A novel data-driven sensor placement optimization method for unsupervised damage detection using noise-assisted neural networks with attention mechanism

Sheng Shi [a,b], Dongsheng Du [b,*], Oya Mercan [c], Erol Kalkan [d], Shuguang Wang [b]

[a] *Changshu Institute of Technology, Suzhou, China*
[b] *Nanjing Tech University, Nanjing, China*
[c] *University of Toronto, Toronto, ON, Canada*
[d] *CEO, QuakeLogic Inc., Roseville, CA, USA*

ARTICLE INFO

ABSTRACT

Optimization of sensor placement (OSP) is one of the important steps in structural health monitoring to reduce the instrumentation cost, and improve damage detection. Although modal analysis is an optional intermediate process in terms of damage detection, conventional OSP methods for damage detection are mostly dependent on mode shapes. In this case, the sensor placement and damage detection are highly relying on the accuracy of modal analysis, and the results may not be adapted to different type of excitations. In this article, a novel noise-assisted neural network with attention mechanism is proposed based on the above challenges. This method which can be used to optimize sensor placement in an unsupervised and data-driven manner, is verified using a dataset simulated from the ASCE benchmark and an experimental dataset obtained from shake table tests. The results from the simulated dataset show that the percentage of the sensors that could be removed are higher than the conventional effective in-dependence (EFI) method, with a highest of 62.5% for cases with low noise levels. In the meantime, the occurrence and the level of damage can still be well detected with a reduced number of sensors. Most importantly, in the proposed approach the optimal sensor placement configurations can be determined adaptively to account for different forms of excitations and noise levels, which is impossible for conventional model-driven methods. The results obtained from the experimental dataset show that the proposed method is also effective for real-world applications. As a result, the proposed data-driven OSP method skips the conventional model analysis process and directly focuses on the sensor arrangement that enables accurate detection of damage. It also has the potential for application in related fields, such as the monitoring of aerospace and mechanical infrastructures.

1. Introduction

To ensure safety and reliability of civil structures, inspections should be conducted periodically to evaluate their conditions. Conventionally, inspections are conducted on site by technicians. With the rapid increase of the number of critical civil structures, the

---

demand for inspections is also increasing. With the development of data acquisition, storage and analysis techniques, AI (artificial intelligence)-powered structural health monitoring (SHM) systems are more widely investigated and implemented [1,2].

Traditionally, SHM is referred to as the process of implementing a damage identification strategy for aerospace, civil, and mechanical infrastructures [3]. Vibration-based damage detection approaches are among the earliest SHM techniques, which use vibrational data such as acceleration to assess the structural condition. Nowadays, with the implementation of machine learning techniques, vibration-based approaches can be considered in two different categories, namely the model-driven and data-driven approaches. The model-driven approaches cover most of the conventional methods that detect damage by monitoring the damage-sensitive features constructed by modal parameters, including modal assurance criterion (MAC) [4], coordinate modal assurance criterion (COMAC) [5], modal curvature [6,7], modal flexibility [8], modal strain energy [9], etc. While the damage-sensitive features monitored by model-driven methods are hand-crafted according to physical laws, damage-sensitive features monitored by the data-driven methods are automatically constructed by using historical data, and therefore they have no explicit physical meaning. Typical data-driven damage detection are realized by training a classifier represented by support vector machine [10–14], random forest [15–18], multi-layer perceptron [19–22], convolutional neural networks [23–27], deep believe networks [28,29]. Due to the requirement of labeled data during training, these classifier-based methods are considered as supervised. However, despite the effectiveness of supervised methods in numerical simulation, their feasibility is largely limited due to the lack of data from the actual damaged civil structures [30]. Therefore, some unsupervised methods [31–36] were developed. These unsupervised methods need to be trained with healthy data only, which makes them more suitable for real-world applications.

The optimization of sensor placement (OSP) is one of the important procedures in both model-driven and data-driven SHM, which can be described as a combination of optimization problem [37]: "given a set of n candidate positions, finding m positions, where $m \leq n$, which could maximize or minimize the value of the proposed evaluation criterion for the sensor configuration". There are three main reasons [38] why the OSP is important:

1) To improve the performance of SHM system by selecting the sensor locations that are most sensitive to damage, and minimize irrelevant or redundant data.
2) To reduce the instrumentation cost by optimizing number of sensors.
3) To decrease SHM system operating costs, such as data management and energy consumption.

For vibration-based SHM, current OSP methods are mostly model driven, which aim to optimize a hand-crafted cost function related to the mode shapes. One of the earliest OSP methods is effective independence (EFI) that optimizes sensor locations by maximizing the determinant of Fisher information matrix (FIM) [39,40]. However, since EFI-based methods are prone to noise, several alternatives have been proposed. In these methods, the cost functions designed for optimization include modal kinetic energy distribution [41], information entropy [42], MAC [4,43], singular value decomposition ratio (SVDR) [38], etc. Apart from the conventional approaches, some recent researches start considering the influence of uncertain factors such as the actual load cases and structural responses [44], potential damage scenarios [45]. In addition to its application in civil engineering, the OSP is also applied to various research fields, such as robotics [46], fluid dynamics [47,48], and plant sensing [49]. Despite the achievements of existing methods, there are three main problems remained:

1) Although modal analysis is an optional intermediate process in terms of damage detection, conventional model-driven OSP methods in damage detection highly rely on the modal parameters calculated by experimental or numerical approaches, making the accuracy of modal analysis extremely important. However, experimental modal analysis methods are sensitive to noise. For the numerical methods, accurate material information and boundary conditions of structural components are difficult to obtain.
2) The OSP schemes are not only related to the structure itself, but also the type of excitations and the quality of data. Conventional methods normally assume that the structure is subjected to known excitation and the dynamic response is perfectly collected. However, the actual excitation is usually not exactly the same as assumed, and the quality of the recorded data is affected by the noise and the condition of data collection systems.
3) Currently, all mentioned OSP methods are model-driven and only applicable for damage detection based on modal parameters. To the best of the authors' knowledge there is no investigation that focuses on the data-driven OSP methods applicable for unsupervised data-driven damage detection.

Based on these considerations, this paper aims to develop a novel data-driven OSP method for unsupervised damage detection, which skips the model analysis process and directly focuses on the sensor arrangement that enables accurate detection of damage. For this purpose, a new type of neural networks combining the concept of noise-assisted analysis and neural networks with attention mechanism is proposed; thence it is referred to as NANNAM (noise-assisted neural networks with attention mechanism). The noise-assisted approach allows for measuring the relative contribution of data obtained from candidate positions compared to the artificial noise. The attention mechanism forces the neural networks to pay more attention to the data obtained from important positions, which helps select optimal sensor positions in an unsupervised and data-driven manner.

One of the advantages of the proposed method is purely data-driven, as a result, no physical information or modal analysis is required. More importantly the selected optimal sensor positions are automatically adjusted according to the type of excitations and the quality of data. The number of sensors will be automatically increased with an increase in the noise levels to provide redundancy, and more sensors will be arranged around the positions where excitations are applied. For structures to be monitored, this method can provide guidance on the minimum number, and the optimal locations of sensors. For structures already instrumented with sensors, this

method can be used to identify important data channels to reduce the computational cost, and eliminate the irrelevant or noisy data channels. In addition to civil engineering, the proposed method also has the potential for application in related fields, such as the monitoring of aerospace and mechanical infrastructures.

The rest of the paper is organized as follows: Section 2 introduces related works including channel importance measurement and noise-assisted analysis. Section 3 describes the architecture and training algorithm of the proposed NANNAM. Section 4 and Section 5 present the test results for the simulated International Association for Structural Control–American Society of Civil Engineering (IASC-ASCE) Benchmark dataset and University of California, San Diego (UCSD) shake table dataset, respectively. Finally, the discussion and conclusions are presented in Section 6 and Section 7. All of the algorithms investigated in this paper are implemented using Pytorch [50].

2. Literature review

2.1. Channel importance measurement

For data-driven damage detection methods, the OSP problem requires accurate assessment of the importance of input channels. In machine learning, the neural networks are trained to pay more attention to the important input data channels which is also known as the attention mechanism. Once the importance of input channels is calculated, data channels with higher importance measurement are considered as the optimal solution for the sensor placement. Data-driven channel importance measurement methods for neural networks can be summarized in three main categories as follows:

1) **Channel importance measurement directly calculated by weights**. In a neural network with fully connected layers, the neurons embedded in the input layer and the first hidden layer are connected by weights and biases. The absolute values of weights reflect the contribution and importance of input layer neurons to the hidden layer neurons. For convolutional layers, the neurons embedded in the input layer and the first hidden layer are connected by convolutional kernels. The absolute values of elements in the kernels can reflect the contribution and importance of the input channels. Therefore, the weights of neural networks can be directly used to calculate the channel importance measurement. For example, pruning of the neural networks results in the removal of the irrelevant and redundant features and weights without influencing the loss function, thus the absolute values of weights are commonly used for removing less important neurons and connections during structured [51–53] and unstructured pruning [54–56]. Another feasible channel importance measurement is the gradient-based quantities [57,58] calculated by using the weights where input channels with lower mean gradient are removed during pruning.
2) **Channel importance measurement by factors tuned during training**. To automatically quantify the importance of the channels by neural networks themselves, trainable scaling factors [59], also known as gate[60] or gain [61], are introduced to the networks. In this approach, the input channels or neurons are multiplied by trainable scaling factors before they are connected to the next layer. During training, with sparsity regularization imposed on the scaling factors the values of the scaling factors are determined by suppressing the output of the channels with a small scaling factor as they are considered as redundant or irrelevant. Another typical example is class active mapping in computer vision, where the trainable combination coefficients after global average pooling layer are used to represent the contribution and importance of input image channels [62].
3) **Channel importance measurement represented by a sub-network**. For the cases when the scaling factors are dependent on the input, the channel importance varies for different input. Thus, fixed factors are no longer applicable and the channel importance should be measured by a sub-network connected to the input. This approach is adopted by some image processing [63] and natural language processing [64] methods to have the model automatically pay more attention to the important channels, pixels, and words.

The abovementioned channel importance measurements are all relative quantities making it difficult to select proper thresholds for pruning unimportant channels. To address this challenge the noise-assisted analysis is introduced as described next.

2.2. Noise-assisted analysis

In most cases noise is considered a source of disturbance for data analysis, however sometimes data analysis is assisted by the introduction of artificial noise. There are four well-known noise-assisted applications in data analysis, as follows:

1) **Adding noise to amplify weak data components**. A typical example under this category is the stochastic resonance method [65,66], which uses noise to help amplify weak periodic or quasi-periodic data components.
2) **Adding noise to suppress disturbing data components**. In EESD (ensemble empirical mode decomposition) [67] white-noise is used as a background to help data components of different scales to be decomposed into the proper IMFs (intrinsic mode functions). Subsequently the added white-noise is automatically canceled by averaging the ensemble. This noise-assisted approach relieves the mode-mixing problem in EMD (empirical mode decomposition) [68].
3) **Adding noise to test the anti-noise ability of algorithms**. Unlike numerical simulations, real-world data obtained by sensors is often distorted by unavoidable noise, which poses challenges for the performance of data processing algorithms. To test and evaluate them, artificial white noise is normally added to the dataset when conducting numerical experiments.

4) **Using white noise as a source of randomness**. For generative models like GAN (generative adversarial networks)[69], artificial white noise is fed into the input layer and then filtered by the neural networks to generate fake data that resembles the training target. Another application under this category is the Monte Carlo simulation [70,71], where white noise is used during the sampling of random parameters.

In this paper, the artificial white noise is used to assist data analysis in a new novel way. Since the channel importance measurements is a relative quantity, it is difficult to determine the threshold for channel pruning if there is no channel with known actual importance or contribution as a reference. Fortunately, the contribution of the noise channel is a priori information. By comparing the channel importance measurements of data channels and the noise channel, the relative contribution of data channels can be evaluated.

3. Noise-assisted neural networks with attention mechanism (NANNAM)

3.1. Unsupervised damage detection

The proposed OSP method is developed for the unsupervised damage detection, which detects structural damage by training with healthy monitoring data only. The unsupervised damage detection is composed of two main phases, namely the statistical modeling and decision-making [30].

