


RESEARCH ARTICLE

A novel unsupervised real-time damage detection method for structural health monitoring using machine learning

Sheng Shi^{1,2}  | Dongsheng Du¹ | Oya Mercan² | Erol Kalkan³ | Shuguang Wang¹

¹Department of Civil Engineering, Nanjing Tech University, Nanjing, China

²Department of Civil and Mineral Engineering, University of Toronto, Toronto, ON, Canada

³CEO, QuakeLogic Inc., Roseville, CA, USA

Correspondence

Sheng Shi, Department of Civil Engineering, Nanjing Tech University, Nanjing, China.

Email: tmss@njtech.edu.cn; tmshisheng@outlook.com

Funding information

China Scholarship Council (CSC); National Natural Science Foundation of China, Grant/Award Number: 51808291

Summary

Real-time structural damage detection is one of the main goals of establishing an effective structural health monitoring system. However, due to the lack of training data for possible damage patterns, supervised methods tend to be difficult for such applications. This article therefore proposes a novel unsupervised real-time damage detection method using machine learning, which consists of a statistical modeling approach using neural networks and a decision-making process using deep support vector domain description. To choose an optimal window length while extracting damage-sensitive features, an iterative training strategy is proposed to remove redundant samples from an oversized window. The proposed method is then verified using a simulated dataset from the International Association for Structural Control–American Society of Civil Engineering benchmark and an experimental dataset from shake table tests. The results show that the mean alarm density can be used as an indicator of damage existence and damage levels for the single-sensor approach. Higher performance of damage detection and lower performance of identifying damage levels are observed for the multi-sensor approach when the rotational modes are amplified by asymmetric damage patterns. The results of mean false alarm density show that the presented method has a low probability of generating false alarms. The effectiveness of iterative pruning strategy is observed through the visualization of loss function and weights in the neural networks. Finally, the capability of real-time execution of the proposed damage detection method is investigated and verified. As a result, trained with healthy data only, the proposed method is effective in detecting damage existence and damage levels.

KEYWORDS

convolutional neural networks, deep support vector data description, IASC–ASCE benchmark, machine learning, structural health monitoring, unsupervised damage detection

1 | INTRODUCTION

Although structural safety and reliability have always been important considerations for civil engineering structures during their design stage, it is equally important to pay attention to them during their service life. Aging and damage

can accumulate over time, whereas partial or complete failure of structural components may occur during extreme events, such as earthquakes, windstorms, or hurricanes.¹ Fortunately, with the rapid development of techniques related to data acquisition, transmission, and storage, a considerable number of structural health monitoring (SHM) systems have been developed and implemented in the last few decades.^{2,3} The SHM applications cover bridges, high-rise buildings, tunnels, deep excavations, and so on. With the assistance of machine-learning techniques such as neural networks, it becomes possible for SHM systems to help save time and money by detecting potential damages at an early stage using a massive amount of sensor data collected from the monitored structures and without requiring human intervention.

“In the most general terms, damage can be defined as changes introduced into a system that adversely affect its current or future performance.”^{1,4,5} Hence, a basic idea of damage detection is to detect these changes that are potentially introduced by the damage. From a vibration-based damage detection perspective, these changes can be reflected by indexes deduced based on physical laws or statistics.

For the damage detection approaches established based on physical laws, the monitored features are mainly related to the modal characteristics calculated from vibration responses of structures under ambient excitation. Commonly used indexes include modal frequency, modal shape,⁶ modal damping,^{7,8} modal curvature,⁹ modal strain energy,¹⁰ and modal flexibility.¹¹ The damage can then be indicated by the change of these monitored modal characteristics. Under environmental excitation, the modal characteristics are often calculated by output-only methods,¹² where the excitation is assumed to be independent white noise. However, this assumption may not be strictly satisfied due to the complex environmental conditions, which will introduce errors to the calculated modal characteristics and possibly trigger a false alarm. Especially as minor injuries usually cannot result in drastic changes of structural dynamic responses and usually make an observable difference to the higher mode frequencies only, it will become even more difficult to detect damages with high-frequency noise introduced by the data acquisition devices. On the other hand, modal characteristics of civil infrastructures vary with the change in environmental conditions such as temperature¹³ and mass (oil storage level of offshore platforms⁴ and human occupation level of public buildings), which makes it difficult to determine a proper threshold.

Alternatively, for the damage detection approaches established based on statistics, the monitored features are mainly deduced by the statistical modeling of the vibration response of structures under ambient excitation. One of the early feasible solutions for damage detection in SHM is proposed by Sohn et al,¹⁴ where they fit the vibration response with an autoregressive (AR) model and then use the coefficients of the AR model as the damage-sensitive features. By applying a statistical process control (SPC) technique, the potential damage can be indicated when a certain number of features are outside the control limits. It can be concluded that the statistic-based damage detection approaches are mainly composed of two critical elements, namely, the statistical modeling and the decision-making criteria. The main purpose of statistical modeling is to extract low-dimensional features from the high-dimensional vibration data, where the feasible choices of feature extraction techniques include time-series analysis,^{14,15} time-frequency analysis,^{16,17} principal component analysis,¹⁸ artificial neural networks,^{19–21} and auto-encoders.^{22,23}

After feature extraction follows decision-making, where a threshold is established to decide if the extracted features are within a normal and acceptable range. Conventionally, the decision-making criteria is usually a threshold determined by SPC,^{14,15,18,24} pattern classification,^{25–27} or clustering,^{16,19} where the extracted features are categorized into a healthy or unhealthy group in a supervised or unsupervised manner. Supervised methods train decision-making models with labeled data, which makes the practical implementation difficult due to lack of anomalies in the data collected from a real structure. For a particular structure, the number of damage patterns is numerous already, let alone the combination of them. For unsupervised methods,^{22,28–31} the labeled data from possible damage patterns is no longer necessary, and the decision-making models can be trained with healthy data only. Once trained, these methods can subsequently identify abnormalities by only knowing what is normal regardless of numerous possible damage patterns. This is particularly important for real-world applications because the abnormal data are not usually available for training in many cases.

Currently, although successful SHM applications exist in areas such as monitoring rotating machinery,³² a few obstacles still remain for the monitoring of civil structures, such as

1. Different from rotating machinery, civil structures are continuously in service after construction, and a large amount of data can be continuously collected by sensors. However, with the limited computational efficiency of the conventional algorithms that use matrix manipulations, only a very small percentage of the data is used to produce new information about the monitored structures.

2. Because the monitoring data are collected continuously, it is important for SHM algorithms to make decisions in real time to ensure a timely assessment of the structural condition.
3. Because almost all the monitoring data are collected from a healthy structure, extremely few anomaly data can be obtained in most cases, which makes it challenging to identify the damage by using supervised methods. Hence, there is a need to develop an unsupervised damage detection method with healthy data only. With the rapid development of artificial intelligence (AI) in recent years, statistic-based methods demonstrate a great potential to address this need.

Based on the aforementioned considerations, this paper aims to establish an unsupervised damage detection method, which uses either single-channel or multi-channel real-time monitoring data collected from a healthy structure to develop statistical models and establish a decision-making criterion with no need for anomaly data. Although the discussions in Section 3 focus on acceleration data specifically, the data type of this framework is not restricted to acceleration data. The proposed damage detection framework sends out alarms in real time when the adaptively generated threshold is exceeded and provides information about the level of the severity of the relative damage. The proposed framework has the potential to provide a solution for mining information from the massive monitoring data collected and stored by SHM systems without requiring simulations. The proposed framework makes it possible to use the data even under a high noise level, preventing the risk of eliminating or suppressing useful information in the data during denoising. Because the proposed framework can adaptively learn patterns from real monitoring data collected from a healthy structure without human intervention, it has the potential to establish an intelligent self-learning SHM system.

The rest of the paper is organized as follows: Section 2 describes the proposed damage detection framework using single-channel and multi-channel data, respectively. Section 3 investigates and discusses the capability of these two types of approaches under various levels of noise and damage using the simulated International Association for Structural Control–American Society of Civil Engineering (IASC–ASCE) benchmark dataset. Finally, the summary, conclusions, and future work are presented in Section 4. All of the algorithms investigated in this paper are implemented using Pytorch.³³

2 | PROPOSED DAMAGE DETECTION FRAMEWORK: AN UNSUPERVISED APPROACH BASED ON MACHINE LEARNING

Proposed here is an unsupervised damage detection framework (Figure 1), which contains three steps: (i) data preparation, (ii) statistical modeling, and (iii) decision-making. The statistical modeling and decision-making are realized by artificial neural networks and one-class classification, respectively.

2.1 | Data preparation

Data type: for this data-driven method, damage detection is performed by discovering the temporal and special correlations in the input data. Therefore, as long as the input contains the same amount of information about the structure, the input type may not influence the results.

Sensor locations: sensor locations may influence the amount of information in the input data and therefore influence the performance of the proposed method. Sensor locations should be selected to cover the motion of each dynamic

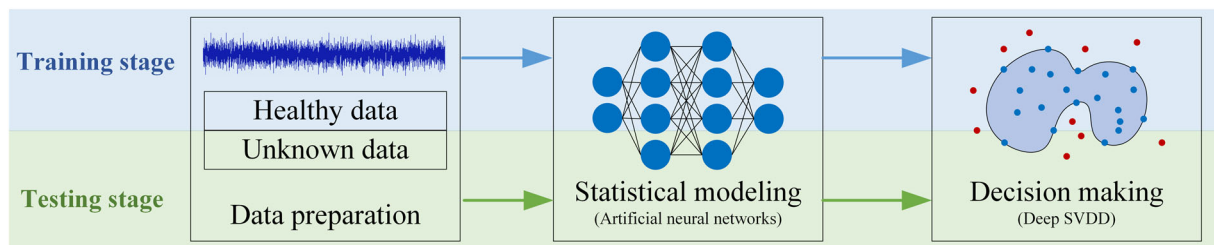


FIGURE 1 Flowchart for the proposed framework. SVDD, support vector data description.

degree of freedom; otherwise, some changes will not be detected. For example, if the sensors are located exactly at the rotational center of each floor, the rotational response induced by asymmetric damage cannot be captured. Also, the amplitude of dynamic response at different locations varies, which leads to different signal-to-noise ratios (SNRs). Therefore, locations with higher dynamic responses are preferred.

Standardization: neural networks are preferable to be trained with standardized data. High-input magnitudes may have some negative effects on the convergence of the algorithm due to the nature of gradient descent and back-propagation (BP), but this can be solved by standardization.

Denoising: to make sure that no useful information is removed from the data, caution needs to be exercised while denoising. Hence, it is preferred to feed the model with the data before denoising. Although for neural networks, noisy data may result in an over-fitting, some training strategies can be implemented to alleviate this problem (see Section 2.5).

Data splitting: data should be segmented into three parts: training, validation, and testing. The training data from a healthy structure is used to tune the model parameters in the statistical modeling phase. The validation data (also from the healthy structure) is used to determine the termination condition for the training process. Because the validation data are not directly used in tuning, the model parameters over-fitting can be alleviated. Testing data can be collected from a healthy structure or damaged structure when available, which are used to test the performance of the trained model and to determine if there is a need for retraining with improved model hyper-parameters.

2.2 | Statistical modeling

One of the axioms of damage detection is that “The damage assessment requires a comparison between two systems.”³⁴ Therefore, statistical modeling aims to extract damage-sensitive features and reveals the difference between healthy structures and damaged structures. Under environmental excitations, the dynamic response of a structure is inherently random. For an independent random process, the changes can be reflected by mean (median) or standard deviation, and then, some change detection algorithms cumulative sum control chart (CUSUM)³⁵ or generalized likelihood ratio (GLR)³⁶ can be directly implemented. However, a transformation (or inverse filtering) from observations to innovations is required because the structural dynamic responses are normally dependent and auto-correlated.³⁷

As an independent random process, the structural dynamic response can be generally described by a conditional distribution model³⁷ as

$$\hat{y}_k = p_{\theta}(y_k | Y_{k-p}^{k-1}), \quad (1)$$

where \hat{y}_k and y_k are the observed and expected dynamic response at time instant k ; θ is a vector containing parameters related to the state of the monitored structure; $Y_{k-p}^{k-1} = (y_{k-1}, y_{k-2}, \dots, y_{k-p+1}, y_{k-p})^T$ is a vector of the observed historic dynamic responses at previous p instants; and $p_{\theta}(\cdot)$ is a parameterized conditional probability density of the observed dynamic response at the current instant. When the monitored structure is damaged, the elements in the vector θ change, resulting in a different conditional probability density compared with that of a healthy structure. Fitted with enough data collected from a healthy structure, this conditional probability density function can then be identified, and the residual or innovation at instant k can be obtained by:

$$\varepsilon_k = y_k - E_{\theta_0}(y_k | Y_{k-p}^{k-1}), \quad (2)$$

where θ_0 is a vector containing parameters obtained from a healthy structure and E_{θ_0} is the expected dynamic response calculated by the fitted probability density function. When Y_{k-p}^{k-1} in Equation (2) is obtained from a healthy structure, the innovation ε_k is supposed to be an independent Gaussian sequence, and the damage can be indicated if the innovation is no longer Gaussian. In some conventional cases, the expectation E_{θ_0} of the monitored dynamic system is simply calculated by a discrete state-space model^{38,39} or a linear AR model. However, these models will cause limitations when the monitoring data are massive and nonlinearly correlated; hence, a machine-learning model is adopted in this paper to generate the expectations.

Intuitively, the key idea of statistical modeling in the proposed damage detection framework is to train a prediction model using recorded history data obtained from a healthy structure, and when an anomaly occurs, the prediction will become inaccurate, which can be taken as an indication of damage. By training a prediction model using monitoring data, we can obtain this pattern in an unsupervised and adaptive manner. The logic behind this modeling strategy is that when something is identified as abnormal, it means that the observation will be different from the expectations.

Depending on the number of data channels used for statistical modeling, two types of approaches are defined here, namely, the single-sensor approach and the multi-sensor approach.

