KSCE Journal of Civil Engineering, 17, pp. 216–223, 2013. https://doi.org/10.1007/s12205-012-1864-2

[11] Yin, T., Jiang, Q.-H., Yuen, K.-V. "Vibration-based damage detection for structural connections using incomplete modal data by Bayesian approach and model reduction technique", Engineering Structures, 132, pp. 260–277, 2017. https://doi.org/10.1016/j.engstruct.2016.11.035

[12] 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

[13] Fan, G., Li, J., Hao, H. "Lost data recovery for structural health monitoring based on convolutional neural networks", Structural Control and Health Monitoring, 26(10), e2433, 2019. https://doi.org/10.1002/stc.2433

[14] Jiang, H., Wan, C., Yang, K., Ding, Y., Xue, S. "Continuous missing data imputation with incomplete dataset by generative adversarial networks–based unsupervised learning for long-term bridge health monitoring", Structural Health Monitoring, 21(3), pp. 1093–1109, 2022. https://doi.org/10.1177/14759217211021942

[15] Ibrahim, A., Eltawil, A., Na, Y., El-Tawil, S. "A machine learning approach for structural health monitoring using noisy data sets", IEEE Transactions on Automation Science and Engineering, 17(2), pp. 900–908, 2020. https://doi.org/10.1109/TASE.2019.2950958

[16] de Castro, B. A., Baptista, F. G., Ciampa, F. "New signal processing approach for structural health monitoring in noisy environments based on impedance measurements", Measurement, 137, pp. 155–167, 2019. https://doi.org/10.1016/j.measurement.2019.01.054

[17] Das, S., Saha, P. "Performance of hybrid decomposition algorithm under heavy noise condition for health monitoring of structure", Journal of Civil Structural Health Monitoring, 10, pp. 679–692, 2020. https://doi.org/10.1007/s13349-020-00412-5

[18] Ma, S.-L., Jiang, S.-F., Li, J. "Structural damage detection considering sensor performance degradation and measurement noise effect", Measurement, 131, pp. 431–442, 2019. https://doi.org/10.1016/j.measurement.2018.08.040

[19] Dang, V.-H., Vu, T.-C., Nguyen, B.-D., Nguyen, Q.-H., Nguyen, T.-D. "Structural damage detection framework based on graph convolutional network directly using vibration data", Structures, 38, pp. 40–51, 2022. https://doi.org/10.1016/j.istruc.2022.01.066

[20] Worden, K., Farrar, C. R., Manson, G., Park, G. "The fundamental axioms of structural health monitoring", Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 463(2082), pp. 1639–1664, 2007. https://doi.org/10.1098/rspa.2007.1834

[21] 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(3), pp. 447–453, 1984. https://doi.org/10.1115/1.3269216

[22] 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

[23] 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

[24] Dang, 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

[25] Tang, Z., Chen, Z., Bao, Y., Li, H. "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring", Structural Control and Health Monitoring, 26(1), e2296, 2019. https://doi.org/10.1002/stc.2296

[26] Kaveh, A., Servati, H. "Design of double layer grids using back-propagation neural networks", Computers and Structures, 79(17), pp. 1561–1568, 2001. https://doi.org/10.1016/S0045-7949(01)00034-7

[27] Iranmanesh, A., Kaveh, A. "Structural optimization by gradient base neural networks", International Journal of Numerical Methods in Engineering, 46(2), pp. 297–311, 1999. https://doi.org/10.1002/(SICI)1097-0207(19990920)46:2%3C297::AID-NME679%3E3.0.CO;2-C

[28] Kaveh, A., Elmieh, R., Servati, H. "Prediction of moment-rotation characteristic for semi-rigid connections using BP neural networks", Asian Journal of Civil Engineering, 2(2), pp. 131–142, 2001. [online] Available at: https://sid.ir/paper/298583/en

[29] Dang, V.-H., Le-Nguyen, K., Nguyen, T.-T. "Semi-supervised vibration-based structural health monitoring via deep graph learning and contrastive learning", Structures, 51, pp. 158–170, 2023. https://doi.org/10.1016/j.istruc.2023.03.011

[30] Kaveh, A., Khavaninzadeh, N. "Efficient training of two ANNs using four meta-heuristic algorithms for predicting the FRP strength", Structures, 52, pp. 256–272, 2023. https://doi.org/10.1016/j.istruc.2023.03.178

[31] Feichtenhofer, C., Fan, H., Li, Y., He, K. "Masked autoencoders as spatiotemporal learners", presented at 36th Conference on Neural Information Processing Systems (NeurIPS2022), Online, Nov, 28. – Dec, 9., 2022. ISBN: 9781713871088 https://doi.org/10.48550/arXiv.2205.09113

[32] Kaveh, A., Iranmanesh, A. "Comparative study of backpropagation and improved counter propagation neural nets in structural analysis and optimization", International Journal of Space Structures, 13(4), pp. 177–185, 1998. https://doi.org/10.1177/026635119801300401

[33] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., ..., Chintala, S. "Pytorch: An imperative style, high-performance deep learning library", presented at 33rd Conference on Neural Information Processing Systems (NeurIPS2019), Vancouver, Canada, Dec, 8–14, 2019. ISBN: 9781713807933 https://doi.org/10.48550/arXiv.1912.01703

[34] Loh, W.-Y. "Classification and regression trees", WIREs Data Mining and Knowledge Discovery, 1(1), pp. 14–23, 2011. https://doi.org/10.1002/widm.8

[35] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y. "Lightgbm: A highly efficient gradient boosting decision tree", presented at 31st Conference on Neural Information Processing Systems (NIPS2017), Long Beach, CA, USA, Dec, 4–9, 2017. ISBN: 9781510860964 https://doi.org/10.5555/3294996.3295074

[36] 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

[37] Nakashima, M., Kurata, M., Rathje, E., Sharifi Mood, M. "Quantification of collapse margin for steel high-rises", DesignSafe-CI, PRJ-1640, 2018. https://doi.org/10.17603/DS2T375

[38] Chen, T., Guestrin, C. "XGBoost: A scalable tree boosting system", In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2 https://doi.org/10.1145/2939672.2939785

[39] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., Gulin, A. "CatBoost: unbiased boosting with categorical features", Advances in Neural Information Processing Systems, Preprint, arXiv:1706.09516 [cs.LG], 2018. https://doi.org/10.48550/arXiv.1706.09516