During the statistical modeling phase damage-sensitive features are extracted using equation (1). Where $\varepsilon(k)$ is the extracted damage-sensitive feature; $x(k)$ is the measured acceleration response at time instant k; $E_{\theta_0}(\cdot)$ is the predictive function represented by neural networks, which calculates the expected acceleration response at time instant k by observing the data from recent p instants; θ_0 is a vector containing parameters of the predictive function; $X_{k-p}^{k-1} = \{x(k-1), x(k-2), \cdots, x(k-p)\}$ is a set of acceleration response within a moving window of length p.

$$\varepsilon(k) = x(k) - E_{\theta_0}\left(x(k)|X_{k-p}^{k-1}\right) \tag{1}$$

During the decision-making phase deep support vector domain description (DSVDD) is used to make decisions on the damage condition of the monitored structure based on the damage-sensitive features extracted during the first phase. The DSVDD maps low-dimensional features into a high-dimensional space, where the features from healthy data cluster within a minimal hypersphere. This hypersphere can be calculated by optimizing the equation (2).

$$\min_{R,\omega} R^2 + \frac{1}{\nu n_s}\sum_{i=1}^{n_s}\max\left\{0, \|\Phi(\varepsilon_i|\omega) - a\|^2 - R^2\right\} + \frac{\lambda}{2}\sum_i\|\omega_i\|_F^2 \tag{2}$$

where R is the radius of the hypersphere; $\nu \in (0,1)$ is a hyper-parameter controls the trade-off between the volume of the hypersphere and the violations of the boundary; n_s is the total number of training samples; $\Phi(\bullet|\omega)$ is the mapping function calculated by a neural network with unknown parameter ω; a is the center coordinate of the hypersphere; $\lambda > 0$ is a weight decay regularizer on the network parameters; $\|\bullet\|_F$ denotes the Frobenius norm. The condition of the monitored structure can then be determined by equation (3). Where R_{unk} is the distance between the features of unknow data and the center of the hypersphere; ε_{unk} is the feature calculated from the unknown data.

$$\text{condition} = \begin{cases} \text{healthy, if } R_{unk} \leqslant R \\ \text{unhealthy, if } R_{unk} > R \end{cases} \tag{3}$$

where

$$R_{unk} = \sqrt{\|\Phi(\varepsilon_{unk}|\omega) - a\|^2} \tag{4}$$

To evaluate true-positives and false-positives, MAD (mean alarm density) and MFAD (mean false alarm density) were adopted respectively, as presented in equation (5) and equation (6).

$$\text{MAD} = \frac{\sum_{i=1}^{N_s}H\left(R_i^d - R\right)}{N_s} \tag{5}$$

$$\text{MFAD} = \frac{\sum_{i=1}^{N_s}H\left(R_i^u - R\right)}{N_s} \tag{6}$$

where N_s is the number of samples in the investigated testing fragment; $H(\bullet)$ is the Heaviside function; R_i^d is the distance between the i-th feature calculated from the damaged testing data and the center of the hypersphere; R_i^u is the distance between the i-th feature calculated from the undamaged testing data and the center of the hypersphere. The MAD is also adopted as a measurement of damage degree.

3.2. The architecture of NANNAM

The predictive function $E_{\theta_0}(\cdot)$ in equation (1) is an important element of unsupervised damage detection, which is described by a neural network called NANNAM in this article. The most important innovation of this work lies in the proposal of NANNAM and its training algorithm that enables not only unsupervised damage detection but also data-driven OSP. It enables the sensor positions adaptive to various excitation forms and noise levels, which is currently impossible for conventional modal-driven methods.

The basic architecture of NANNAM is composed of an input layer with added artificial noise, trainable channel importance factors (CIFs), two inception layers, a flatten layer and an output layer, as illustrated in Fig. 1.

The added artificial white noise is the first difference between the proposed NANNAM and normal artificial neural networks. Each time when a data fragment is extracted by a moving window, a series of artificial white noise is generated and added to the data fragment as a new channel. The RMS (root mean square) of the artificial white noise equals to the maximum RMS of the other data channels. This process is essential for the evaluation of the relative contribution of each channel. If the contribution measurement of a certain channel is not significantly higher than the white noise channel, then this channel can be safely pruned without significantly affecting the output. Another difference between the proposed NANNAM and normal neural networks is the trainable CIFs, which are used to measure the contribution of each channel. This is a pure data-driven process and the CIFs are completely learned from data itself. The scaled input in the first part of Fig. 1 can be expressed by

$$SI(n) = \begin{pmatrix} CIF_1 x_1(n) \\ CIF_2 x_2(n) \\ CIF_3 x_3(n) \\ CIF_4 x_4(n) \\ CIF_w W(n) \end{pmatrix}, n = 1, 2, 3, \cdots, p \tag{7}$$

where $SI(n)$ is the element of scaled input at n-th time instant; CIF_i and CIF_w are the CIFs of the i-th data channel and the white noise channel respectively; $x_i(n)$ and $W(n)$ is the n-th element of the i-th data channel and the added white noise channel; p is the length of the moving window.

The scaled input is then processed by two inception layers, which are proposed and used in the GoogLeNet [72]. The main idea of inception layer is to find an optimal local sparse structure for a convolutional neural network, which allows for increasing the depth and width of the network while maintaining the computational efficiency. The inception layer used here is composed of 11 1D-CNN layers with different kernel size and LReLU (leaky rectified linear unit) [73] activation functions. The output of inception layer can be expressed by

$$IL(n,m) = f_a \left(\sum_{v=n-k_m}^{n+k_m} \sum_{c=1}^{5} K^{(m)}(v,c) SI(v,c) \right) \tag{8}$$
$$n = 1, 2, 3, \cdots, p; m = 1, 2, 3, \cdots, 11$$

where $IL(n,m)$ is the output of inception layer; $f_a(\bullet)$ is the LReLU activation function; k_m is the radius of m-th kernel, which is 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, respectively for each of the 11 kernels; $K^{(m)}(v,c)$ is the element of m-th kernel at v-th instant and c-th channel. A flatten layer is then connected to the second inception layer so that the two-dimensional tensor can be ready as the input for the following fully connected layer.

The statistical characteristics of response obtained from the same floor are normally similar, which means these data is sparse and the number of parameters in the neural networks can be further reduced. Hence, for cases when several sensors are instrumented on the same floor, the NANNAM with grouped input is developed here to process data in groups. As illustrated in Fig. 2, the data obtained
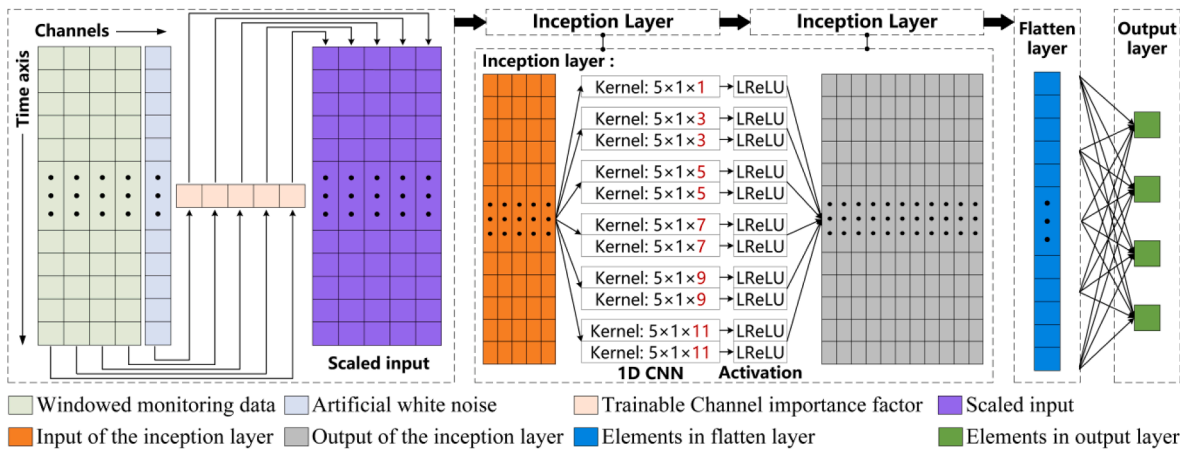

Fig. 1. Basic architecture of the proposed NANNAM (Noise-assisted neural networks with attention mechanism).

from each floor is first organized in groups, then each group of the data is respectively connected to the NANNAM and processed in parallel. In this case, the RMS of the artificial white noise equals to the maximum RMS of the data channels within the group.

3.3. Preprocessing of the input data

Data preprocessing is an important procedure before training. There are two steps used for data preprocessing, namely the baseline adjustment and standardization. The baseline adjustment is meant to remove the trend introduced during measurement to ensure the input data is zero-centered, which is especially important for real-world data. The standardization aims to make the dataset internally consistent by scaling the data with a factor. Since the relative amplitude of the data from different channels is related by the structural damage condition, the information of the relative amplitude should be retained during standardization. Hence, the standardization for NANNAM is conducted by scaling the data from all channels with the same factor, as described in equation (9).

$$x_i' = \frac{x_i}{\max\limits_j \sqrt{\frac{1}{N} \sum\limits_n x_j(n)^2}} \tag{9}$$

where x_i and x_i' are the data from i-th channel before and after standardization respectively; the denominator is the maximum RMS of data among all channels.

3.4. The training algorithm of NANNAM

For NANNAM, the trainable CIFs and other network parameters should be trained simultaneously. However, to obtain stable results a warm-up process is introduced to the training algorithm, as shown in Fig. 3. The training algorithm of NANNAM is therefore composed of two main blocks, namely training algorithm I for warm-up and training algorithm II, which are illustrated in detail in Fig. 4.

During the warm-up stage when the training algorithm I shown in Fig. 4a is used, the CIFs are fixed to 1 and temporarily untrainable. Then the other network parameters are tuned by the backpropagation [74,75] of training loss TL_2, which can be expressed by:

$$TL = TL_2 = \sum_n \sum_{i=1}^4 (y_i(n) - \widehat{y}_i(n))^2 + \gamma_2 \sum w \tag{10}$$

where $y_i(n)$ and $\widehat{y}_i(n)$ are the measured and predicted acceleration response for i-th channel at n-th instant; γ_2 is the weight decay for network parameters; w is a vector of trainable network parameters. Each time the validation loss VL is less than the minimum previous validation loss PVL, the model is saved. Until the training epoch N_0 is higher than P_0, the warm-up process is then finished.

In training algorithm II, the CIFs and network parameters start to be trained simultaneously. The training loss at this stage is calculated by

$$TL = TL_1 + TL_2$$

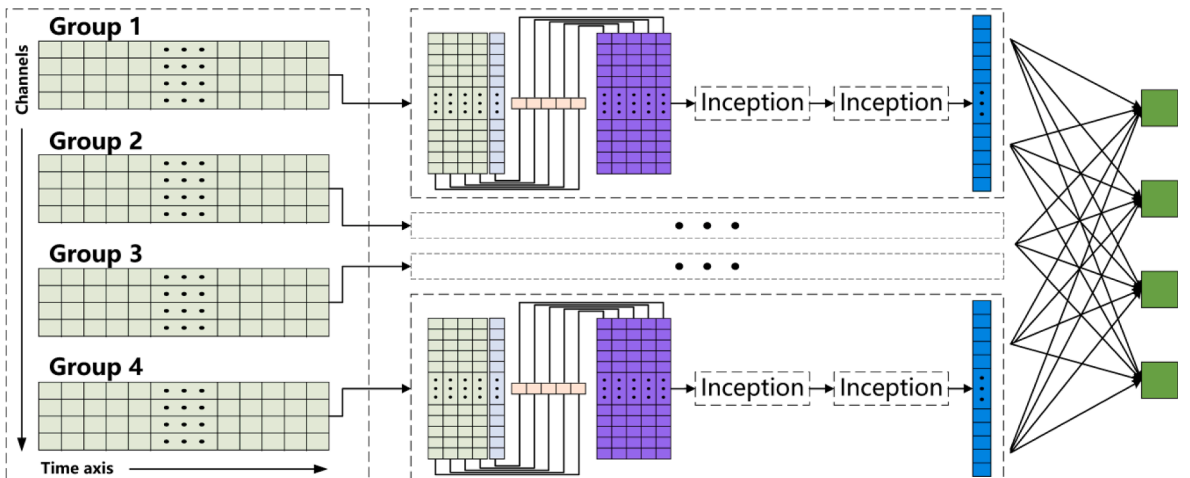$$TL_1 = \frac{\gamma_1}{2\pi} \arctan \left(\sum_{i=1}^4 CIF_i + CIF_w \right) \tag{11}$$

Fig. 2. Architecture of the NANNAM (Noise-assisted neural networks with attention mechanism) with grouped input.
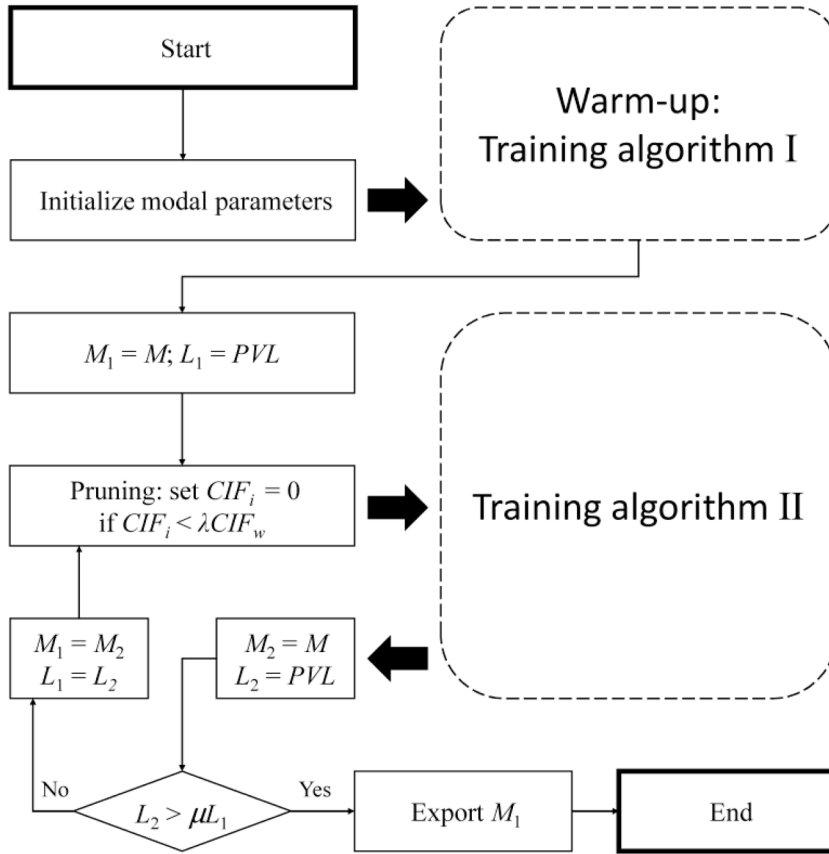
Fig. 3. The training algorithm of NANNAM.

where TL_2 is the training loss of network parameters that already described in equation (10); γ_1 is the weight decay for CIFs, and a higher value of γ_1 means fewer channels will be selected; CIF_i and CIF_w are the CIFs of the i-th data channel and the white noise channel respectively. The arctangent function is equivalent to the activation function for TL_1, which is essential for the selection of optimal sensor positions by suppressing CIFs with low values. Using the backpropagation, the CIFs and network parameters are trained respectively with dynamic learning rates. When the validation loss has not decreased for P_3 successive epochs, the training algorithm II is finished.