2.2.1 | Single-sensor statistical modeling approach based on multi-layer perceptron (MLP)

Although conventional auto-regressive models predict the data at the current instant as a linear combination of the latest history data, MLP⁴⁰ can introduce a more complex nonlinear autocorrelation. The MLP is one of the most popular machine-learning models and is categorized as a type of feedforward neural network, whose connections between units do not form a cycle. A typical MLP is often composed of an input layer, one or multiple hidden layers, and an output layer, and each layer contains a certain number of units called neurons (Figure 2a).

The neurons in adjacent layers are usually fully connected through three operators, namely, multiplying a weight, adding a bias, and passing through an activation function, which is considered the main source of nonlinearity in MLP, as illustrated in Figure 2b and Equation (3).

$$s_j = f_a \left(\sum_i \omega_{ij} r_i + b_j \right), \quad j = 1, 2, 3, \dots, \quad (3)$$

where r_i is the value of i -th neuron in the previous layer and s_j is the value of j -th neuron in the next layer; $f_a(\cdot)$ is the activation function; and ω_{ij} and b_j are the weight and bias, respectively, for calculating s_j from r_i . Calculating the expectations of structural dynamic response by an MLP, the innovation in the Equation (2) can then be rewritten as

$$\varepsilon_k = y_k - MLP_{\theta_0} \left(y_k | Y_{k-p}^{k-1} \right), \quad (4)$$

where $MLP_{\theta_0}(\cdot)$ is the MLP trained with data obtained from a healthy structure. The number of hidden layers and neurons within each layer are hyper-parameters, which should be determined according to the complexity of the problem.

With a large amount of input and output data as training dataset, the parameters (weights and biases) of an MLP are then tuned with the objective of minimizing the loss function. In the tasks of regression and prediction, the loss function is often defined as the deviation between the predicted data and the true data and formulated as the mean square error or the mean absolute error. It is almost impossible to determine the weights and biases analytically due to

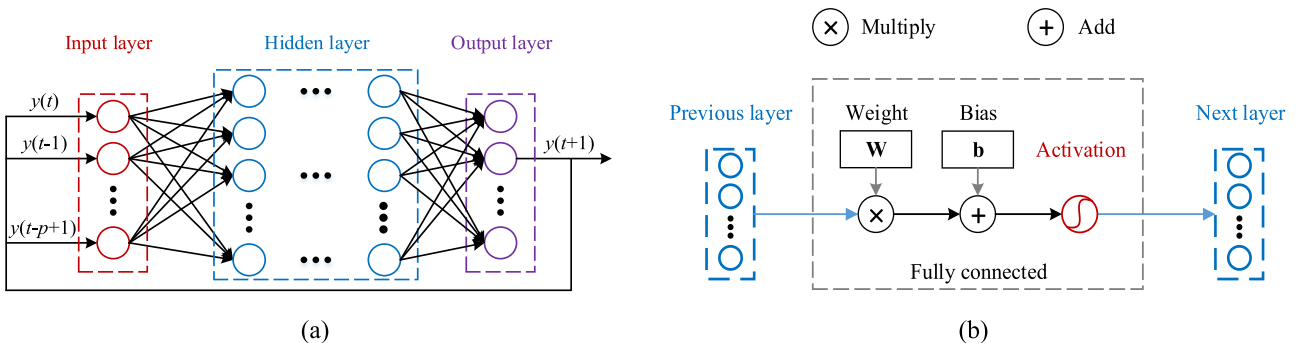


FIGURE 2 The architecture of a typical multi-layer perceptron (MLP). (a) A typical architecture of MLP and (b) three operators between adjacent layers.

the nonlinearity and complex nature of MLP. For this reason, the model parameters in MLP are calculated iteratively with BP,^{41,42} which is widely used for training feedforward neural networks.

2.2.2 | Multi-sensor statistical modeling approach based on convolutional neural network (CNN)

In a multi-sensor approach, the input and output data of the statistical model can be a multi-dimensional tensor; hence, the CNN is adopted for this approach to process high-dimensional data. The CNN^{43,44} is also a type of feedforward neural network, which is usually composed of convolution layers, fully connected layers (FCLs), and multi-layer perceptions. However different from MLP, the input layer of CNN can be multi-dimensional matrices, not limited to vectors, which makes CNN especially suitable for image processing.

A typical CNN usually consists of an input layer, one or multiple convolutional layers (CLs) and pooling layers (PLs), one or multiple FCLs, and an output layer. The CL is one of the most distinctive features of CNN, which convolves the data from the previous layer with several kernels, and then passes it through an activation function to obtain data in the next layer (see Figure 3). The number of channels (often referred to as depth) of the input layer should be equal to the depth of kernels, whereas the depth of the output layer should be equal to the number of kernels. Pooling is essentially a down-sampling process, which reduces the size of data by certain sampling rules, such as max pooling and mean pooling. In general, with CLs and PLs, CNN can capture the spatial information from the input data and reduce the depth and size of high-dimensional input data with relatively few numbers of undetermined parameters.

For neurons in adjacent CLs, the calculation rules can be described by the following equation if no padding has been implemented:

$$s(u, v, l) = f_a \left(\sum_k \sum_{i=1}^m \sum_{j=1}^n r(u+i-1, v+j-1, k) K^{(l)}(i, j, k) \right), \quad (5)$$

$$u = 1, 2, \dots, n-m+1; \quad v = 1, 2, \dots, p-n+1$$

where $r(i, j, k)$ and $s(i, j, k)$ are elements in the tensors of the previous layer and the next layer respectively, which are located at row i , column j , and depth k ; $f_a(\cdot)$ is the activation function; $K^{(l)}(i, j, k)$ is an element of the l -th kernel function located at row i , column j , and depth k ; and m and n are the maximum number of rows and columns of the previous layer, respectively. Calculating the expectations of structural dynamic response by a CNN, the innovation vector at instant k can then be rewritten as:

$$\mathbf{e}_k = \mathbf{y}_k - \text{CNN}_{\theta_0} \left\{ \begin{pmatrix} \hat{y}_{1,k} \\ \hat{y}_{2,k} \\ \hat{y}_{3,k} \\ \vdots \end{pmatrix} \left| \begin{pmatrix} y_{1,k-1} & y_{1,k-2} & \cdots & y_{1,k-p} \\ y_{2,k-1} & y_{2,k-2} & \cdots & y_{2,k-p} \\ y_{3,k-1} & y_{3,k-2} & \cdots & y_{3,k-p} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \right. \right\}, \quad (6)$$

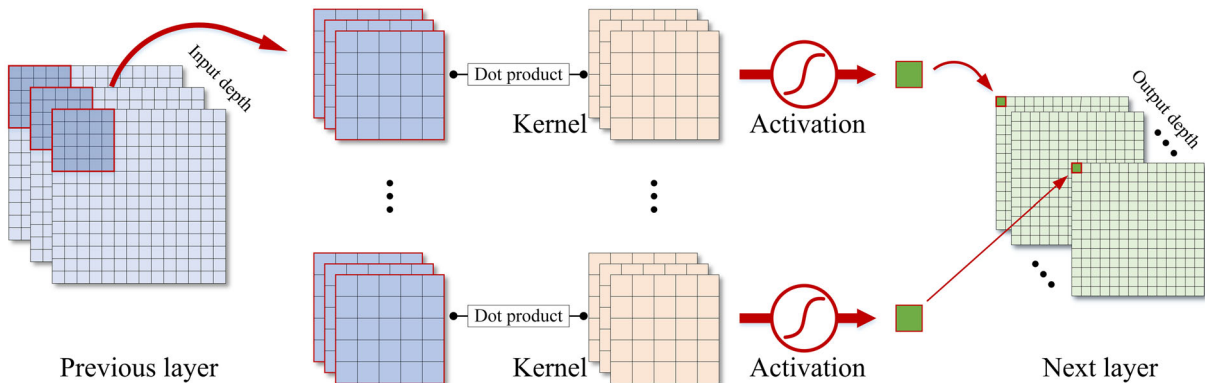


FIGURE 3 The architecture of a typical convolution layer.

where \mathbf{e}_k becomes a vector with dimension equal to the number of channels; $\mathbf{y}_k = (y_{1,k}, y_{2,k}, y_{3,k}, \dots)^T$ is the observed dynamic response at instant k ; $\hat{y}_{i,k}$ and $y_{i,k}$ are the expected and observed dynamic response for i -th channel at instant k calculated by CNN; and $CNN_{\theta_0}(\cdot)$ is the CNN trained with data obtained from a healthy structure. The hyper-parameters are explained in detail in Section 3.

The loss function of CNN is also defined as the deviation between the predicted data and true data in the task of regression and prediction. With enough training data, the undetermined model parameters of CNN are tuned to minimize the loss function of CNN using BP, similar to the training process of MLP but with a little difference when propagating the errors through PLs and CLs.

2.3 | Training the statistical model with an iterative pruning strategy

The numbers of hidden layers and neurons embedded in each layer are important hyper-parameters, which often influence the performance of the model and should be determined before training. Although the computational efficiency of neural networks is positively correlated with the complexity of model construction, the accuracy may, however, not be the case. This is due to the fact that among a large number of model parameters, some are redundant and do not contribute much to the output. The redundancy can be reduced by removing some of the low-ranked connections between neurons, and a more concise model can then be obtained after fine-tuning without losing accuracy. This process is also known as pruning.^{45–47}

In this research, due to the variable size of the window length, the optimal architecture of neural networks changes accordingly. Therefore, in order to adaptively obtain the optimal model architecture for different application scenarios and conduct a reasonable parametric study in Section 3.5, an iterative pruning strategy is proposed here for the training of the statistical models, as demonstrated in Figure 4. The proposed iterative pruning process includes the following five steps:

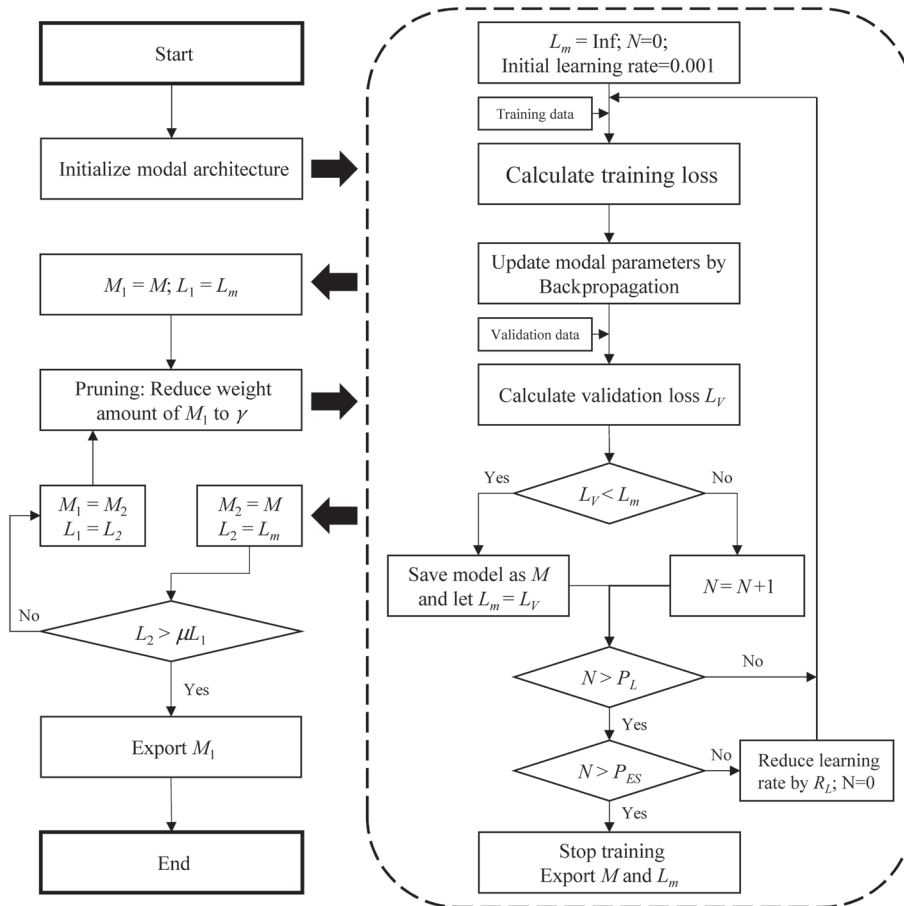


FIGURE 4 Flowchart of iterative pruning.

- Step 1. Generating an initial neural network for training. The determination of the initial model structure is still intuitive, but redundant neurons can be accepted because they can be automatically pruned during training.
- Step 2. Tuning the model parameters using BP (indicated within the dotted box in Figure 4), then preserving the trained model as M_1 and its validation loss as L_1 . The performance of a machine-learning model, MLP or CNN, can be affected by the adopted training strategy, including the selection of optimizers,⁴⁸ the determination of learning rates, and the selection of the stopping criteria. Here, a mini-batch learning strategy is adopted with “Adam”⁴⁹ as the optimizer, which can be directly implemented using Pytorch. The learning rate is designed to decrease by a reduction factor R_L when the validation loss is not reduced in P_L epochs. In order to avoid under-fitting and over-fitting, an early stopping strategy is used to determine a training termination, where the training algorithm is terminated when the validation loss is not reduced in P_{ES} epochs. Values of the relevant parameters adopted in this paper are presented in Table 1.
- Step 3. Pruning the trained model by reducing its amount of weight. The weights are first ranked by their absolute values, which reflect the contribution of one neuron to another or the importance of the connection between two neurons, then the minimum $1-\gamma$ of the weights are manually set to zero. In this paper, the parameter γ is set as 80% (Table 1).
- Step 4. Repeat step 2 to fine-tune the model parameters after pruning, then preserve the pruned model as M_2 and its validation loss as L_2 .
- Step 5. Repeat step 2 through step 4 until L_2 is larger than μL_1 , and output M_1 as the ultimate model when the loop terminates. Otherwise, replace L_1 and M_1 with L_2 and M_2 , respectively, then jump to step 2 to continue the loop. The parameter μ is set as 1.001 in this paper (Table 1).