After two stages of training, the loop in Fig. 3 will be terminated when no channels are left to be pruned without affecting the output. The training parameters used in the algorithm are suggested in the Table 1.

4. Verification using the benchmark dataset

In this section, the proposed method is verified on the benchmark dataset. The conventional and widely-used EFI [40] is adopted as a comparison. This method reduces the number of sensors by removing low-ranked sensor locations iteratively. The importance of the sensor locations is measured by the diagonal of a matrix E_D calculated from mode shapes, as presented in equation (12).

$$E_D = \Phi[\Phi^T\Phi]^{-1}\Phi^T \tag{12}$$

where Φ is the matrix of mode shapes partitioned to the candidate sensor locations. The optimal sensor positions calculated by the proposed method are presented in Fig. 9 and Fig. 10.

4.1. The benchmark dataset

The benchmark dataset is simulated from a four-story two-bay by two-bay steel frame (illustrated in Fig. 5a) developed by IASC (International Association for Structural Control) and ASCE (American Society of Civil Engineering) [76]. Each floor has a 2.5 m × 2.5 m plan and is 0.9 m tall. The floor slab of analytical model is assumed to be rigid, and the mass of each floor is condensed at the central nodes. Therefore, the analytical model has 12 degrees of freedom.

There are four slabs on each floor. Normally the mass of slabs on each floor is set as the same to obtain a simple symmetric structure,
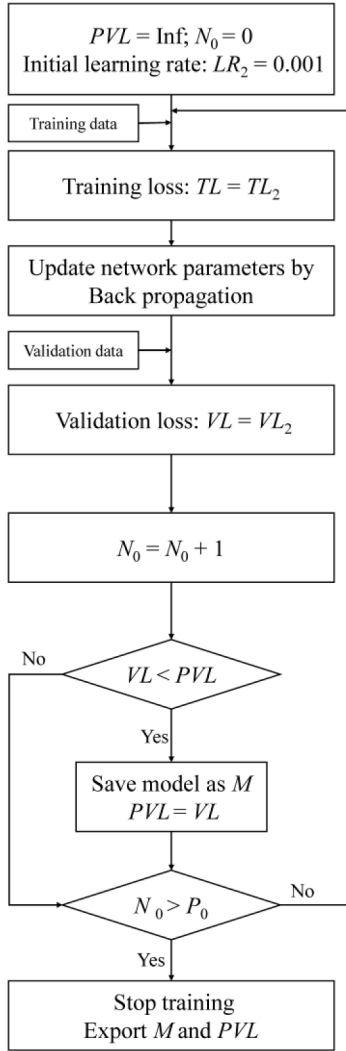
(a) Training algorithm I: warm-up   (b) Training algorithm II
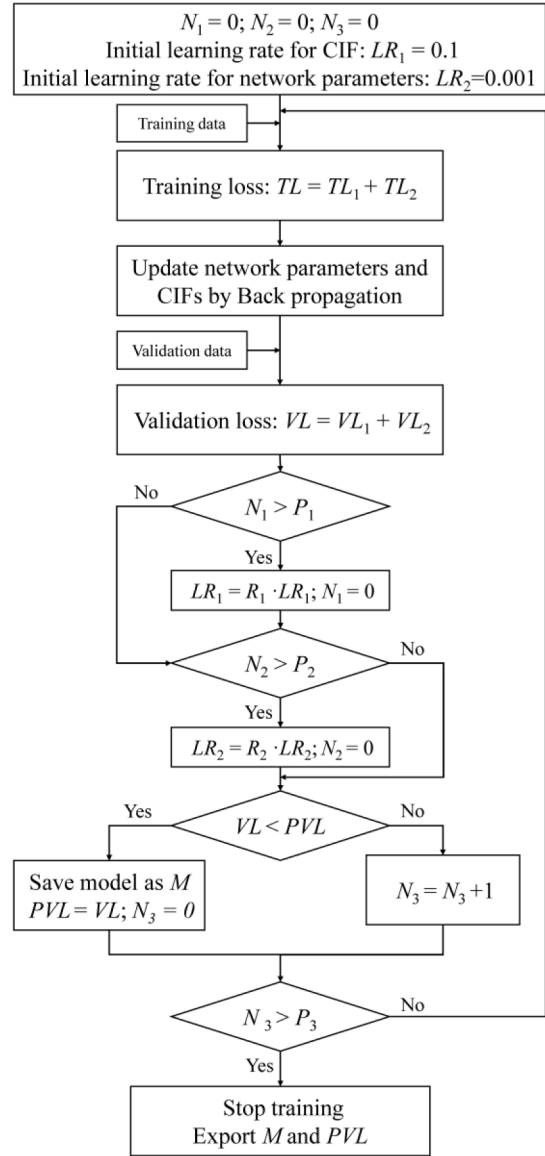
Fig. 4. Two blocks of training algorithms for two stages.

Table 1
Suggested values of training parameters.

Parameter	algorithm I	algorithm II	Pruning loop							
	P_0	γ_2	P_1	P_2	P_3	R_1	R_2	γ_1	λ	μ
Value	10	10^{-3}	2	4	10	0.3	0.3	10^{-3}	10	0.05

but damage detection may become easier for asymmetric damage patterns due to the unexpected rotational response [30]. In this study, a more complexed asymmetric case is considered: four 800 kg slabs on the first floor, four 600 kg slabs on the each of second and third floors, three 400 kg slabs and one 550 kg slab on the fourth floor.

4.2. Six damage patterns

As shown in Fig. 5b through 5 g, there are six damage patterns predefined for the benchmark structure simulated by reducing the
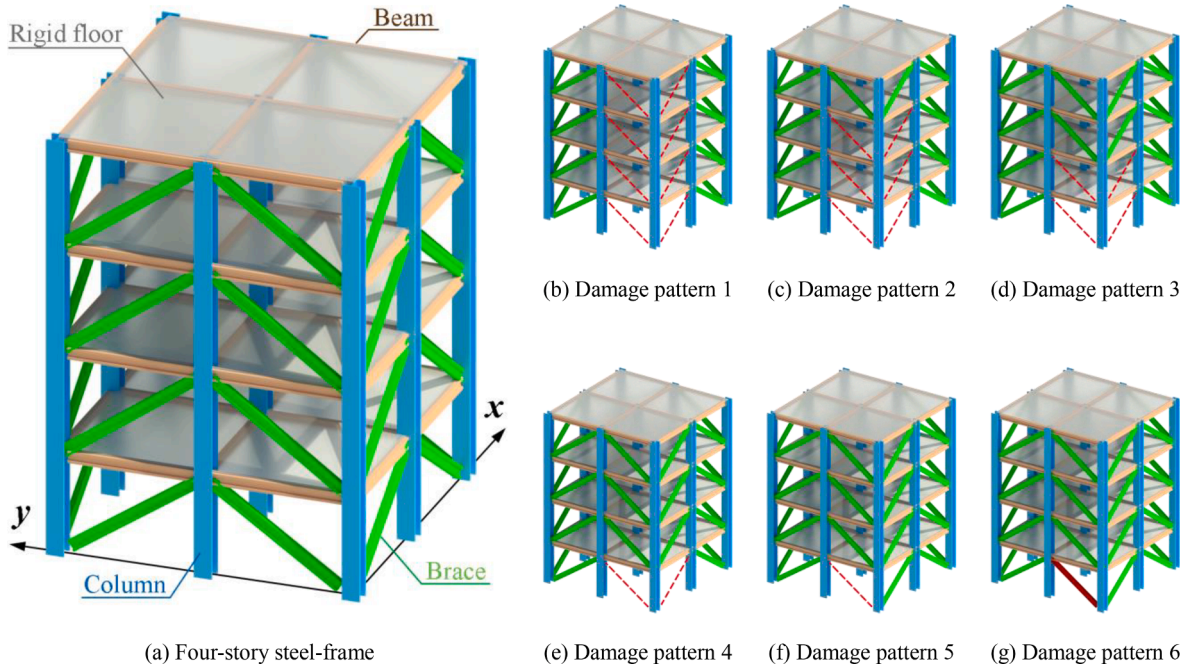
(a) Four-story steel-frame   (b) Damage pattern 1   (c) Damage pattern 2   (d) Damage pattern 3

(e) Damage pattern 4   (f) Damage pattern 5   (g) Damage pattern 6

Fig. 5. (a) The benchmark structure and (b)~(g) six damage patterns.

stiffness of certain number of braces. In damage pattern 1 (DP1), the stiffness of two braces on each floor is reduced by 100 %. In damage pattern 2 (DP2), the stiffness of two braces on each of the first, second and third floor is reduced by 100 %. In damage pattern 3 (DP3), the stiffness of two braces on each of the first and second floor is reduced by 100 %. In damage pattern 4 (DP4), the stiffness of two braces on the first floor is reduced by 100 %. In damage pattern 5 (DP5), the stiffness of one brace on the first floor is reduced by 100 %. In damage pattern 6 (DP6), the stiffness of one brace on the first floor is reduced by 1/3.

Structural damage reduces inter-story stiffness, and thereby alters the modal characteristics. The first five modal frequencies and their percentage changes after damage are presented in Table 2. As revealed by the mean percentage change of the first five modal frequencies, the severity of six damage patterns can be sorted in a descending order, i.e., DP1 > DP2 > DP3 > DP4 > DP5 > DP6. The mean percentage change in DP 6 is only 4.1 %, which poses a challenge to the damage detection under high noise.

4.3. Two loading cases

To test the influence of different loading conditions on the OSP results, two loading cases are defined.

Loading case 1 (loading at the roof level): for the first loading case, excitations are applied to the center of the roof, which can be represented by equation (13). Where $F_{1,x}$ and $F_{1,y}$ are the forces applied to the x and y directions respectively; M_1 is the magnitude of generated forces; F_G is a set of time series sampled from the standard Gaussian process.

$$\begin{bmatrix} F_{1,x} \\ F_{1,y} \end{bmatrix} = M_1 F_G \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$ (13)

Loading case 2 (loading at all stories): for the second loading case, excitations are generated by ambient environmental forces applied to each floor. In this case, the forces can be represented by equation (14). Where $F_{2,x}$ and $F_{2,y}$ are vectors of the forces applied to each floor in the x and y directions respectively; M_2 is the magnitude of generated forces; $F_{G,i,x}$ and $F_{G,i,y}$ are source ambient forces for the i-th ($i = 1, 2, 3, 4$) floor in the x and y directions, independently sampled from the standard Gaussian process.

Table 2
The first five modal frequencies of DP0 and percentage changes after damage.