The proposed iterative pruning method reduces the number of model weights by iteratively removing redundant connections between neurons, and the process of pruning and termination becomes automatic once the parameters presented in Table 1 are set.

2.4 | Decision-making process using one-class classification

After damage-sensitive features are extracted by statistical modeling, a real-time decision-making process is designed to evaluate the condition of the monitored structure. The basic idea of real-time decision-making is to calculate a decision boundary (or threshold) in advance using the innovation extracted from the healthy data, and then decide on the condition of the monitored structure by a timely comparison of the innovation extracted from unknown data with this pre-determined boundary. For the single-sensor approach, the extracted innovation is a one-dimensional time series. But for the multi-sensor approach, the extracted innovation is a high-dimensional time series, which means a decision boundary should be determined in a high-dimensional space. One conventional alternative is to fit the innovation series with a parameterized probability density function and declare a certain quantile as the boundary.^{29,50} However, a small probability of occurrence does not necessarily indicate an anomaly when the probability density function is obtained by using healthy data only, and a false alarm will be triggered in this case. Therefore, non-probability-based one-class classification methods such as support vector domain description (SVDD)^{51,52} and deep support vector domain description (DSVDD)⁵³ are introduced to calculate the decision boundary.

2.4.1 | SVDD

SVDD belongs to a class of one-class data classification method. It is used to obtain a description of a training set composed of normal objects and to detect the new objects that resemble this training set.⁵³ The soft-boundary one-class classification can be described generally by an optimization problem in Equation (7), where the decision boundary is

TABLE 1 Parameters for model training with iterative pruning strategy

Training parameters	R_L	P_L	P_{ES}	M	γ
Value	0.3	5	10	1.001	80%

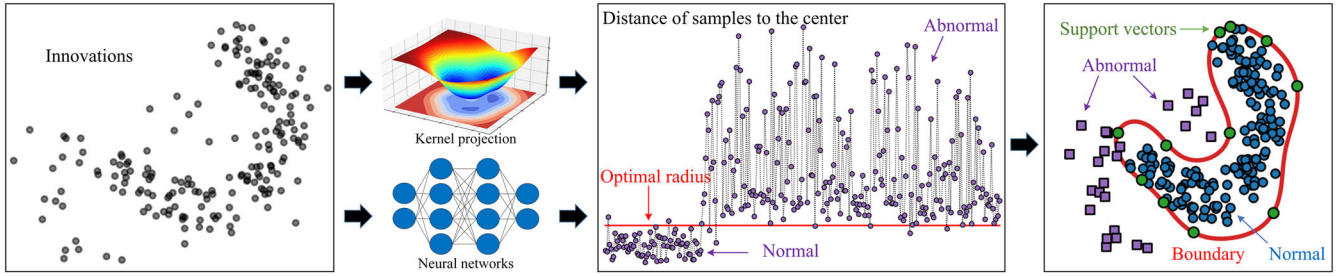


FIGURE 5 Illustration of the support vector domain description and deep support vector domain description.

defined as the minimal hypersphere that contains the mapped innovations in a high-dimensional space, as illustrated in Figure 5.

$$\begin{aligned} \min_{\mathbf{a}, R, \xi} R^2 + C \sum_i \xi_i \\ \text{s.t. } \|\mathbf{K}(\mathbf{e}_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, 3, \dots, \end{aligned} \quad (7)$$

where parameters \mathbf{a} and R are the center coordinate and radius of the hypersphere, respectively; parameter ξ_i is a slack variable for the i -th sample of innovation; hyper-parameter C controls the trade-off between the volume and the errors, which avoids an over-estimated radius caused by unexpected anomaly data; and $\mathbf{K}(\cdot)$ is a kernel function that maps the low-dimensional input into a higher dimensional space, in which the mapped innovations can be clustered. Incorporating the constraint into the objective, Equation (7) can be transformed into a standard quadratic programming problem by using Lagrange multipliers,⁵² as described in Equation (8).

$$\begin{aligned} \min_{\alpha} \sum_{i,j} \alpha_i \alpha_j \langle K(\mathbf{e}_i), K(\mathbf{e}_j) \rangle - \sum_i \alpha_i \langle K(\mathbf{e}_i), K(\mathbf{e}_i) \rangle \\ \text{s.t. } 0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i = 1, \end{aligned} \quad (8)$$

where α_i is the Lagrange multiplier for the i -th sample of innovation, which is the only parameter to be optimized in this problem and $\langle \cdot, \cdot \rangle$ represents the dot multiplication of the two contained elements. After the optimal value of α is solved, the center coordinate \mathbf{a} and radius R of the hypersphere can be obtained using Equation (9). Those data points with Lagrange multipliers in the range of $(0, C)$ are defined as the support vectors.

$$\begin{aligned} \mathbf{a} &= \sum_i \alpha_i K(\mathbf{e}_i) \\ R &= \sqrt{\langle K(\mathbf{e}_{sv}), K(\mathbf{e}_{sv}) \rangle - 2 \sum_i \alpha_i \langle K(\mathbf{e}_{sv}), K(\mathbf{e}_i) \rangle + \sum_{i,j} \alpha_i \alpha_j \langle K(\mathbf{e}_i), K(\mathbf{e}_j) \rangle}, \end{aligned} \quad (9)$$

where \mathbf{e}_{sv} is one of the support vectors. Then a decision can be timely made on the unknown sample \mathbf{e}_{un} according to the criteria described in Equations (10) and (11).

$$\text{condition} = \begin{cases} \text{healthy,} & \text{if } R_{un} \leq R \\ \text{unhealthy,} & \text{if } R_{un} > R \end{cases}, \quad (10)$$

where R_{un} is the distance of an unknown sample innovation to the center of the hypersphere in the high-dimensional space, which can be calculated by

$$R_{un} = \sqrt{\langle K(\epsilon_{un}), K(\epsilon_{un}) \rangle - 2 \sum_i \alpha_i \langle K(\epsilon_{un}), K(\epsilon_i) \rangle + \sum_{i,j} \alpha_i \alpha_j \langle K(\epsilon_i), K(\epsilon_j) \rangle}. \quad (11)$$

Although the aforementioned decision-making strategy establish based on SVDD is effective in dealing with small-scale data,²⁸ poor computational efficiency occurs when the data scale gets larger because of the computational cost associated with the construction and manipulation of the kernel matrices.⁵³

2.4.2 | DSVDD

Developed from the SVDD, the DSVDD is also a type of one-class data classification method, but kernel matrix construction and manipulation are no longer needed. Because the kernel function used in Equation (7) is actually employed as a mapping from a low-dimensional space to a high-dimensional space, a forward neural network is adopted in DSVDD to learn this mapping relationship adaptively from the training data (see Figure 5). Therefore, the objective of DSVDD with a soft-boundary can be defined as

$$\min_{R, \omega} R^2 + \frac{1}{\nu n_s} \sum_{i=1}^{n_s} \max \{0, \|\Phi(\epsilon_i | \omega) - \mathbf{a}\|^2 - R^2\} + \frac{\lambda}{2} \sum_i \|\omega_i\|_F^2, \quad (12)$$

where $\nu \in (0,1)$ is a hyper-parameter that 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; $\Phi(\cdot | \omega)$ is the mapping calculated by a neural network with unknown parameter ω ; $\lambda > 0$ is a weight decay regularizer on the network parameters; and $\|\cdot\|_F$ represents the Frobenius norm. A mapping relationship can be found by solving the optimization problem in Equation (12), such that the data are closely clustered into a hypersphere in the mapped space and also allowing for a few potential unexpected outliers.

Considering Equation (12) as the loss function of a neural network, the parameters ω can be solved by stochastic gradient descent (SGD) given a fixed R . Thereby, a fixed value of R is initiated for a few early training epochs to train the parameters ω , and then after warming up, the R is updated with ω after each epoch. A detailed description of the parameter optimization of DSVDD can be found in a study by Ruff et al.⁵³ After the DSVDD is trained with data obtained from a healthy structure, a decision can then be timely made on the unknown sample ϵ_{un} according to the criteria described in Equations (10) and (13).

$$R_{um} = \sqrt{\|\Phi(\epsilon_{un} | \omega) - \mathbf{a}\|^2} \quad (13)$$

2.5 | Under-fitting and over-fitting

The under-fitting and over-fitting are two of the most common problems encountered when training neural networks. At the statistical modeling stage, an under-fitted prediction model may not perform well on both the training data and testing data, whereas an over-fitted prediction model may perform well on training data but perform poorly on testing data. These two types of problems will lead to erroneous decisions on the condition of structures during the decision-making stage.

As illustrated in Figure 6a, for the decision-making with a perfectly fitted model, the decision boundary is generated by the distance of training innovations to the center of the hypersphere, such that the distance of the innovations calculated from abnormal testing data are outside the decision boundary, whereas the distance of the innovations calculated from normal testing data are within the decision boundary, which allows the SHM system to make an accurate decision on the condition of the monitored structure. However, an under-fitted model cannot fit the data well enough. As illustrated in Figure 6b, the training innovations of the under-fitted model are relatively high compared with the testing innovations, and the difference between the testing innovations of normal data and abnormal data cannot be

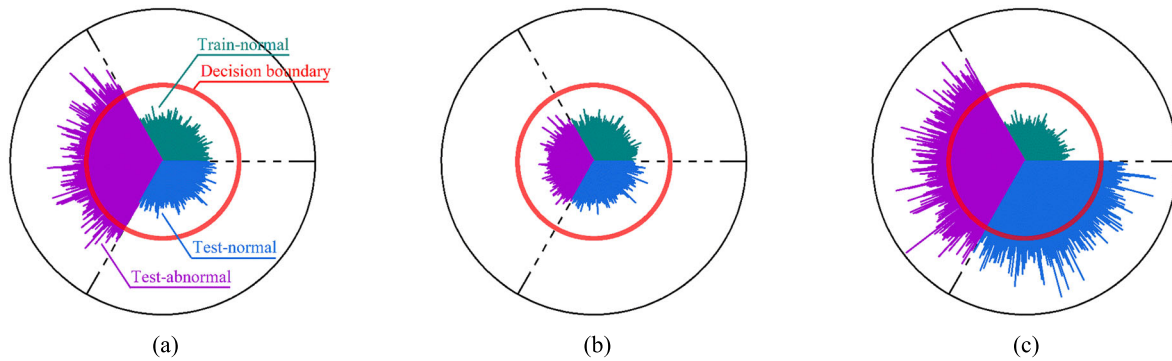


FIGURE 6 Decision-making mistakes caused by the three types of fitting conditions. (a) Decision-making with a perfectly fitted model, (b) decision-making with an under-fitted model, and (c) decision-making with an over-fitted model.

distinguished by an exaggerated decision boundary. On the other hand, although an over-fitted model may perform well on the training data, it can issue false positives on the testing data. This means that false indications of damage will be declared by the SHM system when none is present, as illustrated in Figure 6c. It is worth pointing out that the decision mistakes caused by the under-fitted model and over-fitted model in the proposed framework exactly coincide with the two typical categories⁴ of false indications for damage detection in SHM systems.

Because a large amount of data can be collected and used by the SHM system, the main cause of under-fitting in this framework is the inappropriate selection of the model complexity for the neural networks used for statistical modeling. When the construction of neural networks is too simple (i.e., a small number of input features and parameters), the pattern contained in the training data cannot be fully learned, which is the reason why an under-fitted model always generates large innovations for both training data and testing data. Therefore, the under-fitting can be avoided by reasonably increasing the model complexity. On the other hand, the over-fitting is usually related to a high level of noise contained in the training data. Specifically, the over-fitted model extracts meaningless information from the noise and fails to recognize the untrained healthy data. Hence, as described in Section 2.3, an early stopping strategy is adopted in this research to prevent over-fitting, where the training process is designed to be terminated when the validation loss can no longer be reduced in certain epochs, and the model parameters that can produce a minimum validation loss are adopted as the optimal solution. Also, an iterative pruning strategy is adopted to prevent over-fitting by adaptively seeking a model with the optimal model structure during training.

3 | VERIFICATION USING THE BENCHMARK DATASET

3.1 | Data simulated by the phase I IASC–ASCE benchmark

3.1.1 | Definition of damage patterns

The phase I IASC–ASCE benchmark⁵⁴ is a simulated SHM dataset developed by International Association for Structural Control and American Society of Civil Engineering, which provides a common platform for researchers worldwide to investigate the performance of damage detection and damage classification algorithms. The analytical structure monitored in this benchmark is a four-story, two-bay by two-bay quarter-scale steel frame, as illustrated in Figure 7a. The beams and columns are modeled as Euler–Bernoulli beams, whereas the slabs are considered as rigid diaphragms. Because the mass on each floor is condensed into the central nodes, a 12-degree-of-freedom analytical model is thereby obtained.