Direction	DP0 (Hz)	DP1	DP2	DP3	DP4	DP5	DP6
x	9.29	−10.6 %	−10.3 %	−8.8 %	−5.5 %	−5.4 %	−1.5 %
y	11.64	−7.2 %	−7.0 %	−5.9 %	−3.6 %	0.0 %	0.0 %
Rotational	16.19	−7.3 %	−7.2 %	−6.5 %	−4.6 %	−2.4 %	−0.7 %
x	25.27	−10.6 %	−8.9 %	−4.6 %	−3.6 %	−3.6 %	−1.1 %
y	38.26	−23.2 %	−22.1 %	−19.7 %	−19.3 %	−17.2 %	−17.2 %
Mean percentage change		−11.8 %	−11.1 %	−9.1 %	−7.3 %	−5.7 %	−4.1 %

$$\begin{bmatrix} \boldsymbol{F}_{2,x} \\ \boldsymbol{F}_{2,y} \end{bmatrix} = M_2 \begin{bmatrix} F_{G,1,x} & F_{G,2,x} & F_{G,3,x} & F_{G,4,x} \\ F_{G,1,y} & F_{G,2,y} & F_{G,3,y} & F_{G,4,y} \end{bmatrix} \tag{14}$$

The dynamic response at locations of interest can be measured by sensors, but the input excitations from these two loading cases are assumed to be unknown. This means that the damage detection and OSP must be conducted in an output-only condition.

4.4. Data generation and splitting

Acceleration response at certain nodes is recorded by 16 sensors, as illustrated in Fig. 6. The collected acceleration data is used for the training, validation, and testing of neural networks.

To simulate the real-world conditions and evaluate the anti-noise ability of the proposed method, white noise with levels of 0 % ~20 % is added to the collected acceleration data using the equation (15).

$$y_i(t) = \eta\varepsilon(t)\max_i\left\{ \sqrt{\frac{\int_{t_1}^{t_2} x_i^2(t)dt}{t_2 - t_1}} \right\} + x_i(t) \tag{15}$$

Where $y_i(t)$ and $x_i(t)$ ($i = 1, 2, \cdots, 16$) is the i-th channel data with and without noise respectively; η is the noise level, which is set as 0 %, 2 %, 4 %, 6 %, 8 %, 10 %, 20 %; $\varepsilon(t)$ is a time series subject to standard normal distribution; t_1 and t_2 are the start time and end time of the selected data fragment.

As presented in Table 3, three fragments of acceleration data are simulated from the undamaged structure for training, validation, and testing. Six fragments of acceleration data for six damage patterns are generated for testing. The duration of each fragment is 1 h, and the sampling frequency is 20 Hz.

4.5. Window size selection

Window size is an important hyper-parameter that should be determined before model training. Window size determines the length of data inspected by the algorithm at each step. An over-estimated window size will reduce the computational efficiency and accuracy by including redundant information, while an under-estimated window size will reduce the accuracy by neglecting useful information [30]. The optimal window size is also related to the noise level and excitations, which are difficult to be identified for real-world situation. Therefore, a window size determination method for the proposed model is presented in this section.

The optimal window size is determined as the length of data that contains just enough information, and thus ensures both accuracy and efficiency of the algorithm. To obtain the optimal window size, an enumeration-based strategy is adopted. The candidate window sizes are first discretized within the range 100, then the RMSE (root mean square error) between the predicted and measured acceleration response is obtained for each case. The optimal window size is selected when the trend of RMSE stops decreasing based on the fact that the prediction error reflects the information from the input data that can be exploited. Similar approaches can be found in the model-order selection for the auto-regressive analysis of time series [77].

For widow size selection, to improve the efficiency of enumeration, the NANNAM is first reduced to a normal neural network by removing the introduced artificial white noise channel and CIFs, and then the reduced model is trained on the reduced dataset, i.e., the data amount is reduced to 1/3 of the fragment defined above.

For the first loading case (loading at the roof level only), with acceleration response from different floors as the prediction target, the RMSE corresponding to each window size under various noise levels are presented in Fig. 7. As can be seen, the RMSEs are decreasing when more information is included into the input with the increase of window size. However, in some cases when the window size is large the RMSEs are instead increasing, revealing the negative effects of redundant and irrelevant information introduced by an over-estimated window. With increasing noise level, the optimal window size is increasing, indicating that for data with low SNRs (signal-to-noise ratios) the negative effects of noise can be alleviated by including more data using a larger window. Therefore, to obtain a low value of RMSE with low computational cost, the optimal window size is selected once the trend of RMSE
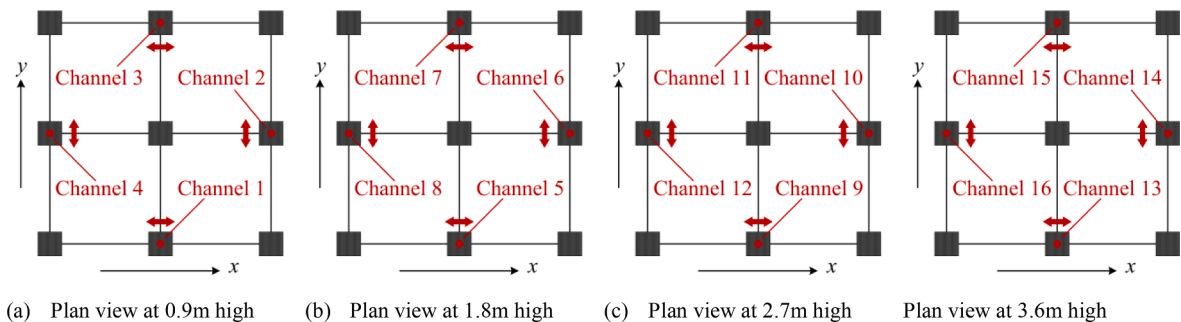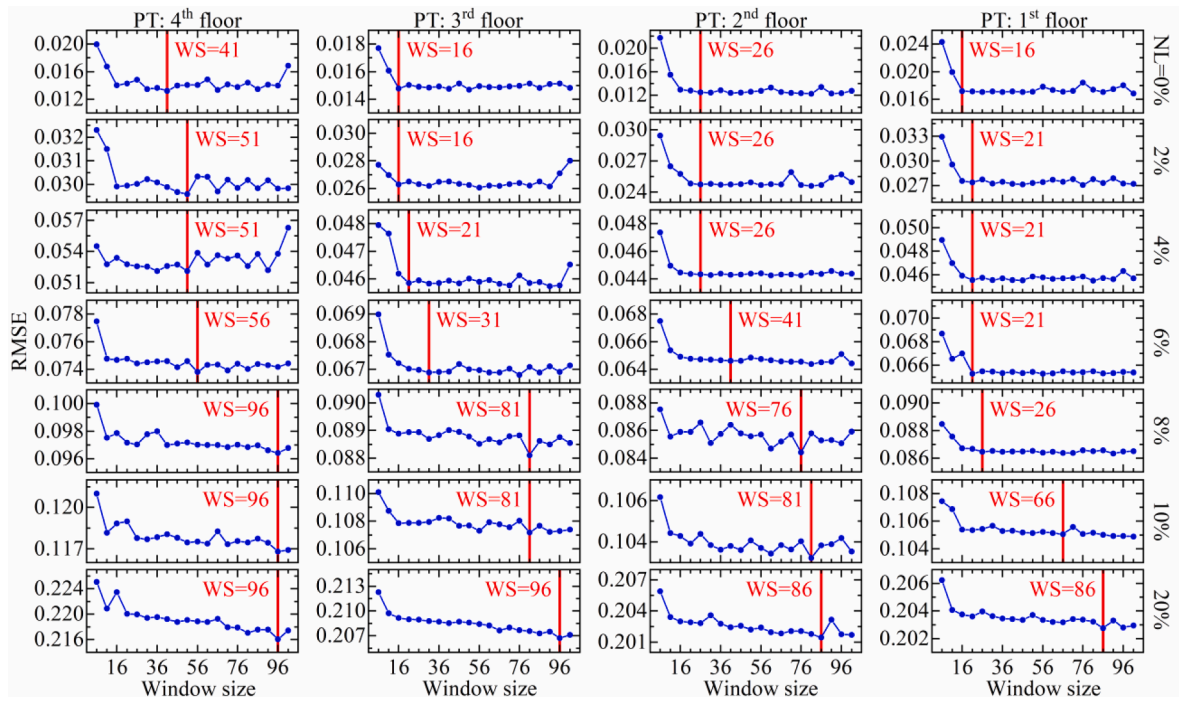

(a)   Plan view at 0.9m high   (b)   Plan view at 1.8m high   (c)   Plan view at 2.7m high   Plan view at 3.6m high

Fig. 6. Sensor layout at each story (taken from [30]).

Table 3
Details for the generated data fragments.

Type	Undamaged	Damaged							
		DP1	DP2	DP3	DP4	DP5	DP6		
Usage	Training	Validation	Testing	Testing	Testing	Testing	Testing	Testing	Testing
Duration	3600 s × 20 Hz								

Fig. 7. Optimal window size for the first loading case (load at roof) (PT: prediction target; NL: noise level; WS: window size).

stops decreasing, as shown in Fig. 7.

As illustrated in Fig. 8, for the second loading case (loading at all stories), the optimal window size is also increasing with increasing noise levels. The only difference is that the range of window size is wider than the first loading case, i.e., when the noise level is low the optimal window size is smaller, and when the noise level is high the optimal window size is larger.

5. Damage detection with optimally selected sensors

5.1. Loading case 1: Loading at the roof level

For this loading case, with acceleration response from different floors as the prediction target, the CIFs under various noise levels are illustrated in Fig. 9. The black squares represent the positions of pruned channels. The red frames represent the floors whose acceleration response is considered as the prediction target. The CRR (channel reduction ratio) is defined as the ratio of pruned channels to all candidate channels. As can be observed, the CRRs are around 60 % for most of the cases, indicating that the number of channels is largely reduced. The last column in Fig. 9 represents the optimal sensor positions calculated by EFI. It can be revealed that the EFI endeavors to capture response from every dynamic degree-of-freedoms, leading to a CRR of 25 %. The proposed method, however, exploits the correlation between the degree-of-freedoms, leading to higher CRRs compared to EFI.

In Fig. 9, the number of pruned channels is decreasing with increasing noise levels, indicating that more sensors are automatically selected by the algorithm to handle the information loss caused by higher noise levels. For the cases where acceleration response from the first, second and third floors is considered as the prediction target, the selected sensors are mostly located at the fourth floor where the excitation is provided. This indicates that the algorithm can intelligently select the channels that contain original information, instead of channels located at other floors whose data is merely a result of the excitation and therefore is redundant. However, for the case where acceleration response from the fourth floor is considered as the prediction target, selected sensors are mostly located at the third floor, revealing that the geometric information obtained from the adjacent floor is considered more important by the algorithm than the historical information from the predicted fourth floor.
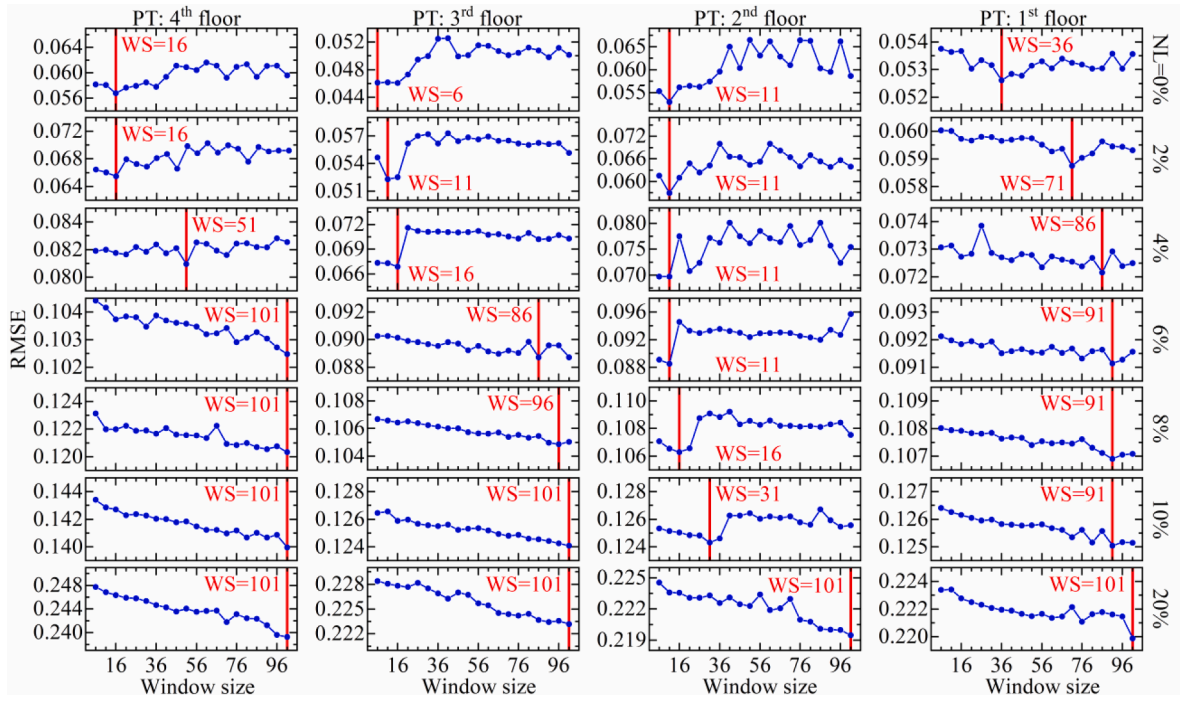
Fig. 8. Optimal window size for the second loading case (load at all stories) (PT: prediction target; NL: noise level; WS: window size).
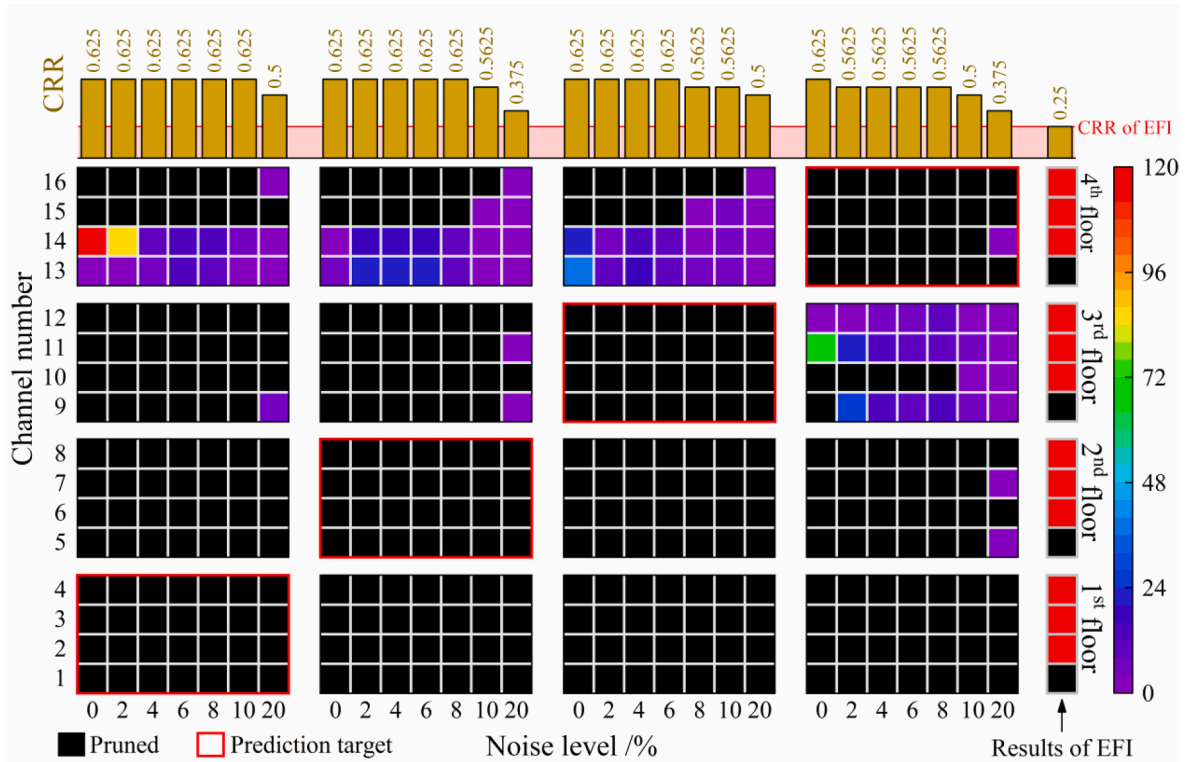

Fig. 9. Channel importance factors (CIFs) calculated with different floor response as the prediction target under excitation at roof. (CRR: channel reduction ratio).

The MADs and MFADs calculated for the first loading case are presented in Tables 4-7. For comparison, the damage patterns are sorted in an ascending order according to the damage severity. As can be seen, for each noise level, the value of MAD is positively related to the damage severity, indicating that the algorithm can accurately identify the damage level. For each damage pattern, the value of MAD is inversely related to the noise level, indicating that the damage level is under-estimated since part of the information is distorted by the noise. The value of MFAD/MAD is far less than 1 indicating that the algorithm can accurately identify the damage existence without triggering false alarms. However, when acceleration response from lower floors is considered as the prediction target, the maximum value of MFAD/MAD is increasing and the mean value of MAD is decreasing, revealing that to improve algorithm performance the response with higher SNRs is preferred for the prediction target. So that the channel with lower SNRs can be automatically pruned during training.

5.2. Loading case 2: Loading at all stories

For the second loading case, with acceleration response from different floors as the prediction target, the CIFs under various noise levels are illustrated in Fig. 10. As can be observed, the number of selected channels is also increasing with the increasing noise level. But most of the CRR here is lower compared with the results obtained from the first loading case, indicating that more channels are selected by the algorithm to receive information from multiple excitation sources. By comparing the distribution of pruned channels calculated from those two excitation cases, it is also revealed that the optimal channels and sensor locations are related to the loading pattern. However, for the EFI method, the optimal sensor positions calculated in this case are still the same as in the loading case 1 in spite of the change in excitation forms and noise levels. The CRR calculated by the EFI is still lower than the CRRs calculated by the proposed method.

The MADs and MFADs calculated for the second loading case are presented in Tables 8-11. Similar to the first loading case, the MAD is positively related to the damage severity and inversely related to the noise level. As compared with the results obtained from the first loading case, the mean values of MAD here decrease more dramatically with the rise of the noise level, and the mean values of MFAD/MAD are higher. This suggests that the noise-resistance capability of the damage detection algorithm is reduced when more excitation sources are applied to the structure. Another important observation is that the mean values of MAD are among the lowest when the first-floor response is considered as the prediction target, and the value of MFAD/MAD in this case exceeds 1 for DP6 with 20 % noise level, which emphasizes again the importance of maintaining high SNRs for channels that considered as the prediction target.

In general, the CRR, which represents the percentage of the sensors that could be removed, of the proposed method are higher than the conventional EFI method, with a highest of 62.5 % for the loading case 1 (Fig. 9) under low noise levels. In the meantime, the occurrence and the level of damage can still be well detected with a reduced number of sensors. The code for the simulated benchmark testing case 1 is on: https://github.com/tmshisheng/OSP_BenchmarkResults_Case1. The code for the testing case 2 is on: https://github.com/tmshisheng/OSP_BenchmarkResults_Case2.

5.3. Optimal sensor positions after damage

The optimal sensor positions calculated by the proposed method and conventional EFI method under pre- and post-damage scenarios are investigated in this section. The CIFs for the case 1 and case 2 after damage pattern 1 are presented in Fig. 11 and Fig. 12 respectively. Compared with the results under undamaged scenarios, the optimal sensor positions under damaged scenarios calculated by the proposed method are adaptively changed. This is because that the proposed method is purely data driven. However, the results calculated by the conventional model-driven EFI method are still the same as the undamaged scenarios, since no additional dynamic degree of freedom is created by the damage pattern 1.

Table 4
MAD and MFAD calculated with the 1st floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP2	19.99	19.97	19.77	19.36	18.86	18.39	14.95	18.75	0.01	0.05	0.02	0.01	0.00	0.00	0.03
DP1	19.99	19.96	19.60	18.93	18.11	17.18	12.24	18.00	0.01	0.05	0.02	0.01	0.00	0.00	0.04
DP4	19.85	19.68	16.44	13.22	11.29	8.77	3.27	13.22	0.01	0.05	0.03	0.02	0.00	0.01	0.14
DP5	19.85	19.69	16.66	13.26	11.35	8.79	3.56	13.31	0.01	0.05	0.03	0.02	0.00	0.01	0.12
DP3	19.84	19.60	15.29	12.40	10.13	7.55	2.77	12.51	0.01	0.05	0.03	0.02	0.00	0.01	0.16
DP6	14.80	9.56	3.00	1.07	0.54	0.29	0.46	4.25	0.01	0.11	0.15	0.22	0.05	0.24	0.96
mean	19.05	18.08	15.13	13.04	11.71	10.16	6.21	/	0.01	0.06	0.05	0.05	0.01	0.05	0.24

Table 5
MAD and MFAD calculated with the 2nd floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP2	19.99	19.97	19.81	19.50	18.99	18.51	16.15	18.99	0.02	0.01	0.01	0.01	0.01	0.01	0.03
DP1	19.97	19.91	19.61	19.12	18.36	17.39	13.64	18.29	0.02	0.01	0.01	0.01	0.01	0.01	0.04
DP4	19.91	19.74	18.69	17.06	15.87	14.20	9.95	16.49	0.02	0.01	0.01	0.01	0.01	0.01	0.05
DP5	19.91	19.75	18.78	17.09	15.85	14.17	10.34	16.56	0.02	0.01	0.01	0.01	0.01	0.01	0.05
DP3	19.88	19.56	18.31	16.52	15.06	13.47	9.62	16.06	0.02	0.01	0.01	0.01	0.01	0.01	0.05
DP6	17.42	16.22	8.93	5.95	3.97	2.95	1.27	8.10	0.02	0.02	0.01	0.02	0.04	0.05	0.38
mean	19.52	19.19	17.35	15.87	14.68	13.45	10.16	/	0.02	0.02	0.01	0.01	0.01	0.02	0.10

Table 6
MAD and MFAD calculated with the 3rd floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP2	20.00	19.95	19.79	19.44	18.52	17.39	14.41	18.50	0.03	0.01	0.02	0.04	0.02	0.01	0.01
DP1	19.98	19.82	19.49	18.87	17.27	15.73	11.60	17.54	0.03	0.01	0.02	0.04	0.02	0.01	0.01
DP4	20.00	19.90	19.26	18.91	17.15	15.19	9.65	17.15	0.03	0.01	0.02	0.04	0.02	0.01	0.01
DP5	19.99	19.90	19.28	18.89	17.17	15.16	10.00	17.20	0.03	0.01	0.02	0.04	0.02	0.01	0.01
DP3	19.99	19.85	18.91	18.57	16.45	14.41	9.15	16.76	0.03	0.01	0.02	0.04	0.02	0.01	0.02
DP6	19.63	16.95	13.12	11.23	7.56	4.02	0.60	10.44	0.03	0.01	0.03	0.07	0.05	0.03	0.23
mean	19.93	19.40	18.31	17.65	15.69	13.65	9.24	/	0.03	0.01	0.02	0.05	0.03	0.01	0.05

Table 7
MAD and MFAD calculated with the 4th floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP2	20.00	19.81	19.23	18.03	17.08	16.09	12.09	17.47	0.01	0.02	0.02	0.01	0.03	0.02	0.00
DP1	20.00	19.79	19.00	17.60	16.31	14.63	9.73	16.72	0.01	0.02	0.02	0.01	0.03	0.02	0.00
DP4	19.99	19.69	18.74	17.73	15.78	14.68	8.14	16.39	0.01	0.02	0.02	0.01	0.03	0.02	0.01
DP5	19.99	19.71	18.83	17.72	15.90	14.73	8.41	16.47	0.01	0.02	0.02	0.01	0.03	0.02	0.00
DP3	19.98	19.38	18.26	17.23	14.84	13.62	6.98	15.76	0.01	0.02	0.02	0.01	0.03	0.02	0.01
DP6	19.81	16.88	12.38	8.46	4.56	2.49	0.27	9.27	0.01	0.02	0.02	0.02	0.11	0.12	0.15
mean	19.96	19.21	17.74	16.13	14.08	12.71	7.60	/	0.01	0.02	0.02	0.01	0.05	0.04	0.03

6. Verification using the experimental dataset

6.1. Experimental dataset from shake table tests

The experimental dataset is obtained by shake table tests on a three-story reinforced concrete frame (Fig. 12a), which are conducted by researchers from the University of California at San Diego, Standford University and the University of Colorado at Boulder [78]. The
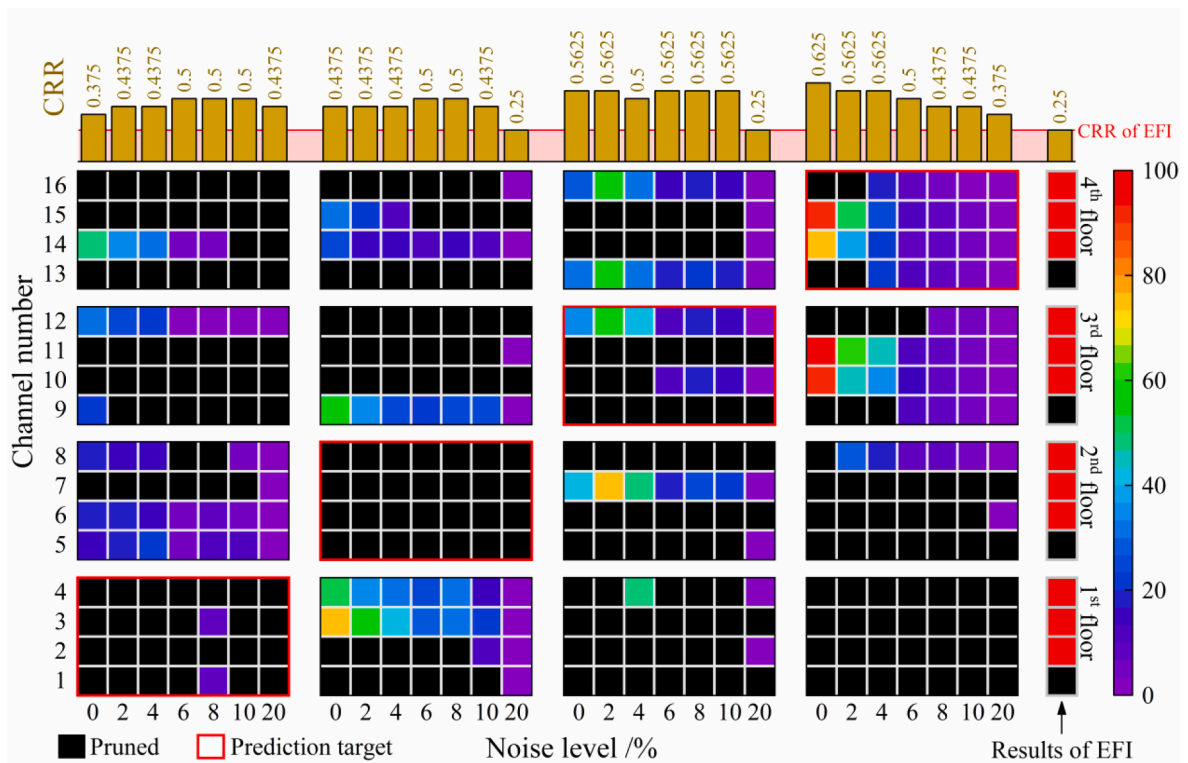
Fig. 10. Channel importance factors (CIFs) calculated with different floor response as the prediction target under excitation at all stories.