As shown in Figure 7, IASC and ASCE predefined six damage patterns for this benchmark, simulated by reducing the stiffness of certain numbers of braces and weakening beam–column connections. The damage patterns are specifically defined as follows: (i) in damage pattern 1 (DP1), the stiffness of all braces of the first story is reduced by 100%, whereas the broken braces still contribute to the mass; (ii) in damage pattern 2 (DP2), the stiffness of all braces of the first and third stories is reduced by 100%; (iii) in damage pattern 3 (DP3), the stiffness of one brace on one side of the first story is reduced by 100%; (iv) in damage pattern 4 (DP4), the stiffness of one brace in the first story and one brace

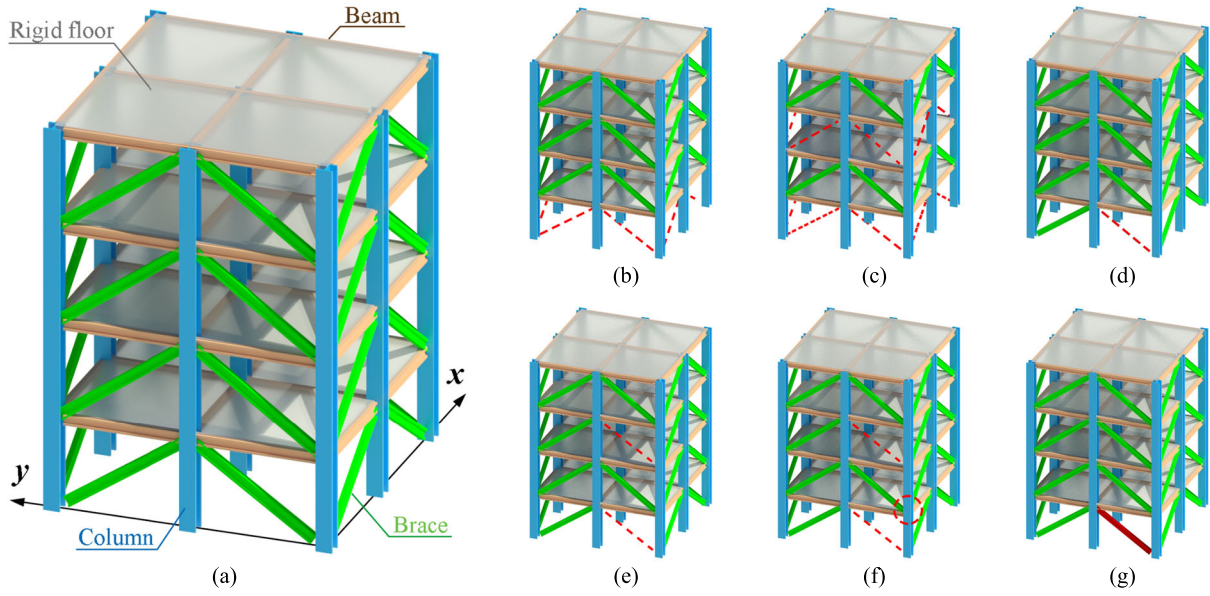


FIGURE 7 (a) The benchmark structure (four-story steel-frame), (b) damage pattern 1, (c) damage pattern 2 (d) damage pattern 3 (e) damage pattern 4, (f) damage pattern 5, and (g) damage pattern 6 (reproduced from a study by Johnson et al.⁵⁴).

in the third story is reduced by 100%; (v) in damage pattern 5 (DP5), the stiffness of one brace in the first story and one brace in the third story is reduced by 100%, and one beam–column connection between the first and second story is modified to only transmit forces without moments; (vi) in damage pattern 6 (DP6), the stiffness of one brace in the first story is reduced by 1/3. The percentage loss in horizontal story stiffness caused by each damage pattern is calculated and presented in Table 2. To evaluate algorithm performance under different levels of damage, the six damage patterns are thus categorized into three severity levels, namely, severe, moderate, and minor.

3.1.2 | Sensor layouts and data generation

In this study, the acceleration response of certain nodes is assumed to be recorded by sensors when the analytical model is excited by an independent Gaussian white noise in the x - and y -direction simultaneously. The locations for 16 channels of acceleration sensors are provided in Figure 8. Several levels of sensor noise are also introduced to simulate real conditions and evaluate the robustness of the proposed method. In the low noise level cases, acceleration data with noise levels of 2%, 4%, 6%, and 8% are generated. For the monitoring of structures excited by environmental excitations with very low amplitudes, such as low-rise buildings, the corresponding amplitude of the dynamic response can also be low. This may lead to very low SNRs. To cover a wide range of SNRs, noise levels of 10%, 20%, 30%, 40%, and 50% are considered in this study. Different levels of noise are added to the training data by the rule defined in Equation (14).

$$y_i(t) = \eta \epsilon(t) \max_{i \in \mathbf{I}} \left\{ \sqrt{\frac{\int_{t_1}^{t_2} x_i^2(t) dt}{t_2 - t_1}} \right\} + x_i(t), \quad (14)$$

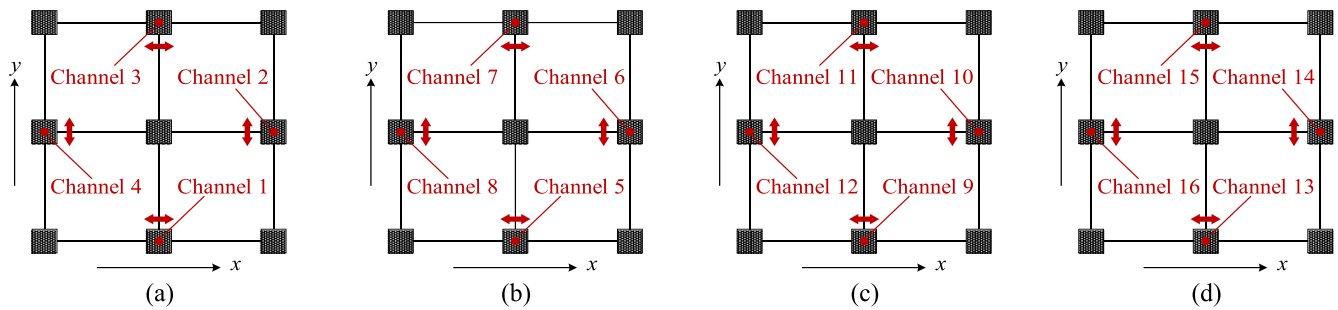
where $y_i(t)$ and $x_i(t)$ ($i \in \mathbf{I}$) are the i -th channel data with and without noise, respectively; \mathbf{I} is a vector of all available channels; η is the noise level; $\epsilon(t)$ is a set of normally distributed random numbers with a unit root mean square (RMS); and t_1 and t_2 are the start time and end time of the selected data fragment.

The acceleration data simulated from benchmark structures are split into three fragments for training, validation, and testing, respectively. The training and validation sets are both composed of healthy data fragments with a total duration of 3 h generated under each noise level, whereas the testing set is composed of both healthy and unhealthy

TABLE 2 Percentage loss in horizontal story stiffness caused by damages (reproduced from a study by Stavridis⁵⁶)

Story	Direction	Severe		Moderate			Minor
		DP1	DP2	DP3	DP4	DP5	DP6
First story	X	45.24%	45.24%	0	0	0	0
	Y	71.03%	71.03%	17.76%	17.76%	17.76%	5.92%
	Rotation	64.96%	64.96%	9.87%	9.87%	9.87%	2.88%
Second story	X	0	0	0	0	0	0
	Y	0	0	0	0	0	0
	Rotation	0	0	0	0	0	0
Third story	X	0	45.24%	11.31%	11.31%	11.31%	0
	Y	0	71.03%	0	0	0	0
	Rotation	0	64.96%	9.16%	9.16%	9.16%	0
Fourth story	X	0	0	0	0	0	0
	Y	0	0	0	0	0	0
	Rotation	0	0	0	0	0	0

Abbreviations: DP1, damage pattern 1; DP2, damage pattern 2; DP3, damage pattern 3; DP4, damage pattern 4; DP5, damage pattern 5; DP6, damage pattern 6.

**FIGURE 8** Sensor layout at each story. (a) Plan view at 0.9 m high, (b) plan view at 1.8 m high, (c) plan view at 2.7 m high, and (d) plan view at 3.6 m high.

data fragments with a total duration of 1 h generated for each damage and noise levels. All of the datasets are sampled with a frequency of 20 Hz.

3.2 | Model construction for the single-sensor approach

The initial architecture of the MPL before pruning used in statistical modeling for single-sensor approach is presented in Figure 9. As illustrated, a moving window with a length of $(L_s + 1)$ samples is implemented on the acceleration data obtained from channel #13 to generate input–output samples for model training and testing. The acceleration data within a window is segmented into two parts, where the first sample is taken as the output and the rest L_s samples are taken as the input. Then, the input data are passed through two FCLs with, respectively, 64 embedded neurons to calculate the output.

Replacing the statistical model in Equation (4) with the MLP described in Figure 9, the innovation can thereby be calculated by

$$\epsilon_k = y_{13,k} - FCL^{(1, \text{Linear})} \left(FCL^{(64, \text{LReLU})} \left(FCL^{(64, \text{LReLU})} (\mathbf{Y}_S) \right) \right), \quad (15)$$

where

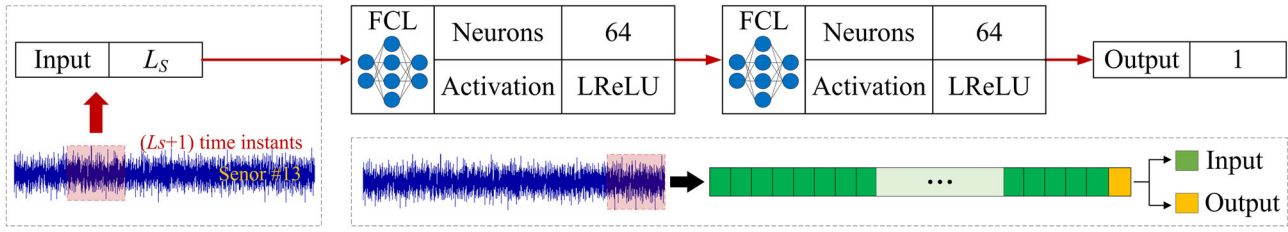


FIGURE 9 The data format and the statistical model construction for the single-sensor approach.

$$\mathbf{Y}_S = (y_{13,k-1} \ y_{13,k-2} \ y_{13,k-3} \ \cdots \ y_{13,k-L_S+1} \ y_{13,k-L_S}), \quad (16)$$

where $y_{13,k}$ is the dynamic response of channel #13 at instant k ; $FCL^{(1,Linear)}(\cdot)$ represents the FCL with 1 neuron and linear activation function, similarly for $FCL^{(64,LReLU)}(\cdot)$ with 64 neurons and activation function leaky rectified linear unit (LReLU)⁵⁵; and \mathbf{Y}_S is the input for the single-sensory statistical modeling, which is described in Equation (16). As will be explained in Section 3.5, in order to investigate the influence of the different choices of window lengths, numerical simulations are performed with L_S set as 5, 10, 15, 20, 25, 30, 35, and 40, respectively. During the real-life application of the proposed method, the window length can be selected slightly larger than the order of auto-regression estimated from the data.

3.3 | Model construction for the multi-sensor approach

To process the multi-channel input data, CNN is employed for statistical modeling in a multi-sensor approach and used for the verification of benchmark simulations. The amplitude of dynamic response at different locations varies, which leads to different SNRs. In the benchmark case considered here, the amplitude of the dynamic response of the lower stories is lower than those of the higher stories. At the same time, the noise is introduced with RMS relative to the largest RMS of the acceleration response (i.e., typical roof accelerations), which can be seen in Equation (14). Therefore, the information collected by sensors located at lower stories is more contaminated by noise than those located at higher stories. On the other hand, because channels #13 and #14 are located at the roof, their SNRs are higher compared with the sensors located at lower stories.

As illustrated in Figure 10, all the 16 channels of acceleration data within a moving time window with a length of $(L_M + 1)$ data points are first converted into a $16 \times (L_M + 1)$ matrix, and then an $L_M \times 4 \times 4$ tensor is extracted from this matrix to be fed into the input layer. Because the data of channels #13 and #14 are used for calculating the innovations, the data of these two channels at the current instant are excluded in the input. After the input layer, a CL is connected to extract features from the input data. Then the data from the CL are flattened in major column and passed through two FCLs in sequence to obtain the output. The activation function LReLU is also used for the hidden layers in this CNN. The weights in the kernels and between the FCLs are pruned during training using the method described in Section 2.3, so that the redundant network connections introduced by an oversized window can be removed. Unlike commonly used 1D-CNN, this approach takes the data from all sensors at one instant as a snapshot, which is varying through time. In this case, the window length determines the amount of information the algorithm can consider from the past, and by pruning kernels, the irrelevant data from low-ranked time instants can be removed. If using the conventional treatment where convolution is conducted along the time axis, pruning kernels cannot remove the data from low-ranked time instants but just change the filter.