Table 8
MAD and MFAD calculated with the 1st floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP1	19.65	17.54	13.38	6.84	5.12	2.52	1.00	0.02	0.02	0.01	0.02	0.00	0.02	0.08	0.25
DP2	19.46	16.33	11.53	5.48	4.11	2.19	0.77	0.02	0.02	0.01	0.02	0.00	0.03	0.09	0.32
DP3	19.33	15.80	10.81	5.15	3.32	2.03	0.67	0.02	0.02	0.01	0.02	0.01	0.03	0.10	0.37
DP4	17.91	12.76	8.26	3.19	1.77	1.10	0.45	0.02	0.02	0.01	0.03	0.01	0.06	0.18	0.56
DP5	15.80	9.59	4.96	1.10	0.65	0.46	0.32	0.02	0.02	0.01	0.05	0.02	0.16	0.43	0.78
DP6	6.71	0.64	0.33	0.04	0.11	0.20	0.24	0.04	0.04	0.14	0.80	0.63	0.96	0.96	1.03
mean	16.48	12.11	8.21	3.63	2.52	1.42	0.57	/	0.02	0.03	0.16	0.11	0.21	0.31	0.55

dataset is available on: https://www.designsafe-ci.org/data/browser/public/nees.public/NEES-2007–0422.groups/Experiment-68.

The test frame was excited by Gilroy earthquake waveform with incremental intensities. Before and after each ground-motion test, a white-noise test was conducted to capture the change of modal parameters. The mean percentage change of the first two modal frequencies and damping ratios are calculated by the stochastic subspace identification (SSI) and presented in Table 12. It can be seen that with the increase of earthquake intensity, mean percentage change of modal frequency and damping ratio are both increasing, indicating the accumulation of damage.

With the results in Table 12 as the ground truth, the performance of proposed NANNAM is tested and verified in this section. Firstly, using the recorded acceleration response obtained during the white-noise test before the test frame is damaged, the value of RMSEs corresponding to each widow size are calculated by the reduced NANNAM, as shown in Fig. 13. The optimal window size is then selected as 126 samples since the trend of RMSE stops decreasing at this point.

Table 9
MAD and MFAD calculated with the 2nd floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP1	19.43	18.13	16.14	10.18	8.46	7.81	3.32	0.02	0.02	0.01	0.05	0.00	0.00	0.02	0.07
DP2	19.12	17.36	14.99	9.10	7.55	6.73	2.68	0.02	0.02	0.01	0.05	0.00	0.01	0.02	0.09
DP3	18.90	16.78	14.27	8.69	6.69	5.88	2.21	0.02	0.02	0.01	0.05	0.00	0.01	0.02	0.11
DP4	17.32	13.71	10.23	3.86	2.49	1.81	0.63	0.02	0.02	0.01	0.07	0.00	0.02	0.07	0.38
DP5	14.98	10.26	6.30	1.14	0.69	0.63	0.39	0.03	0.03	0.01	0.12	0.01	0.06	0.20	0.60
DP6	5.09	0.87	0.81	0.03	0.05	0.13	0.24	0.08	0.08	0.14	0.91	0.55	0.83	0.94	0.99
mean	15.81	12.85	10.46	5.50	4.32	3.83	1.58	/	0.03	0.03	0.21	0.10	0.15	0.21	0.37

Table 10
MAD and MFAD calculated with the 3rd floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP1	18.51	17.80	14.98	13.30	12.58	10.03	3.64	0.01	0.01	0.02	0.00	0.01	0.04	0.01	0.01
DP2	18.33	17.53	14.43	12.91	12.23	9.63	3.22	0.01	0.01	0.02	0.00	0.01	0.04	0.01	0.01
DP3	17.86	16.73	13.10	11.29	10.04	7.43	1.95	0.01	0.01	0.02	0.00	0.01	0.04	0.02	0.01
DP4	16.26	14.18	9.18	6.29	5.27	2.71	0.44	0.01	0.01	0.02	0.00	0.03	0.09	0.05	0.06
DP5	13.80	10.99	4.99	2.64	2.36	0.94	0.18	0.01	0.01	0.03	0.00	0.06	0.19	0.13	0.15
DP6	3.78	1.26	0.09	0.19	0.47	0.14	0.04	0.04	0.04	0.25	0.14	0.87	0.97	0.87	0.72
mean	14.76	13.08	9.46	7.77	7.16	5.15	1.58	/	0.01	0.06	0.02	0.17	0.23	0.18	0.16

Table 11
MAD and MFAD calculated with the 4th floor response as the prediction target.

	MAD								MFAD/MAD						
	0%	2%	4%	6%	8%	10%	20%	mean	0%	2%	4%	6%	8%	10%	20%
DP1	19.11	17.42	16.36	13.56	11.44	9.34	6.72	0.01	0.01	0.00	0.04	0.02	0.02	0.02	0.08
DP2	18.99	17.16	15.84	12.89	10.84	8.61	5.95	0.01	0.01	0.00	0.04	0.02	0.02	0.02	0.09
DP3	18.23	16.25	14.71	11.61	8.92	6.83	4.59	0.01	0.01	0.00	0.04	0.02	0.02	0.03	0.12
DP4	16.67	13.70	11.40	7.16	4.61	2.63	1.85	0.01	0.01	0.01	0.05	0.04	0.04	0.07	0.30
DP5	14.48	10.44	7.71	3.47	1.87	0.93	1.19	0.01	0.01	0.01	0.08	0.07	0.10	0.19	0.46
DP6	4.64	0.94	0.89	0.34	0.22	0.18	0.62	0.04	0.04	0.08	0.66	0.75	0.86	0.95	0.88
mean	15.36	12.65	11.15	8.17	6.32	4.75	3.49	/	0.01	0.02	0.15	0.15	0.18	0.21	0.32

6.2. Damage detection with optimally selected sensors

The acceleration response from the 3rd floor is taken as the prediction target, and the acceleration response from all floors are taken as the input. After the NANNAM is trained with the data obtained from the white-noise test on the undamaged test frame, unnecessary channels are automatically pruned. The optimal sensor placement for each floor is presented in Fig. 14b through 14d. The CRRs of the first and second floors are 81.82 % and 66.67 %, respectively. The overall CRR is 54.84 %, indicating that the number of channels is
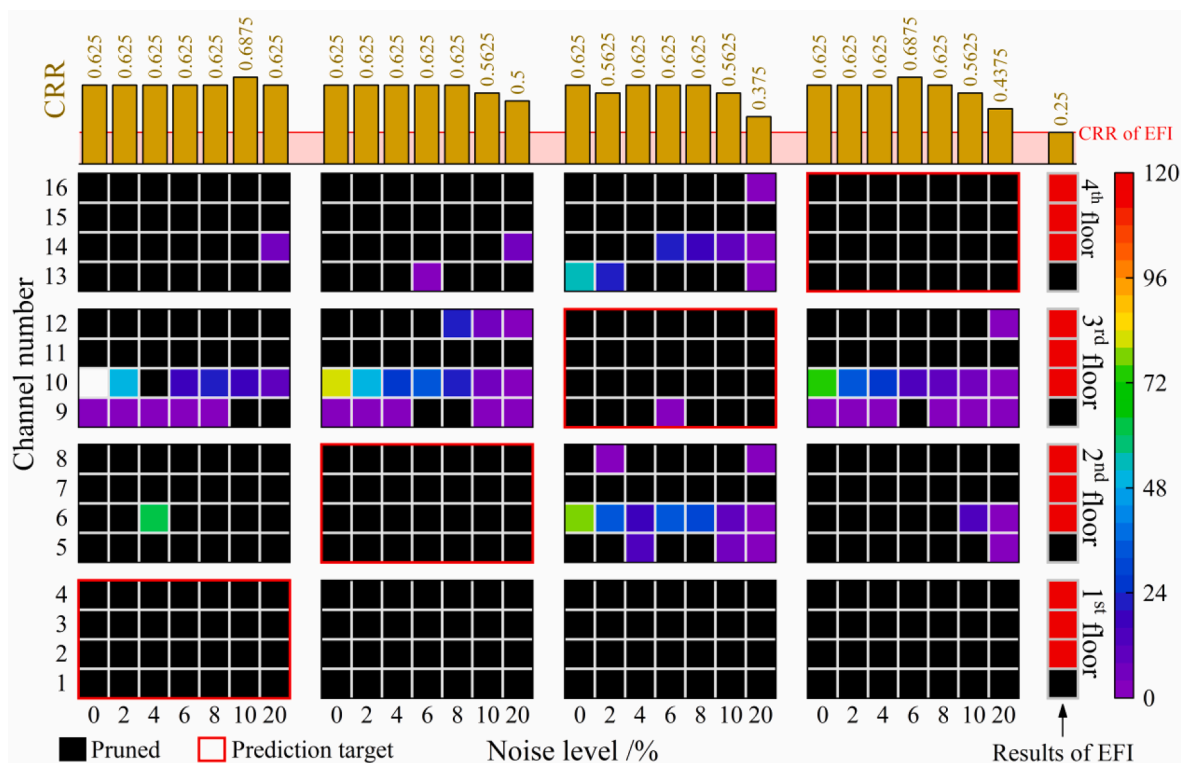
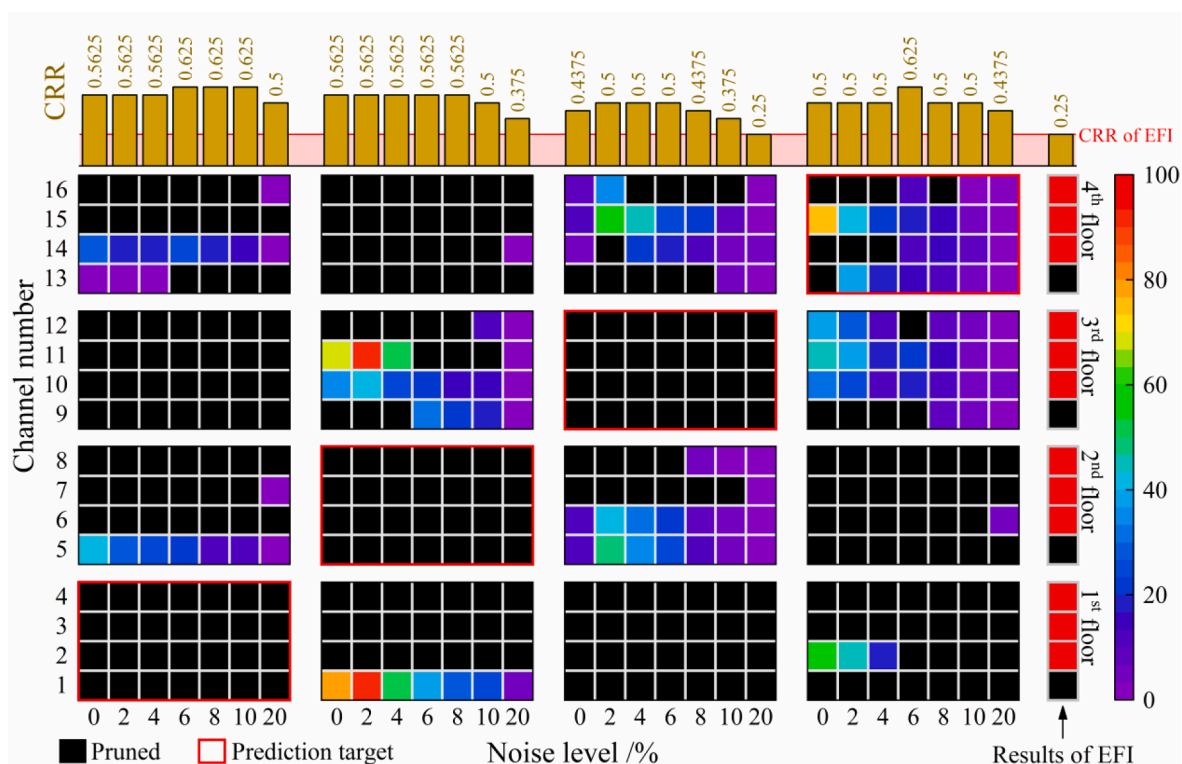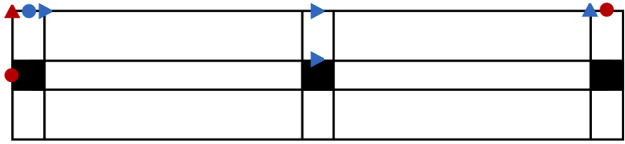Fig. 11. Channel importance factors (CIFs) for the case 1 after damage pattern 1.