Using the CNN described above and replacing the statistical model in Equation (6), the innovation at instant k can then be calculated by Equation (17)

$$\mathbf{e}_k = \begin{pmatrix} y_{13,k} \\ y_{14,k} \end{pmatrix} - FCL^{(2,Linear)} \left(FCL^{(128,ReLU)} \left(FCL^{(128,ReLU)} \left(FL \left(CL^{(6,2 \times 2,ReLU)}(\mathbf{Y}_M) \right) \right) \right) \right), \quad (17)$$

where the $y_{13,k}$ and $y_{14,k}$ are the acceleration responses of channels #13 and #14 at instant k ; $FCL^{(2,Linear)}(\cdot)$ represents the FCL with 2 neurons and a linear activation function, similarly for $FCL^{(64,LReLU)}(\cdot)$; $FL(\cdot)$ denotes the flatten

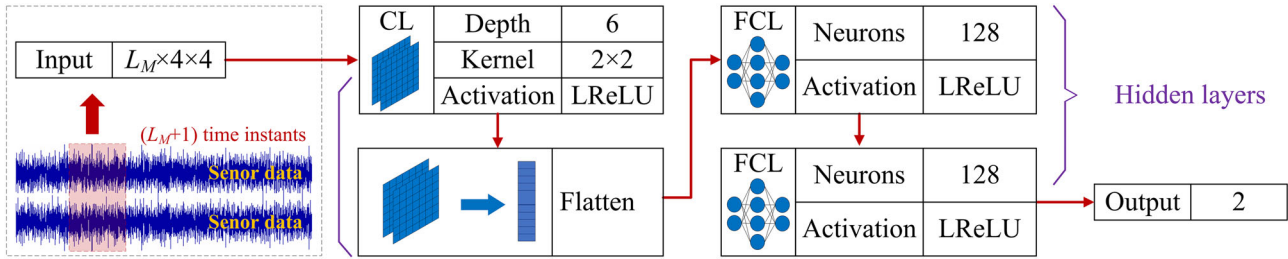


FIGURE 10 The data format and the statistical model construction for the multi-sensor approach.

layer; $CL^{(6,2 \times 2, \text{LReLU})}(\cdot)$ represents the CL with kernel depth 6, kernel size 2×2 , and with an LReLU activation function; and \mathbf{Y}_M is the input for multi-sensory statistical modeling as described specifically in Equation (18).

$$\mathbf{Y}_M = \begin{pmatrix} y_{1,k} & y_{1,k-1} & y_{1,k-2} & \cdots & y_{1,k-L_M+2} & y_{1,k-L_M+1} \\ y_{2,k} & y_{2,k-1} & y_{2,k-2} & \cdots & y_{2,k-L_M+2} & y_{2,k-L_M+1} \\ y_{3,k} & y_{3,k-1} & y_{3,k-2} & \cdots & y_{3,k-L_M+2} & y_{3,k-L_M+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{13,k-1} & y_{13,k-2} & y_{13,k-3} & \cdots & y_{13,k-L_M+1} & y_{13,k-L_M} \\ y_{14,k-1} & y_{14,k-2} & y_{14,k-3} & \cdots & y_{14,k-L_M+1} & y_{14,k-L_M} \\ y_{15,k} & y_{15,k-1} & y_{15,k-2} & \cdots & y_{15,k-L_M+2} & y_{15,k-L_M+1} \\ y_{16,k} & y_{16,k-1} & y_{16,k-2} & \cdots & y_{16,k-L_M+2} & y_{16,k-L_M+1} \end{pmatrix} \quad (18)$$

3.4 | Construction of neural networks for DSVDD

As described in Section 2.3.2, a neural network is used to replace the kernel projection in DSVDD. The input of this neural network is the innovation extracted from statistical modeling, and the output is a high-dimensional mathematical quantity with no physical meaning. In order to project low-dimensional features into a high-dimensional space, the input dimension is designed to be smaller than of the output dimension. The input dimension for the single-sensor approach and multi-sensor approach are 1 and 2, respectively, whereas the output dimensional is 8 for both approaches. The hidden layers are composed of two FCLs with 32 neurons and LReLU as the activation function.

3.5 | Sensitivity analysis to investigate the choice of the window length

The window length is one of the most important hyper-parameters in the proposed algorithm. It determines how much information the statistical model can extract from the monitoring data at a single step. When the window length is much smaller than the optimal value, the algorithm will have less capability to distinguish the difference between the damaged and undamaged states. On the other hand, if the window length is much larger than the optimal value, too much noise will be introduced to the statistical model and the resulting algorithm will lose accuracy. In addition, with the implementation of iterative pruning, it is also worth investigating if the optimal value of window length can be automatically found by the algorithm itself. Therefore, in this section, the proposed method is evaluated with different choices for the window length, which range from 5 to 40 data points with a step of 5.

3.5.1 | Evaluation criteria

To compare the performance of the single-sensor and the multi-sensor approaches, two types of evaluation criteria are proposed. The first criterion is mean alarm density (MAD) defined by Equation (19), which is used to evaluate the

mistakes made by not triggering the alarm when anomalies actually exist in the testing data. A higher value of MAD indicates a better performance of the algorithm.

$$\text{MAD} = \frac{\sum_{i=1}^{N_s} H(R_i^d - R)}{N_s}, \quad (19)$$

where N_s is the number of samples in the investigated testing fragment; $H(\cdot)$ is the Heaviside function; and R_i^d is the distance between the i -th innovation calculated from a damaged testing sample and the center of the hypersphere in the high-dimensional feature space. When using DSVDD as the decision-making strategy, R is the optimal radius of the hypersphere or the decision boundary.

The second criterion used in this section is mean false alarm density (MFAD) defined by Equation (20). This is used to evaluate the mistakes made by triggering alarms when no anomalies are actually existed in the testing data. A higher value of MFAD indicates a worse performance of the algorithm.

$$\text{MFAD} = \frac{\sum_{i=1}^{N_s} H(R_i^u - R)}{N_s}, \quad (20)$$

where R_i^u is the distance between the i -th innovation calculated from an undamaged testing sample and the center of the hypersphere in the high-dimensional feature space.

3.5.2 | Discussion on the results of MAD

For the single-sensor approach, the window length L_S is set as 5, 10, 15, 20, 25, 30, 35, and 40, and the simulations are separately conducted 10 times for each case, to eliminate random errors in the results. The mean values of MAD are calculated for each testing dataset under different noise levels and damage levels, as presented in Figure 11.

It can be observed that the MAD increases with the level of damage regardless of the variation in noise levels. Also, the categories for minor, moderate, and severe damage levels can be clearly distinguished in the figure, indicating the first type of criterion can also be considered as an indication of damage level. For the same damage type and noise level, a larger MAD often means a higher damage level, but it is also worth pointing out that because the level of damage is a relative quantity, an exact percentage of damage level is impossible to determine when the 100% damage level remains unknown. Thus, what can be achieved with MAD is to determine the severity of the current event compared with the MAD of historical events. Finally, with the increase of noise level, the MAD has a trend of decreasing, indicating a performance degradation due to the noise.

For the multi-sensor approach as presented in Figure 12, the calculated MAD values are inversely proportional to the noise level for the same damage pattern, but for the same noise level, no significant trend can be observed with the increase in damage level when the noise levels are below 4%. However, the multi-sensory damage features are sensitive to noise, and the trend of MAD becomes similar to that of the single-sensor approach when the noise levels are higher than 6%. This is mainly because the proposed method detects damage by identifying changes in the structural responses. When there is no damage, the algorithm learns from data that no rotational vibration response should exist. But when asymmetric damage happens, a new rotational mode of vibration may appear in the data introducing new correlations between data channels on a new dimension. This sudden change in the structural behavior results in high values of MAD even when the damage level is low, which reduces the performance of the proposed algorithm in identifying the damage level. The amplitude of the rotational response is smaller in comparison to the transverse response. Therefore, at high noise levels, the new correlations between data channels are disturbed by the noise, and they can no longer be detected by the algorithm. In this case, the performance of the proposed algorithm in identifying the damage levels seems to increase for higher noise. The exaggerated MAD values make the algorithm more sensitive to minor damages, which is beneficial for detecting the existence of damage. As for the real-life applications for symmetric structures, this phenomenon should be taken into consideration, and the decisions need to be made with the results obtained from both single- and multi-sensor approaches.

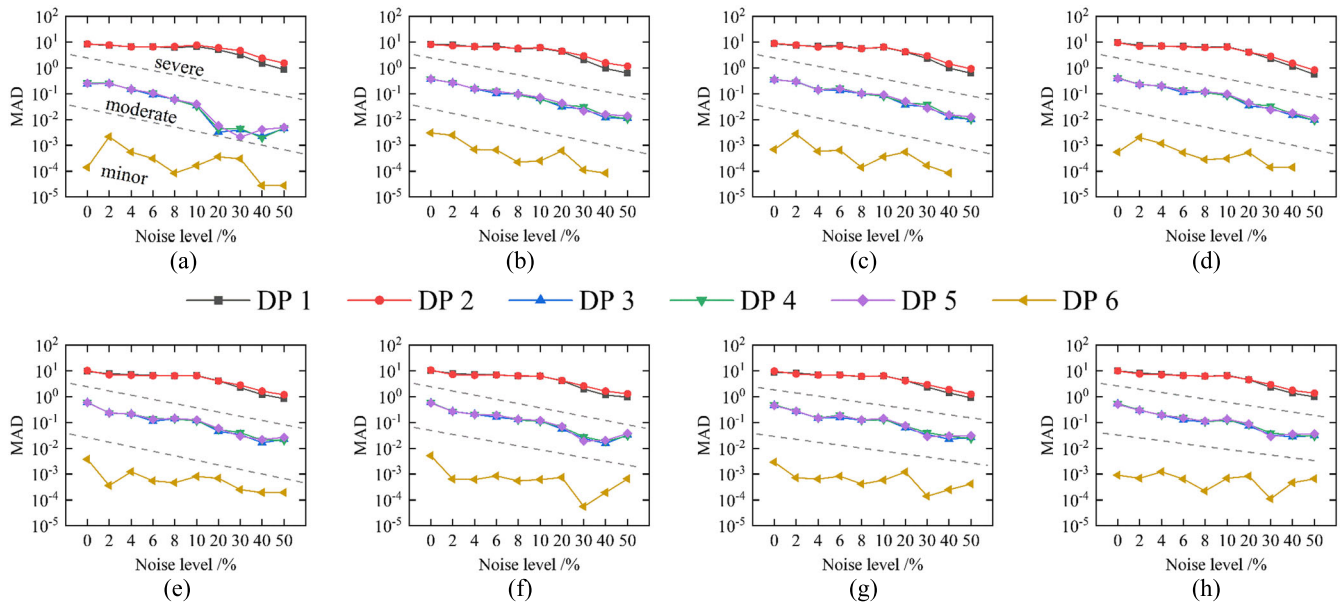


FIGURE 11 Trend of mean alarm density (MAD) calculated by the single-sensor approach with different window lengths, noise levels, and damage patterns. (a) Window length = 5, (b) window length = 10, (c) window length = 15, (d) window length = 20, (e) window length = 25, (f) window length = 30, (g) window length = 35, and (h) window length = 40.

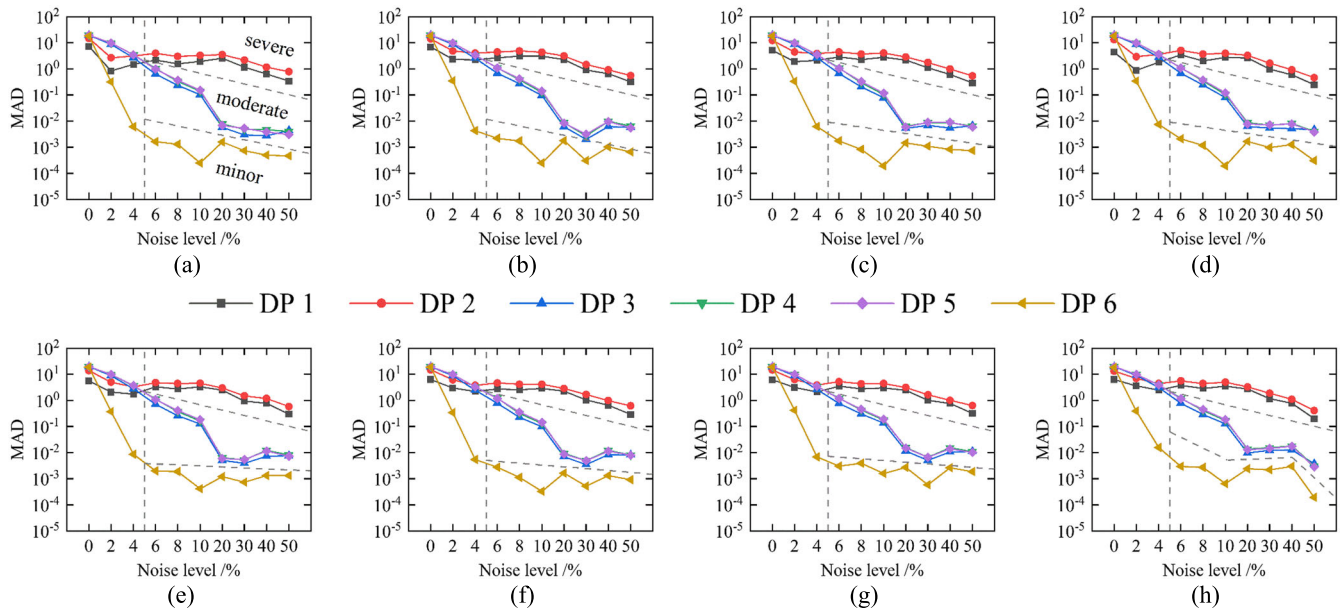


FIGURE 12 Trend of mean alarm density (MAD) calculated by the multi-sensor approach under different window lengths, noise levels, and damage patterns. (a) Window length = 5, (b) window length = 10, (c) window length = 15, (d) window length = 20, (e) window length = 25, (f) window length = 30, (g) window length = 35, and (h) window length = 40.

By comparing the single- and multi-sensor approaches, it can be concluded that (1) although introducing more data channels can reasonably improve the performance of the damage detection algorithm, an increased risk of performance reduction by introducing data with low SNRs should also be taken into consideration. (2) The damage level identification in three-dimensional conditions is more complicated than that in two-dimensional conditions, and the classification of damage patterns may be necessary before identifying damage levels for symmetric structures.

3.5.3 | Discussion on the results of MFAD

Although MAD is used to detect the existence and the level of the damage, MFAD is used to evaluate the reliability of issued real-time alarm. As described in Equation (20), the MFAD is proportional to the mean probability of generating a false alarm on healthy samples per unit time. When the value of MFAD is lower than the value of MAD, the existence and level of damage can be accurately identified. But when the value of MFAD is larger than the value of MAD, it becomes difficult to distinguish between false alarms and true alarms. This means that the algorithm failed to differentiate normal data from abnormal data. Therefore, the ratio of MFAD to MAD is used to investigate the probability of false alarms. A lower value of MFAD/MAD indicates a lower probability of false alarms, and a value of MFAD/MAD larger than 1 means that the damage cannot be accurately detected.

The trends of MFAD/MAD calculated by single- and multi-sensor approaches in terms of different noise levels, window lengths, and damage patterns are presented in Figures 13 and 14. These figures show that the value of MFAD/MAD is increasing with an increasing noise level and a decreasing damage level. This indicates a higher probability of false alarms when the data have a higher noise level, and the structure has a lower damage level. All the values of MFAD/MAD calculated for the single-sensor approach are lower than 1. Most of the MFAD/MAD values calculated for the multi-sensor approach are also lower than 1, except for the three cases where the damage levels are extremely low and the noise levels are high. The investigation of MFAD/MAD reveals that the proposed damage identification method has an extremely low probability of generating false alarms.