Fig. 12. Channel importance factors (CIFs) for the case 2 after damage pattern 1.

Table 12
Mean percentage change of the first two modal frequencies and damping ratios after ground motions with incremental intensities.

Earthquake intensity	Longitudinal frequency	Longitudinal damping ratio	Rotational frequency	Rotational damping ratio
20 % of Gilroy	0.1 %	14.5 %	0.8 %	9.1 %
40 % of Gilroy	0.8 %	7.0 %	0.4 %	11.2 %
67 % of Gilroy	5.1 %	46.1 %	2.2 %	18.2 %
83 % of Gilroy	16.0 %	252.5 %	7.9 %	45.5 %
91 % of Gilroy	23.8 %	432.3 %	15.2 %	54.5 %
100 % of Gilroy	43.4 %	510.7 %	17.3 %	45.5 %
120 % of Gilroy	57.9 %	480.9 %	21.5 %	76.1 %
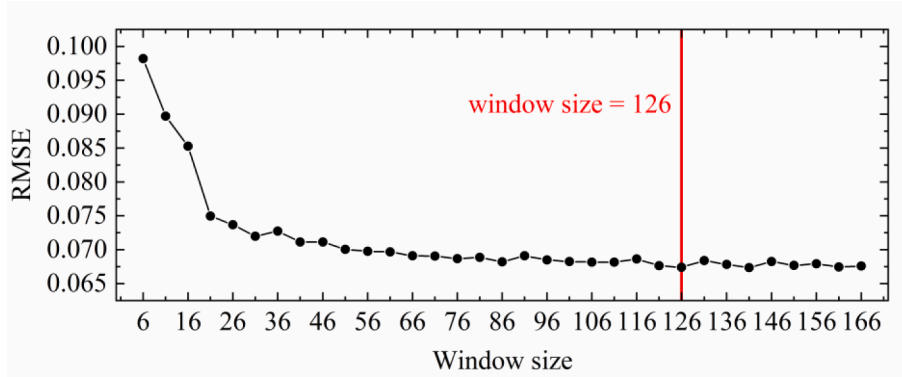
Fig. 13. Optimal widow size for the experimental dataset.

(d) Location of selected sensors on the top of the 3rd floor.
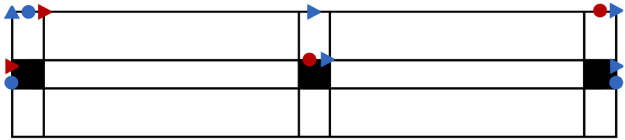
(c) Location of selected sensors on the top of the 2nd floor.

(a) The testing frame (reproduced from [78]).
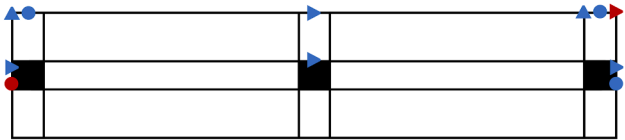
(b) Location of selected sensors on the top of the 1st floor.

▲ Selected accelerometer in Y-direction ● Selected accelerometer in Z-direction ▶ Selected accelerometer in X-direction

▲ Pruned accelerometer in Y-direction ● Pruned accelerometer in Z-direction ▶ Pruned accelerometer in X-direction

Fig. 14. The testing frame and the optimal sensor placement.

largely reduced.

After trained with data obtained from the undamaged test frame, the NANNAM is then tested on the data obtained from the test frame damaged by ground motions with incremental intensity. As shown in Table 13, the calculated MAD is increasing with increasing earthquake intensity, suggesting that the damage level is accurately identified. However, for the case when intensity is 40 % of Gilroy, the MAD is underestimated due to the imperfection of real-world data. The value of MFAD/MAD is less than 1 and is decreasing with

Table 13
MAD and MFAD after excited by ground motions with incremental intensity.

Earthquake intensity	10% of Gilroy	20% of Gilroy	40% of Gilroy	67% of Gilroy	83% of Gilroy	91% of Gilroy	100% of Gilroy	120% of Gilroy
MAD	43.69	71.21	56.67	139.69	166.29	206.99	207.82	230.98
MFAD/MAD	0.3687	0.2262	0.2843	0.1153	0.0969	0.0778	0.0775	0.0697

increasing level of damage, indicating that the damage occurrence can be accurately identified and the chance of sending false alarms is lower for severer damage cases.

7. Discussion

With the dataset simulated from the IASC-ASCE benchmark and the experimental dataset obtained from the shake table tests, the performance and effectiveness of the proposed method are investigated. It is found that:

1) During the tests on the benchmark dataset, the proposed method is tested under two excitation forms and seven noise levels. It is observed that the optimal sensor positions are automatically adapting to the excitation forms and noise levels as expected. For the first loading case when the excitation is applied only at the top floor, at most 62.5 % of the candidate sensor positions are excluded and the optimal sensor positions are gathered around the top floor no matter the selection of prediction target. For the second loading case when the excitation is applied at all the floors, at most 62.5 % of the candidate sensor positions are excluded and the optimal sensor positions are distributed over all the floors.
2) For all cases, the percentage of removed sensors calculated by the proposed method are all higher than the conventional effective independence (EFI) method. Compared with EFI, the results obtained from the proposed method is observed adaptive to excitation forms and noise levels. These observations confirmed the advantages of the proposed data-driven method.
3) During the tests on the benchmark dataset, it is also observed that the optimal sensor positions are automatically adapting to the noise levels. With increasing noise levels, the required number of sensors are increasing, which indicates that the algorithm is using data from more sensors to counter the information loss caused by the noise.
4) After the sensor positions are optimized, the results of MAD (mean alarm density), which is used as a damage indicator, for benchmark dataset show that the occurrence and level of damage can be successfully identified for each loading case and noise level. The results of MFAD (mean false alarm density) show that the damage can be detected without false alarms except for one case when the damage level is low, and the noise level is high.
5) During the tests on the experimental dataset, it is observed that 54.84 % of the candidate sensor positions are excluded. The calculated MAD increases with increasing damage level introduced by exciting the test frame with growing input earthquake intensities, indicating that the occurrence and level of damage can be accurately identified using data from optimized sensor positions. The value of MFAD/MAD remains low for all cases and is reducing with the increase of damage level, indicating that the algorithm is more confident about the damage detection results for higher levels of damage.

8. Conclusions

Optimization of sensor placement (OSP) is one of the important issues in structural health monitoring (SHM). The conventional model-driven methods often identify optimal sensor positions by using mode shapes, which means: 1) these methods are highly dependent on the accuracy of modal analysis; 2) the calculated optimal sensor positions are not adaptive to the excitation forms and noise levels; 3) the applicability of these model-driven methods to unsupervised data-driven damage detection remains questionable. To overcome these issues, a novel data-driven OSP method for unsupervised damage detection in SHM is presented. For this method, a novel neural network called NANNAM (noise-assisted neural networks with attention mechanism) along with a two-stage training algorithm is proposed to adaptively calculate the optimal sensor locations by utilizing the temporal and spatial correlations within the multi-channel acceleration response.

The noise-assisted approach allows for measuring the relative contribution of data obtained from candidate positions compared to the artificial noise. The attention mechanism forces the neural networks to pay attention to the data obtained from important positions, which helps to select optimal sensor positions in an unsupervised and data-driven manner. With the dataset simulated from the IASC-ASCE benchmark and the experimental dataset obtained from the shake table tests, the effectiveness of the proposed method is verified.

CRediT authorship contribution statement

Sheng Shi: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Dongsheng Du:** Conceptualization, Funding acquisition, Project administration, Resources, Software, Supervision, Visualization, Writing – review & editing. **Oya Mercan:** Supervision, Visualization, Writing – review & editing. **Erol Kalkan:** Supervision, Visualization, Writing – review & editing. **Shuguang Wang:** Supervision, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

References

[1] V.G.M. Annamdas, S. Bhalla, C.K. Soh, Applications of structural health monitoring technology in Asia, Struct Heal Monit 16 (2017) 324–346, https://doi.org/10.1177/1475921716653278.
[2] J. Ou, H. Li, Structural health monitoring in mainland china: Review and future trends, Struct Heal Monit 9 (2010) 219–231, https://doi.org/10.1177/1475921710365269.
[3] C.R. Farrar, K. Worden, An introduction to structural health monitoring, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 365 (2007) 303–315, https://doi.org/10.1098/rsta.2006.1928.
[4] R.J. Allemang, D.L. Brown, Correlation Coefficient for Modal Vector Analysis, Proc Int Modal Anal Conf Exhib (1982) 110–116.
[5] O.S. Salawu, C. Williams, Bridge Assessment Using Forced-Vibration Testing, J. Struct. Eng. 121 (1995) 161–173, https://doi.org/10.1061/(asce)0733-9445(1995)121:2(161).
[6] C.R. Farrar, W.E. Baker, R.C. Dove, Dynamic parameter similitude for concrete models, ACI Struct. J. 91 (1994) 90–99, https://doi.org/10.14359/4500.
[7] J. Ciambella, A. Pau, F. Vestroni, Modal curvature-based damage localization in weakly damaged continuous beams, Mech. Syst. Sig. Process. 121 (2019) 171–182, https://doi.org/10.1016/j.ymssp.2018.11.012.
[8] A.K. Pandey, M. Biswas, Damage detection in structures using changes in flexibility, J. Sound Vib. 169 (1994) 3–17, https://doi.org/10.1006/jsvi.1994.1002.
[9] Z.Y. Shi, S.S. Law, L.M. Zhang, Structural damage localization from modal strain energy change, J. Sound Vib. 218 (1998) 825–844, https://doi.org/10.1006/jsvi.1998.1878.
[10] N.H. Farhat, Photonit neural networks and learning mathines the role of electron-trapping materials, IEEE Expert Syst Their Appl 7 (1992) 63–72, https://doi.org/10.1109/64.163674.
[11] G. Gui, H. Pan, Z. Lin, Y. Li, Z. Yuan, Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection, KSCE J. Civ. Eng. 21 (2017) 523–534, https://doi.org/10.1007/s12205-017-1518-5.
[12] Y. Bao, C. Song, W. Wang, T. Ye, L. Wang, L. Yu, Damage detection of bridge structure based on SVM, Math. Probl. Eng. (2013; 2013.), https://doi.org/10.1155/2013/490372.
[13] H. Pan, M. Azimi, F. Yan, Z. Lin, Time-Frequency-Based Data-Driven Structural Diagnosis and Damage Detection for Cable-Stayed Bridges, J. Bridg. Eng. 23 (2018) 04018033, https://doi.org/10.1061/(asce)be.1943-5592.0001199.
[14] L.S. Huo, X. Li, Y.Y. Bin, H.N. Li, Damage detection of structures for ambient loading based on cross correlation function amplitude and SVM, Shock Vib. (2016; 2016.), https://doi.org/10.1155/2016/3989743.
[15] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32, https://doi.org/10.1023/A:1010933404324.
[16] Chun P jo, Ujike I, Mishima K, Kusumoto M, Okazaki S. Random forest-based evaluation technique for internal damage in reinforced concrete featuring multiple nondestructive testing results. Constr. Build. Mater. 2020;253:119238. doi: 10.1016/j.conbuildmat.2020.119238.
[17] D.K. Lim, K.B. Mustapha, C.P. Pagwiwoko, Delamination detection in composite plates using random forests, Compos. Struct. 278 (2021) 114676, https://doi.org/10.1016/j.compstruct.2021.114676.
[18] X. Bao, Z. Wang, D. Fu, C. Shi, G. Iglesias, H. Cui, et al., Machine learning methods for damage detection of thermoplastic composite pipes under noise conditions, Ocean Eng. 248 (2022) 110817, https://doi.org/10.1016/j.oceaneng.2022.110817.
[19] S.B. Beheshti Aval, V. Ahmadian, M. Maldar, E. Darvishan, Damage detection of structures using signal processing and artificial neural networks, Adv. Struct. Eng. 23 (2020) 884–897, https://doi.org/10.1177/1369433219886079.
[20] S.F. Jiang, C.M. Zhang, S. Zhang, Two-stage structural damage detection using fuzzy neural networks and data fusion techniques, Expert Syst. Appl. 38 (2011) 511–519, https://doi.org/10.1016/j.eswa.2010.06.093.
[21] U. Dackermann, J. Li, B. Samali, Identification of member connectivity and mass changes on a two-storey framed structure using frequency response functions and artificial neural networks, J. Sound Vib. 332 (2013) 3636–3653, https://doi.org/10.1016/j.jsv.2013.02.018.
[22] N. Bakhary, H. Hao, A.J. Deeks, Structure damage detection using neural network with multi-stage substructuring, Adv. Struct. Eng. 13 (2010) 95–110.
[23] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, D.J. Inman, Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks, J. Sound Vib. 388 (2017) 154–170, https://doi.org/10.1016/j.jsv.2016.10.043.
[24] O. Abdeljaber, O. Avci, M.S. Kiranyaz, B. Boashash, H. Sodano, D.J. Inman, 1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data, Neurocomputing 275 (2018) 1308–1317, https://doi.org/10.1016/j.neucom.2017.09.069.
[25] H. Khodabandehlou, G. Pekcan, M.S. Fadali, Vibration-based structural condition assessment using convolution neural networks, Struct Control Heal Monit 26 (2019) 1–12, https://doi.org/10.1002/stc.2308.
[26] M. Azimi, G. Pekcan, Structural health monitoring using extremely compressed data through deep learning, Comput Civ Infrastruct Eng 35 (2020) 597–614, https://doi.org/10.1111/mice.12517.
[27] S.O. Sajedi, X. Liang, Vibration-based semantic damage segmentation for large-scale structural health monitoring, Comput Civ Infrastruct Eng 35 (2020) 579–596, https://doi.org/10.1111/mice.12523.
[28] T. Lyu, C. Xu, G. Chen, Q. Li, T. Zhao, Y. Zhao, Health state inversion of Jack-up structure based on feature learning of damage information, Eng. Struct. 186 (2019) 131–145, https://doi.org/10.1016/j.engstruct.2019.02.004.
[29] Z. Ding, J. Li, H. Hao, Structural damage identification by sparse deep belief network using uncertain and limited data, Struct Control Heal Monit 27 (2020) 1–20, https://doi.org/10.1002/stc.2522.
[30] S. Shi, D. Du, O. Mercan, E. Kalkan, S. Wang, A novel unsupervised real-time damage detection method for structural health monitoring using machine learning, Struct Control Heal Monit (2022) 1–26, https://doi.org/10.1002/stc.3042.
[31] Y.J. Cha, Z. Wang, Unsupervised novelty detection–based structural damage localization using a density peaks-based fast clustering algorithm, Struct Heal Monit 17 (2018) 313–324, https://doi.org/10.1177/1475921717691260.
[32] O. Avci, O. Abdeljaber, Self-Organizing Maps for Structural Damage Detection: A Novel Unsupervised Vibration-Based Algorithm, J. Perform. Constr. Facil 30 (2016) 04015043, https://doi.org/10.1061/(asce)cf.1943-5509.0000801.
[33] M.H. Rafiei, H. Adeli, A novel unsupervised deep learning model for global and local health condition assessment of structures, Eng. Struct. 156 (2018) 598–607, https://doi.org/10.1016/j.engstruct.2017.10.070.
[34] A.C. Neves, I. González, J. Leander, R. Karoumi, Structural health monitoring of bridges: a model-free ANN-based approach to damage detection, J. Civ. Struct. Heal. Monit. 7 (2017) 689–702, https://doi.org/10.1007/s13349-017-0252-5.