3.5.4 | Discussion on the optimal selection of the window length

The window length is an important parameter for statistical modeling. Although an undersized window length does not include many data points for feature extraction, an oversized window length can however introduce redundant information and noise. To tackle the problem of determining a proper window length, the training strategy with iterative pruning described in Section 2.3 is employed in this study. It can be seen in Figures 11 and 12 that the influence of window length on the MAD values is not significant using this training strategy. In this section, a more in-depth investigation will be presented on the iterative pruning method by visualizing the weights of the neural networks.

For the single-sensor approach, the visualized weights of the first hidden layer are shown in Figure 15 when the window length is 40. The black squares represent the weights being pruned, and the values of the remaining weights are characterized by color. It can be observed that the larger weights are gathered at the most recent instants, indicating the data at these instants are of more importance for feature extraction. For those cases, when noise levels are 6% and

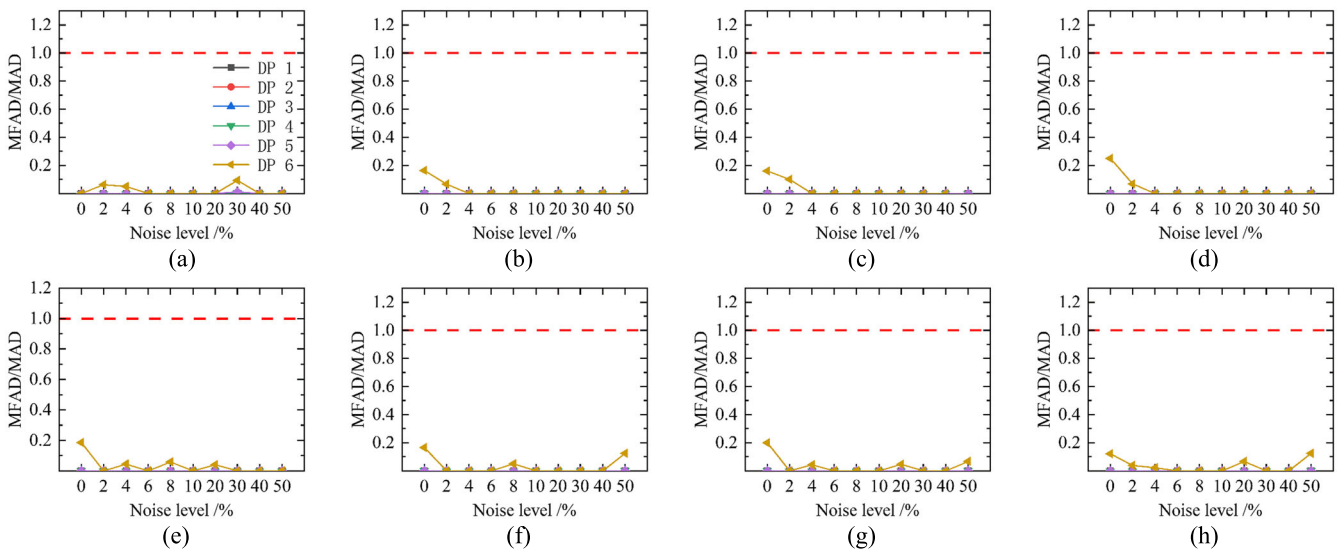


FIGURE 13 Trend of mean false alarm density/mean alarm density (MFAD/MAD) calculated by single-sensor approach. (a) Window length = 5, (b) window length = 10, (c) window length = 15, (d) window length = 20, (e) window length = 25, (f) window length = 30, (g) window length = 35, and (h) window length = 40.

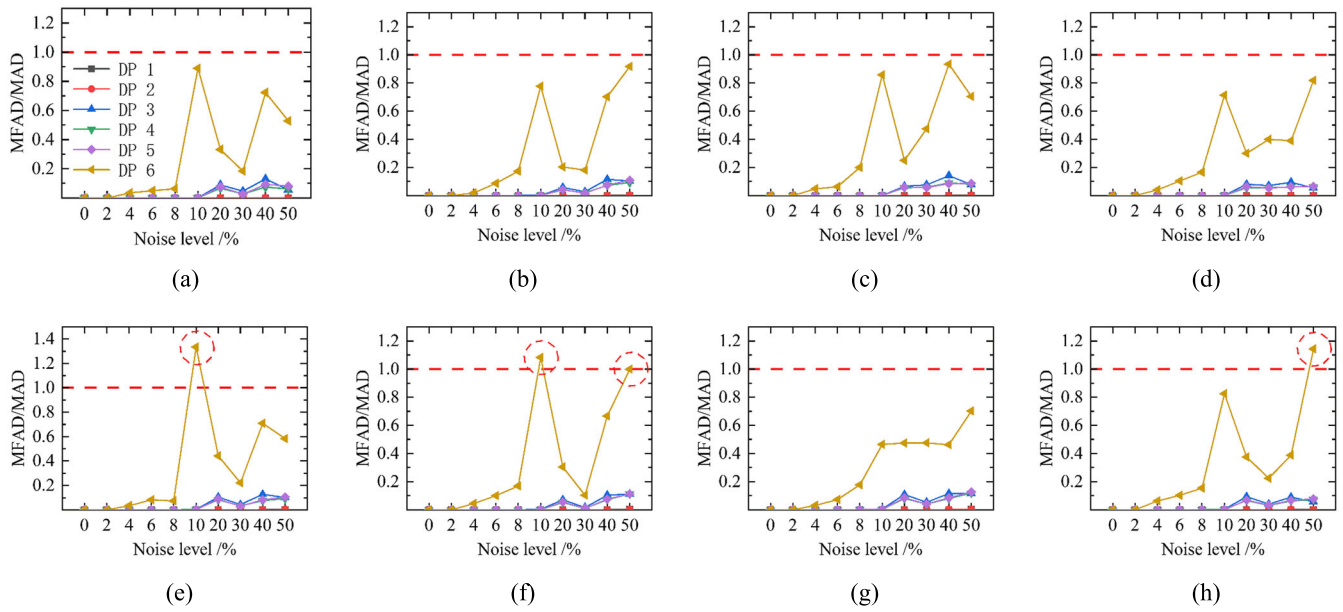


FIGURE 14 Trend of mean false alarm density/mean alarm density (MFAD/MAD) calculated by multi-sensor approach. (a) Window length = 5, (b) window length = 10, (c) window length = 15, (d) window length = 20, (e) window length = 25, (f) window length = 30, (g) window length = 35, and (h) window length = 40.

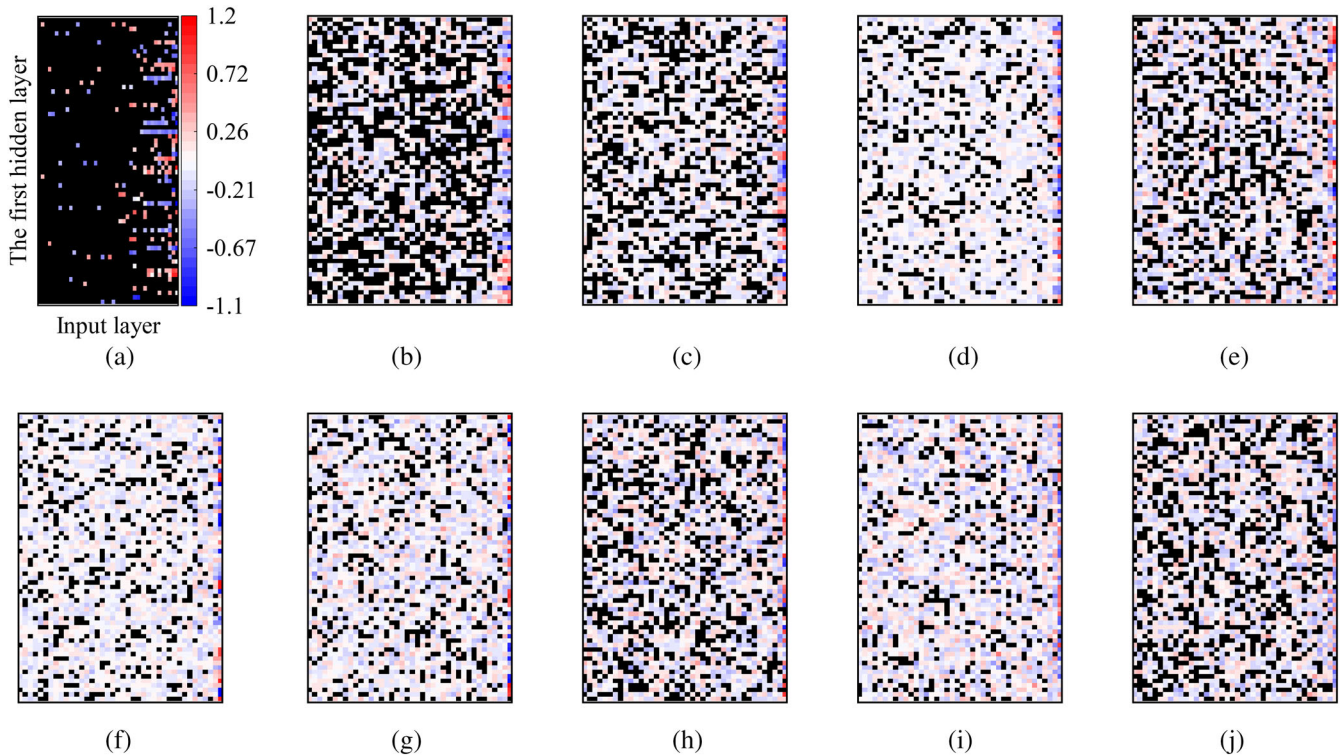


FIGURE 15 Visualization of weights between the input layer and the first hidden layer considering different noise levels for the single-sensor approach with a window length of 40. (a) Noise level 0%, (b) noise level 2%, (c) noise level 4%, (d) noise level 6%, (e) noise level 8%, (f) noise level 10%, (g) noise level 20%, (h) noise level 30%, (i) noise level 40%, and (j) noise level 50%.

40%, the iterations are terminated early, and the percentage of pruned weights (PPWs) is lower than expected. As a result, the PPWs does not exhibit a consistent decreasing then increasing trend. There are two main reasons for this irregular behavior. First, one is the random nature of training neural networks using stochastic gradient descent, where

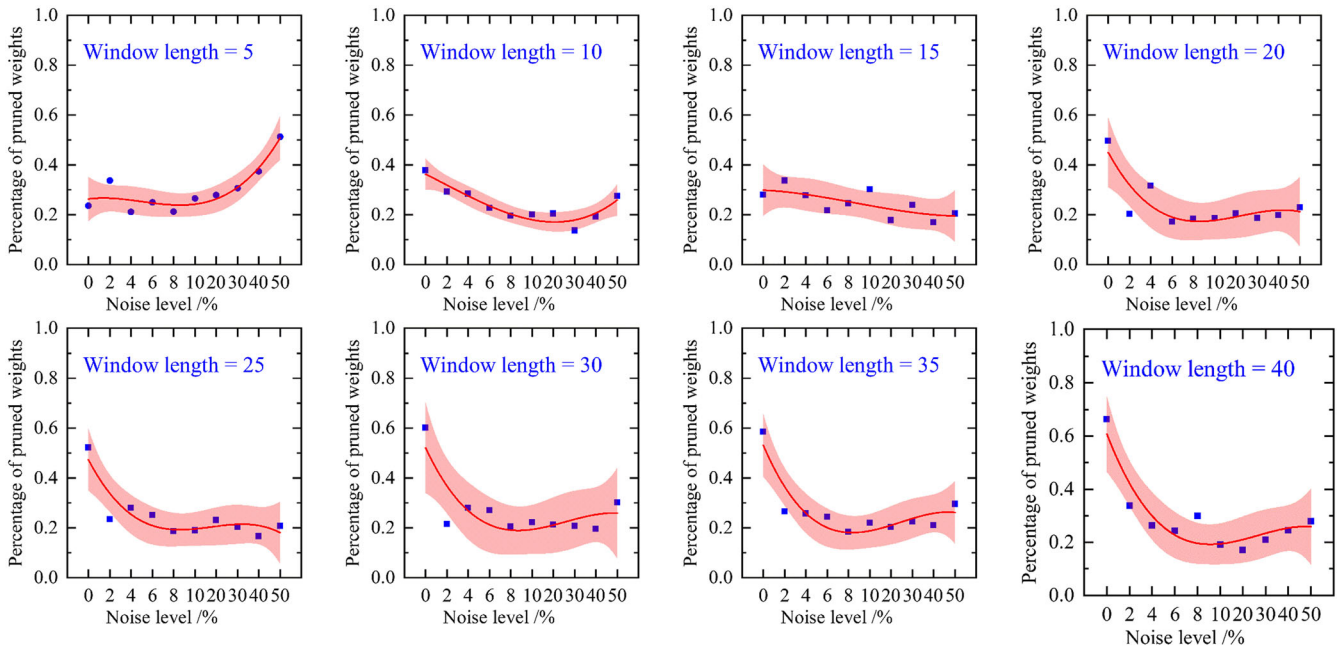


FIGURE 16 Percentage of pruned weights for the single-sensor approach.

the loss function may get stuck at a local minimum. Second, a restricted value of μ is used to obtain a better performance of pruned neural networks. This also increases the chance of early termination of the iteration loop caused by the first reason described above.

It is also observed that the percentage of weights being pruned is related to the noise level. Figure 16 shows the PPWs for each noise level and window length. At the low noise ranges (below 10%), the PPW has a trend of decreasing with the increasing noise level, indicating that the MLP is using more data to overcome the lack of information caused by the noise. However, this phenomenon is less obvious when the window length is small because few data points are being used. Although the noise levels are larger than 10%, the PPW, however, has an increasing trend with increasing noise levels. This indicates that the noise level is so high that the missing information cannot be restored by introducing more data, and this phenomenon is more pronounced when the window length is small.

For the multi-sensor approach, the optimal window length can be found by iteratively removing the low-ranked kernel elements in the first CL. When window length equals 20, the six kernels in the first CL are $2 \times 2 \times 20$ tensors, and the mean absolute values of these six kernels are calculated and presented in Figure 17. It is shown that an element of higher sequence in the kernel (closer to the current instant) corresponds to a higher mean absolute value, indicating that the algorithm is optimally assigning higher importance to the samples in an oversized window. When the noise level increases, the mean absolute values of the element of lower sequence in the kernel are increasing and make the curves smoother, indicating more samples are used by the algorithm to cope with the information loss caused by the noise, which is similar to the single-sensor approach.

3.5.5 | Discussion on the effectiveness of iterative pruning

The iterative training strategy is adopted in this research to tune the parameters of neural networks and remove the redundant connections in the mean time. One typical trend of training loss per epoch and per iteration when window length equals 40 is presented in Figure 18, where the algorithm is tuning the model parameters to minimize the validation loss until it stops dropping down for a few epochs (controlled by the parameter P_L) for each iteration. For the first epoch in each iteration, a jump of validation loss can be observed because of the removal of the connections. For the last epoch in each iteration, the validation losses are decreasing except for the last iteration, and the algorithm will take the results from iteration 10 as the optimal solution according to the flowchart presented in Figure 4. Hence, the prediction accuracy is improved compared with the unpruned iteration 1.

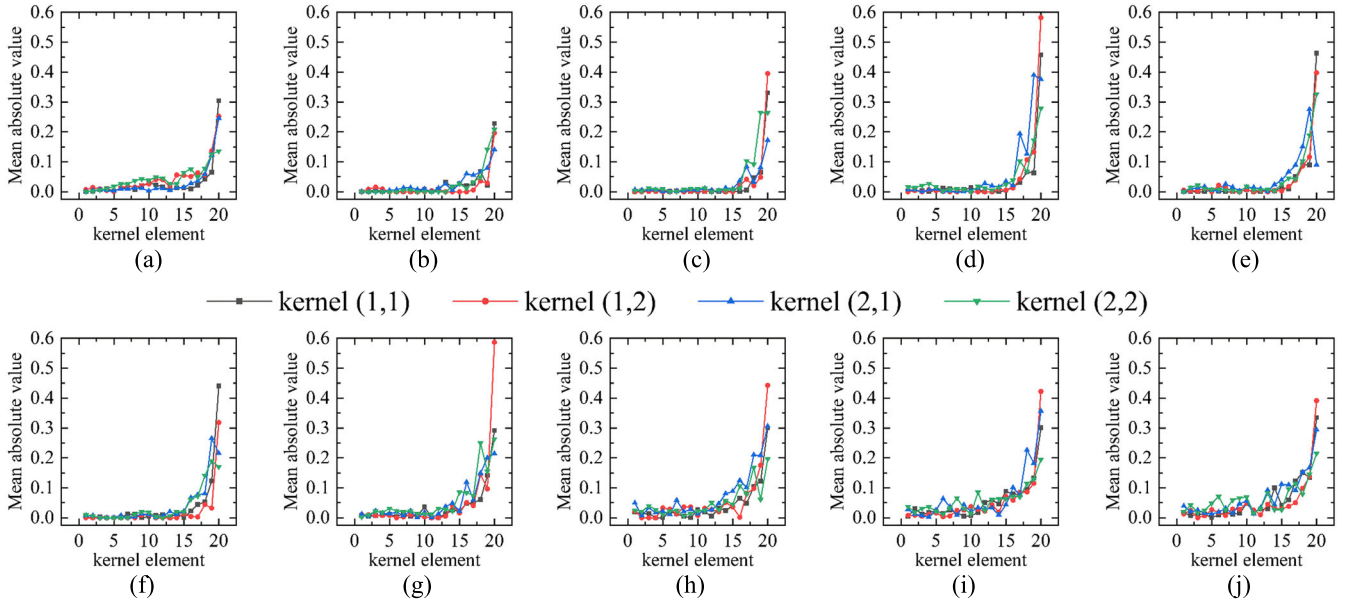


FIGURE 17 Mean absolute values of kernel elements in the first convolutional layer when window length equals 20. (a) Noise level 0%, (b) noise level 2%, (c) noise level 4%, (d) noise level 6%, (e) noise level 8%, (f) noise level 10%, (g) noise level 20%, (h) noise level 30%, (i) noise level 40%, and (j) noise level 50%.

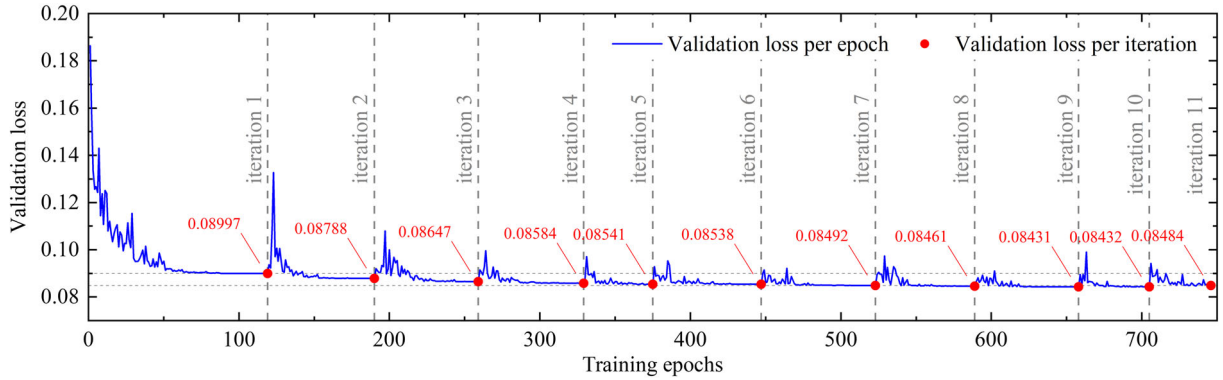


FIGURE 18 The trend of training loss per epoch and per iteration when window length equals 40.

The iterative pruning strategy is therefore effective in two aspects: (1) improving the prediction accuracy by reducing the loss between iterations and (2) reducing the computational effort by reducing the number of model parameters.

3.5.6 | Discussion on the execution time

Another important aspect of the real-time detection algorithm is the execution time. Because the decision maker is making decisions while new data are being collected by the sensors at the same time, the execution time of the decision maker between the two steps must be smaller than the sampling interval to ensure real-time processing. Otherwise, the decisions generated by the algorithm may introduce delays and may be left even days behind when small delays aggregates. Therefore, the sampling frequency should be carefully selected based on the algorithm's execution time. The sampling frequency should meet the requirement described in Equation (21)

$$f_s T_s \leq 1, \quad (21)$$

TABLE 3 The mean values of $f_s T_s$ calculated for single- and multi-sensor approaches ($\times 10^{-5}$)

Window length (samples)	5	10	15	20	25	30	35	40
Single-sensor	5.076	5.052	5.075	5.115	5.088	5.017	5.120	5.080
Multi-sensor	5.188	5.144	5.157	5.138	5.230	5.221	5.211	5.253

TABLE 4 Calculated MAD and MFAD/MAD for experimental dataset

Conditions	After Gilroy 40%	After Gilroy 67%	After Gilroy 100%	After Gilroy 120%
MAD	68.874	103.248	234.076	229.357
MFAD/MAD	0.0059	0.0039	0.0017	0.0018

Abbreviations: MAD, mean alarm density; MFAD, mean false alarm density.

where f_s is the sampling frequency of the algorithm, and T_s is the mean execution time of one single step. Using a personal computer equipped with AMD Ryzen 2700 processor and 32 GB RAM, the mean values of $f_s T_s$ are calculated for single- and multi-sensor approaches, respectively (Table 3). It can be observed that these values are much less than 1, indicating the algorithm can meet the real-time execution requirements.

4 | VERIFICATION BY USING THE EXPERIMENTAL DATASET

To further verify the proposed method with real-life damage conditions, experimental dataset from a three-story reinforced concrete frame is used to train and test the unsupervised damage detection algorithm. The experimental dataset is obtained from shake table tests conducted by the University of California, San Diego. The test frame was subjected to seismic ground motions of increasing intensity, and white-noise tests were conducted between the ground motion tests. More details on these experiments can be found in the dissertation of Staviridis,⁵⁶ and the data can be downloaded on Natural Hazards Engineering Research Infrastructure (NHERI).⁵⁷

The values of MAD and MFAD/MAD calculated after the test frame was excited by Gilroy ground motion scaled by 40%, 67%, 100%, and 120%, respectively, are presented in Table 4. It is shown that the MAD is increasing as the damage level raises with ground motion magnitude, except for Gilroy 120% where MAD value slightly decreases. Although this has no influence on the decision of damage existence, further investigations are still needed to deal with the imperfections of real-world data. The values of MFAD/MAD ratio remain much lower than 1 for all cases, indicating a good performance for distinguishing false alarms. The code for this experimental verification results is also accessible in the previous study.⁵⁸

5 | SUMMARY, CONCLUSIONS, AND FUTURE WORK

In this study, a novel unsupervised real-time damage detection method for structural health monitoring (SHM) is presented, which can effectively detect the damage occurrence and identify the damage level. The key contributions of the proposed method are summarized as follows:

1. A real-time damage-sensitive feature extraction method for single- and multi-channel SHM data by using neural networks with an iterative pruning strategy is presented. The iterative pruning strategy can prevent over-fitting by removing redundant connections between neurons caused by an oversized window. Compared with conventional method using matrix manipulations, the proposed method has lower computational complexity and can be used to generate damage-sensitive features in real time from a massive amount of data.
2. The DSVDD is first implemented in threshold generation for large-scale high-dimensional features extracted from monitoring data. Different from many supervised methods, DSVDD allows generating thresholds using healthy data only, which is rather important for the SHM of civil engineering applications.

3. An index named MAD is defined to measure the level of damages. By comparing the current value of MAD with historical values, a relative quantity of damage severity can be observed. Another index named MFAD is defined to measure the probability of generating false alarms by the proposed damage detection algorithm.
4. To investigate the capability of generating decisions faster than the sampling rate, the product of sampling frequency and mean execution time of one single step is calculated and compared with one to assess the capability of real-time execution of the proposed damage detection algorithm.
5. The effectiveness of the proposed method is verified using both benchmark simulation and experimental dataset.

With the dataset generated from the IASC-ASCE benchmark, the performance and effectiveness of the proposed method are carefully investigated. In order to prevent accidental results, conclusions are obtained by the analysis of 10 separate rounds of training and testing.

Main conclusions drawn are as follows:

1. The values of MAD for the single-sensor approach are found to be proportional to the damage level and inversely proportional to the noise level, indicating the algorithm's capability of identifying damage levels. The MAD values obtained from the multi-sensor approach under noise levels lower than 4% are significantly higher than those obtained from the single-sensor approach, indicating a higher performance for detecting minor asymmetric damages but a lower performance for identifying the damage patterns. With increasing noise levels, the trends of MAD obtained from these two approaches become similar, revealing that the multi-channel damage features are more vulnerable to the level of noise.
2. The values of MFAD are found to be lower than the value of MAD calculated for both the single- and multi-sensor approaches, indicating that true alarms can be accurately distinguished from false alarms. When the noise level is high and the damage level is low, the value of MFAD becomes closer to and even exceeds the value of MAD, indicating that the data are highly distorted and lightly damaged.
3. The visualization of the weights of the neural networks for the single- and multi-sensor approaches shows that the redundant weights can be effectively removed during training to prevent potential over-fitting from an oversized window and model structure. When the noise level is low, the algorithm removes more weights because it is easier to find the most important weights than the case when the noise level is high, which indicates that the algorithm can find the optimal statistical model structure and window length under different levels of noise.
4. The speed of generating the decisions by the algorithm is found to be much higher than the data sampling speed, which ensures the real-time execution of the proposed damage detection algorithm.

Despite the effectiveness and advantages of the proposed method, the following needs to be further studied:

1. More detailed investigations will be conducted with experimental and real-life monitoring data.
2. As this method is established on the detection of changes in the dynamic response data, if the environmental and operational variabilities are ergodic through time, then the changes introduced by these factors can be eliminated by a proper window length, which means that the current method is supposed to work in this case. But when the environmental and operational variabilities are non-ergodic or the time span is too long, the changes introduced by these factors will trigger false alarms. In this case, these factors should be added to the input layer of neural networks, so that correlations can be built between these factors and dynamic responses.
3. From the results of MAD obtained from the multi-channel data, it was observed that despite the high performance for detecting asymmetric damage patterns, the introduction of channels with high noise-to-signal ratio caused the performance to deteriorate. Therefore, further improvements can be made to the optimal selection of data channels to adaptively remove the low-ranked channels during training. This is a topic of future research.

ACKNOWLEDGMENTS

This research was funded by a scholarship from the China Scholarship Council (CSC) and the National Natural Science Foundation of China (grant no. 51808291), whose support is sincerely appreciated.