[35] M.L. Fugate, H. Sohn, C.R. Farrar, Vibration-based damage detection using statistical process control, Mech. Syst. Sig. Process. 15 (2001) 707–721, https://doi.org/10.1006/mssp.2000.1323.

[36] K.A. Eltouny, X. Liang, Bayesian-optimized unsupervised learning approach for structural damage detection, Comput Civ Infrastruct Eng 36 (2021) 1249–1269, https://doi.org/10.1111/mice.12680.

[37] Y. Tan, L. Zhang, Computational methodologies for optimal sensor placement in structural health monitoring: A review, Struct Heal Monit 19 (2020) 1287–1308, https://doi.org/10.1177/1475921719877579.

[38] W. Ostachowicz, R. Soman, P. Malinowski, Optimization of sensor placement for structural health monitoring: a review, Struct Heal Monit 18 (2019) 963–988, https://doi.org/10.1177/1475921719825601.

[39] L. Yao, W.A. Sethares, D.C. Kammer, Sensor placement for on-orbit modal identification of large space structure via a genetic algorithm, Proc IEEE Int Conf Syst Eng (1992) 332–335, https://doi.org/10.1109/ICSYSE.1992.236889.

[40] D.C. Kammer, Sensor placement for on-orbit modal identification and correlation of large space structures, J Guid Control Dyn 14 (1991) 251–259, https://doi.org/10.2514/3.20635.

[41] G. Heo, M.L. Wang, D. Satpathi, Optimal transducer placement for health monitoring of long span bridge, Soil Dyn. Earthq. Eng. 16 (1997) 495–502, https://doi.org/10.1016/S0267-7261(97)00010-9.

[42] C. Papadimitriou, Optimal sensor placement methodology for parametric identification of structural systems, J. Sound Vib. 278 (2004) 923–947, https://doi.org/10.1016/j.jsv.2003.10.063.

[43] H. Sun, O. Büyüköztürk, Optimal sensor placement in structural health monitoring using discrete optimization, Smart Mater. Struct. 24 (2015) 125034, https://doi.org/10.1088/0964-1726/24/12/125034.

[44] C. Yang, Y. Xia, A novel two-step strategy of non-probabilistic multi-objective optimization for load-dependent sensor placement with interval uncertainties, Mech. Syst. Sig. Process. 176 (2022) 109173, https://doi.org/10.1016/j.ymssp.2022.109173.

[45] M. Civera, M.L. Pecorelli, R. Ceravolo, C. Surace, F.L. Zanotti, A multi-objective genetic algorithm strategy for robust optimal sensor placement, Comput Civ Infrastruct Eng 36 (2021) 1185–1202, https://doi.org/10.1111/mice.12646.

[46] A. Spielberg, A. Amini, L. Chin, W. Matusik, D. Rus, Co-learning of task and sensor placement for soft robotics, IEEE Robot Autom Lett 6 (2021) 1208–1215, https://doi.org/10.1109/LRA.2021.3056369.

[47] P. Sashittal, D.J. Bodony, Data-driven sensor placement for fluid flows, Theor. Comput. Fluid Dyn. 35 (2021) 709–729, https://doi.org/10.1007/s00162-021-00584-w.

[48] J. Waeytens, S. Sadr, Computer-aided placement of air quality sensors using adjoint framework and sensor features to localize indoor source emission, Build. Environ. 144 (2018) 184–193, https://doi.org/10.1016/j.buildenv.2018.08.012.

[49] J.P. Argaud, B. Bouriquet, F. de Caso, H. Gong, Y. Maday, O. Mula, Sensor placement in nuclear reactors based on the generalized empirical interpolation method, J. Comput. Phys. 363 (2018) 354–370, https://doi.org/10.1016/j.jcp.2018.02.050.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., PyTorch: An imperative style, high-performance deep learning library, Adv Neural Inf Process Syst 32 (2019).

[51] G. Lee, K. Lee, DNN compression by ADMM-based joint pruning, Knowledge-Based Syst 239 (2022) 107988, https://doi.org/10.1016/j.knosys.2021.107988.

[52] S. Anwar, K. Hwang, W. Sung, Structured Pruning of Deep Convolutional Neural Networks, Am. J. Surg. Pathol. 21 (2015) 399–406, https://doi.org/10.48550/arXiv.1512.08571.

[53] W. Wen, C. Wu, Y. Wang, Y. Chen, H. Li, Learning structured sparsity in deep neural networks, Adv Neural Inf Process Syst (2016) 2082–2090.

[54] Han S, Pool J, Tran J, Dally WJ. Learning both weights and connections for efficient neural networks. Adv Neural Inf Process Syst 2015;2015-Janua:1135–43.

[55] Z. Liu, M. Sun, T. Zhou, G. Huang, T. Darrell, Rethinking the value of network pruning, in: 7th Int Conf Learn Represent ICLR, 2019, 2019,, pp. 1–21.

[56] T. Strypsteen, A. Bertrand, End-to-end learnable EEG channel selection for deep neural networks with Gumbel-softmax, J. Neural Eng. 18 (2021), https://doi.org/10.1088/1741-2552/ac115d.

[57] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning convolutional neural networks for resource efficient inference, in: 5th Int Conf Learn Represent ICLR 2017 - Conf Track Proc, 2017, pp. 1–17.

[58] C. Liu, H. Wu, Channel pruning based on mean gradient for accelerating Convolutional Neural Networks, Signal Process. 156 (2019) 84–91, https://doi.org/10.1016/j.sigpro.2018.10.019.

[59] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning Efficient Convolutional Networks through Network Slimming, Proc IEEE Int Conf Comput vis (2017; 2017-Octob:2755–63.), https://doi.org/10.1109/ICCV.2017.298.

[60] W. Wang, X. Zhang, L. Xie, J. Zhou, H. Su, B. Zhang, et al., Pruning from scratch. AAAI 2020–34th AAAI Conf, Artif. Intell. 1 (2020) 12273–12280, https://doi.org/10.1609/aaai.v34i07.6910.

[61] J.K. Kruschke, J.R. Movellan, Benefits of Gain: Speeded Learning and Minimal Hidden Layers in Back-Propagation Networks, IEEE Trans. Syst. Man Cybern. 21 (1991) 273–280, https://doi.org/10.1109/21.101159.

[62] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning Deep Features for Discriminative Localization. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2016;2016-Decem:2921–9. doi: 10.1109/CVPR.2016.319.

[63] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, in: Proc. IEEE Conf. Comput. vis, Pattern Recognit, 2018, pp. 7132–7141.

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, et al., Attention is all you need, Adv Neural Inf Process Syst (2017). 2017-Decem: 5999–6009.

[65] R. Benzi, A. Sutera, A. Vulpiani, The mechanism of stochastic resonance, J. Phys. A Math. Gen. (1981) 14, https://doi.org/10.1088/0305-4470/14/11/006.

[66] R. Benzi, G. Parisi, A. Sutera, A. Vulpiani, Stochastic resonance in climatic change, Tellus 34 (1982) 10.

[67] Wu Z, E.Huang N. Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method. Adv. Adapt. Data Anal. 2009;01:1–41. doi: 10.1142/S1793536909000047.

[68] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, N. Yen, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc R Soc A 454 (1996) 903–995.

[69] Goodfellow IJ, Pouget-abadie J, Mirza M, Xu B, Warde-farley D. Generative Adversarial Nets n.d.:1–9. doi: 10.48550/ARXIV.1406.2661.

[70] J. Liang, S.R. Chaudhuri, M. Shinozuka, Simulation of Nonstationary Stochastic Processes by Spectral Representation, J. Eng. Mech. 133 (2007) 616–627, https://doi.org/10.1061/(asce)0733-9399(2007)133:6(616).

[71] M. Shinozuka, Y. Sato, Simulation of Nonstationary Random Process, J .Eng. Mech. Div. 93 (1967) 11–40, https://doi.org/10.1061/jmcea3.0000822.

[72] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, Going deeper with convolutions., et al., IEEE Conf. Comput. Vis. Pattern Recognit., vol. 07–12-June, IEEE 2015 (2015) 1–9, https://doi.org/10.1109/CVPR.2015.7298594.

[73] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, Speech Lang Process, ICML Work Deep Learn Audio, 2013, p. 28.

[74] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Leaning internal representations by back-propagating errors, Nature 323 (1986) 318–362.

[75] S. Amari, Backpropagation and stochastic gradient descent method, Neurocomputing 5 (1993) 185–196, https://doi.org/10.1016/0925-2312(93)90006-O.

[76] E.A. Johnson, H.F. Lam, L.S. Katafygiotis, J.L. Beck, Phase I IASC-ASCE Structural Health Monitoring Benchmark Problem Using Simulated Data, J. Eng. Mech. 130 (2004) 3–15, https://doi.org/10.1061/(asce)0733-9399(2004)130:1(3).

[77] P. Stoica, Y. Selen, Model-order selection: a review of information criterion rules, IEEE Signal Process Mag. 21 (2004) 36–47.

[78] A. Stavridis, Analytical and Experimental Study of Seismic Performance of Reinforced Concrete Frames Infilled with Masonry Walls, University of California, San Diego, 2009.