AUTHOR CONTRIBUTIONS

Sheng Shi: conceptualization, investigation, methodology, software, data curation, writing-original draft, writing-review and editing, and funding acquisition. **Dongsheng Du:** conceptualization, writing-original draft,

supervision, and funding acquisition. **Oya Mercan:** writing-original draft, writing-review and editing, and supervision. **Erol Kalkan:** writing-original draft, writing-review and editing, and supervision. **Shuguang Wang:** funding acquisition and supervision.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Sheng Shi  <https://orcid.org/0000-0001-9047-4709>

REFERENCES

1. Farrar CR, Worden K. An introduction to structural health monitoring. *Philos Trans R Soc A Math Phys Eng Sci.* 2007;365(1851):303-315. doi:[10.1098/rsta.2006.1928](https://doi.org/10.1098/rsta.2006.1928)
2. Annamdas VGM, Bhalla S, Soh CK. Applications of structural health monitoring technology in Asia. *Struct Health Monit.* 2017;16(3):324-346. doi:[10.1177/1475921716653278](https://doi.org/10.1177/1475921716653278)
3. Ou J, Li H. Structural health monitoring in mainland China: review and future trends. *Struct Health Monit.* 2010;9(3):219-231. doi:[10.1177/1475921710365269](https://doi.org/10.1177/1475921710365269)
4. Farrar CR, Doebling SW, Nix DA. Vibration-based structural damage identification. *Philos Trans Math Phys Eng Sci.* 2001;359(1778):131-149. doi:[10.1098/rsta.2000.0717](https://doi.org/10.1098/rsta.2000.0717)
5. Farrar CR, Worden K. *Structural health monitoring: a machine learning perspective.* Chichester, UK: John Wiley & Sons, Ltd; 2012. doi:[10.1002/9781118443118.ch1](https://doi.org/10.1002/9781118443118.ch1).
6. Allemang RJ, Brown DL. Correlation coefficient for modal vector analysis. *Proc Int Modal Anal Conf Exhib.* 1982;110-116.
7. Yamaguchi H, Matsumoto Y, Kawai K, Dammika AJ, Saeed Shahzad RT. Damage detection based on modal damping change in bridges, Kandy, Sri Lanka: 2013, p. 14-16.
8. Williams C, Salawu OS. Damping as a damage indication parameter. *Proc Int Modal Anal Conf - IMAC.* 1997;2, Tokyo, Japan:1531-1536.
9. Pandey AK, Biswas M, Samman MM. Damage detection from changes in curvature mode shapes. *J Sound Vib.* 1991;145(2):321-332. doi:[10.1016/0022-460X\(91\)90595-B](https://doi.org/10.1016/0022-460X(91)90595-B)
10. Shi ZY, Law SS, Zhang LM. Structural damage localization from modal strain energy change. *J Sound Vib.* 1998;218(5):825-844. doi:[10.1006/jsvi.1998.1878](https://doi.org/10.1006/jsvi.1998.1878)
11. Pandey AK, Biswas M. Damage detection in structures using changes in flexibility. *J Sound Vib.* 1994;169(1):3-17. doi:[10.1006/jsvi.1994.1002](https://doi.org/10.1006/jsvi.1994.1002)
12. Peeters B, De RG. Stochastic system identification for operational modal analysis: a review. *J Dyn Syst Meas Control Trans ASME.* 2001;123(4):659-667. doi:[10.1115/1.1410370](https://doi.org/10.1115/1.1410370)
13. Li Z, Feng MQ, Luo L, Feng D, Xu X. Statistical analysis of modal parameters of a suspension bridge based on Bayesian spectral density approach and SHM data. *Mech Syst Signal Process.* 2018;98:352-367. doi:[10.1016/j.ymssp.2017.05.005](https://doi.org/10.1016/j.ymssp.2017.05.005)
14. Sohn H, Czarnecki JA, Farrar CR. Structural health monitoring using statistical process control. *J Struct Eng.* 2000;126(11):1356-1363. doi:[10.1061/\(asce\)0733-9445\(2000\)126:11\(1356](https://doi.org/10.1061/(asce)0733-9445(2000)126:11(1356)
15. Entezami A, Shariatmadar H. Structural health monitoring by a new hybrid feature extraction and dynamic time warping methods under ambient vibration and non-stationary signals. *Meas J Int Meas Confed.* 2019;134:548-568. doi:[10.1016/j.measurement.2018.10.095](https://doi.org/10.1016/j.measurement.2018.10.095)
16. Cha YJ, Wang Z. Unsupervised novelty detection-based structural damage localization using a density peaks-based fast clustering algorithm. *Struct Health Monit.* 2018;17(2):313-324. doi:[10.1177/1475921717691260](https://doi.org/10.1177/1475921717691260)
17. Sarmadi H, Entezami A, Daneshvar Khorram M. Energy-based damage localization under ambient vibration and non-stationary signals by ensemble empirical mode decomposition and Mahalanobis-squared distance. *J Vib Control.* 2020;26(11-12):1012-1027. doi:[10.1177/1077546319891306](https://doi.org/10.1177/1077546319891306)
18. Kumar K, Biswas PK, Dhang N. Time series-based SHM using PCA with application to ASCE benchmark structure. *J Civ Struct Health Monit.* 2020;10(5):899-911. doi:[10.1007/s13349-020-00423-2](https://doi.org/10.1007/s13349-020-00423-2)
19. Santos JP, Cr mona C, Calado L, Silveira P, Orcesi AD. On-line unsupervised detection of early damage. *Struct Control Health Monit.* 2016;23(7):1047-1069. doi:[10.1002/stc.1825](https://doi.org/10.1002/stc.1825)
20. Neves AC, Gonz lez I, Leander J, Karoumi R. Structural health monitoring of bridges: a model-free ANN-based approach to damage detection. *J Civ Struct Heal Monit.* 2017;7(5):689-702. doi:[10.1007/s13349-017-0252-5](https://doi.org/10.1007/s13349-017-0252-5)
21. Khodabandehlou H, Pekcan G, Fadali MS. Vibration-based structural condition assessment using convolution neural networks. *Struct Control Health Monit.* 2019;26:1-12. doi:[10.1002/stc.2308](https://doi.org/10.1002/stc.2308)
22. Rafiei MH, Adeli H. A novel unsupervised deep learning model for global and local health condition assessment of structures. *Eng Struct.* 2018;156:598-607. doi:[10.1016/j.engstruct.2017.10.070](https://doi.org/10.1016/j.engstruct.2017.10.070)
23. Ni FT, Zhang J, Noori MN. Deep learning for data anomaly detection and data compression of a long-span suspension bridge. *Comput Civ Infrastruct Eng.* 2020;35(7):685-700. doi:[10.1111/mice.12528](https://doi.org/10.1111/mice.12528)
24. Chaabane M, Mansouri M, Ben Hamida A, Nounou H, Nounou M. Multivariate statistical process control-based hypothesis testing for damage detection in structural health monitoring systems. *Struct Control Health Monit.* 2019;26(1):1-14. doi:[10.1002/stc.2287](https://doi.org/10.1002/stc.2287)

25. Sajedi SO, Liang X. Vibration-based semantic damage segmentation for large-scale structural health monitoring. *Comput-Aided Civ Infrastruct Eng*. 2020;35(6):579-596. doi:[10.1111/mice.12523](https://doi.org/10.1111/mice.12523)
26. Azimi M, Pekcan G. Structural health monitoring using extremely compressed data through deep learning. *Comput-Aided Civ Infrastruct Eng*. 2020;35(6):597-614. doi:[10.1111/mice.12517](https://doi.org/10.1111/mice.12517)
27. Yang J, Zhang L, Chen C, et al. A hierarchical deep convolutional neural network and gated recurrent unit framework for structural damage detection. *Inf Sci (Ny)*. 2020;540:117-130. doi:[10.1016/j.ins.2020.05.090](https://doi.org/10.1016/j.ins.2020.05.090)
28. Wang Z, Cha YJ. Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage. *Struct Health Monit*. 2021;20(1):406-425. doi:[10.1177/1475921720934051](https://doi.org/10.1177/1475921720934051)
29. Entezami A, Shariatmadar H. An unsupervised learning approach by novel damage indices in structural health monitoring for damage localization and quantification. *Struct Health Monit*. 2018;17(2):325-345. doi:[10.1177/1475921717693572](https://doi.org/10.1177/1475921717693572)
30. Eltouny KA, Liang X. Bayesian-optimized unsupervised learning approach for structural damage detection. *Comput-Aided Civ Infrastruct Eng*. 2021;36(10):1249-1269. doi:[10.1111/mice.12680](https://doi.org/10.1111/mice.12680)
31. Modesto AJ, Birgul R, Werlink RJ, Catbas FN. Damage detection of composite overwrapped pressure vessels using ARX models. *Int J Press Vessel pip*. 2021;192:104410. doi:[10.1016/j.ijpvp.2021.104410](https://doi.org/10.1016/j.ijpvp.2021.104410)
32. Farrar CR, Lieven NAJ. Damage prognosis: the future of structural health monitoring. *Philos Trans R Soc a Math Phys Eng Sci*. 2007;365(1851):623-632. doi:[10.1098/rsta.2006.1927](https://doi.org/10.1098/rsta.2006.1927)
33. Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc; 2019;8024-8035.
34. Farrar CR, Worden K. Structural Health Monitoring: A Machine Learning Perspective. *Struct Health Monit A Mach Learn Perspect*. 2012;439. doi:[10.1002/9781118443118](https://doi.org/10.1002/9781118443118)
35. Haq A, Munir W. New CUSUM and Shewhart-CUSUM charts for monitoring the process mean. *Qual Reliab Eng Int*. 2021;37(8):3512-3528. doi:[10.1002/qre.2930](https://doi.org/10.1002/qre.2930)
36. Xu L, Wang S, Reynolds MR. A generalized likelihood ratio control chart for monitoring the process mean subject to linear drifts. *Qual Reliab Eng Int*. 2013;29(4):545-553. doi:[10.1002/qre.1404](https://doi.org/10.1002/qre.1404)
37. Basseville M, Nikiforov IV. *Detection of abrupt changes - theory and application*. Prentice Hall; 1993.
38. Devauchelle-Gach B, Basseville M, Benveniste A. Diagnosing mechanical changes in vibrating systems. *IFAC Proc Vol*. 1991;24(6):159-163. doi:[10.1016/s1474-6670\(17\)51135-0](https://doi.org/10.1016/s1474-6670(17)51135-0)
39. Wahnou E, Benveniste A, El Ghaoui L, Nikoukhah R. An optimum robust approach to statistical failure detection and identification. *Proc IEEE Conf Decis control*. 1992;650-655. doi:[10.1109/cdc.1991.261390](https://doi.org/10.1109/cdc.1991.261390)
40. Hastie T, Tibshirani R, Friedman J. *The elements of statistical learning*. 2nd ed. New York, NY: Springer New York; 2009. doi:[10.1007/978-0-387-84858-7](https://doi.org/10.1007/978-0-387-84858-7)
41. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by back-propagating errors. *Nature*. 1986;323(6088):318-362. doi:[10.1038/323533a0](https://doi.org/10.1038/323533a0)
42. ichi AS. Backpropagation and stochastic gradient descent method. *Neurocomputing*. 1993;5(4-5):185-196. doi:[10.1016/0925-2312\(93\)90006-O](https://doi.org/10.1016/0925-2312(93)90006-O)
43. Goodfellow I, Bengio Y, Courville A. *Deep learning - adaptive computation and machine learning*. Massachusetts, USA: 2017.
44. LeCun Y, B B, D JS, et al. Backpropagation applied to handwritten zip code recognition. *Neural Comput*. 1989;1(4):541-551. doi:[10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541)
45. Koene RA, Takane Y. Discriminant component pruning: regularization and interpretation of multilayered backpropagation networks. *Neural Comput*. 1999;11(3):783-802. doi:[10.1162/089976699300016665](https://doi.org/10.1162/089976699300016665)
46. Jorgensen TD, Haynes BP, Norlund CCF. Pruning artificial neural networks using neural complexity measures. *Int J Neural Syst*. 2008;18(05):389-403. doi:[10.1142/S012906570800166X](https://doi.org/10.1142/S012906570800166X)
47. Wang Y, Zhang X, Xie L, Zhou J, Su H, Zhang B, Hu X. Pruning from scratch. *AAAI 2020-34th AAAI Conf Artif Intell 2020*;1:12273-12280. doi:[10.1609/aaai.v34i07.6910](https://doi.org/10.1609/aaai.v34i07.6910), 07.
48. Ruder S. An overview of gradient descent optimization algorithms. *ArXiv Prepr* 2016.
49. Kingma DP, Ba JL. Adam: a method for stochastic optimization. 3rd Int Conf learn represent ICLR 2015 - Conf track proc 2015:1-15.
50. Zhou HF, Ni YQ, Ko JM. Structural damage alarming using auto-associative neural network technique: exploration of environment-tolerant capacity and setup of alarming threshold. *Mech Syst Signal Process*. 2011;25(5):1508-1526. doi:[10.1016/j.ymssp.2011.01.005](https://doi.org/10.1016/j.ymssp.2011.01.005)
51. Tax DMJ, Duin RPW. Support vector domain description. *Pattern Recognit Lett*. 1999;20(11-13):1191-1199. doi:[10.1016/S0167-8655\(99\)00087-2](https://doi.org/10.1016/S0167-8655(99)00087-2)
52. GhasemiGol M, Monsefi R, Yazdi HS. Support vector data description. *Commun Comput Inf Sci*. 2009;43 CCIS:257-268. doi:[10.1007/978-3-642-03969-0_24](https://doi.org/10.1007/978-3-642-03969-0_24)
53. Ruff L, Vandermeulen RA, Görnitz N, Deecke L, Siddiqui SA, Binder A, et al. Deep one-class classification. 35th Int Conf Mach Learn ICML 2018 2018;10:6981-6996.
54. Johnson EA, Lam HF, Katafygiotis LS, Beck JL. Phase I IASC-ASCE structural health monitoring benchmark problem using simulated data. *J Eng Mech*. 2004;130(1):3-15. doi:[10.1061/\(asce\)0733-9399\(2004\)130:1\(3](https://doi.org/10.1061/(asce)0733-9399(2004)130:1(3)
55. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. *ICML Work Deep Learn Audio, Speech Lang Process* 2013;28.

56. Stavridis A. *Analytical and experimental study of seismic performance of reinforced concrete frames infilled with masonry walls*. San Diego: University of California; 2009.
57. <https://www.designsafe-ci.org/data/browser/public/nees.public/NEES-2007-0422.groups>
58. <https://github.com/tmshisheng/Experimental-verification.git>

How to cite this article: Shi S, Du D, Mercan O, Kalkan E, Wang S. A novel unsupervised real-time damage detection method for structural health monitoring using machine learning. *Struct Control Health Monit*. 2022; e3042. doi:[10.1002/stc.3042](https://doi.org/10.1002/stc.3042)