

Research Article

Unsupervised Structural Damage Detection Technique Based on a Deep Convolutional Autoencoder

Zahra Rastin ¹, Gholamreza Ghodrati Amiri ¹ and Ehsan Darvishan ²

¹Natural Disasters Prevention Research Center, School of Civil Engineering, Iran University of Science & Technology, Tehran, Iran

²Department of Civil Engineering, Roudehen Branch, Islamic Azad University, Roudehen, Iran

Correspondence should be addressed to Gholamreza Ghodrati Amiri; ghodrati@iust.ac.ir

Received 9 November 2020; Revised 21 March 2021; Accepted 13 April 2021; Published 23 April 2021

Academic Editor: Salvatore Caddemi

Copyright © 2021 Zahra Rastin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Structural health monitoring (SHM) is a hot research topic with the main purpose of damage detection in a structure and assessing its health state. The major focus of SHM studies in recent years has been on developing vibration-based damage detection algorithms and using machine learning, especially deep learning-based approaches. Most of the deep learning-based methods proposed for damage detection in civil structures are based on supervised algorithms that require data from the healthy state and different damaged states of the structure in the training phase. As it is not usually possible to collect data from damaged states of a large civil structure, using such algorithms for these structures may be impractical. This paper proposes a new unsupervised deep learning-based method for structural damage detection based on convolutional autoencoders (CAEs). The main objective of the proposed method is to identify and quantify structural damage using a CAE network that employs raw vibration signals from the structure and is trained by the signals solely acquired from the healthy state of the structure. The CAE is chosen to take advantage of high feature extraction capability of convolution layers and at the same time use the advantages of an autoencoder as an unsupervised algorithm that does not need data from damaged states in the training phase. Applications on the two numerical models of IASC-ASCE benchmark structure and a grid structure located at the University of Central Florida, as well as the full-scale Tianjin Yonghe Bridge, prove the efficiency of the proposed algorithm in assessing the global health state of the structures and quantifying the damage.

1. Introduction

Civil infrastructures including buildings and bridges are valuable assets necessary for every society to function well. They are susceptible to damage due to different factors such as material aging, environmental corrosion, poor construction quality, or extreme events such as earthquakes [1]. Therefore, these structures need to be checked regularly to detect damage in the early stage and prevent their propagation through the structure. According to the large size of civil structures, visual inspection is not an efficient way as it is time-consuming, laborious, and dependent on the experience of the inspector. SHM techniques provide a practical automatic means of detecting, locating, and quantifying damage in a structure and assessing its health state under operational conditions. Thus, these techniques

have attracted considerable attention from researchers and engineers [2].

A major focus of SHM studies in recent years has been devoted to developing vibration-based monitoring techniques. The idea behind these techniques is that damage-induced changes in structural properties such as stiffness will change the measured vibration response of the structure [3]. Machine learning techniques have been extensively used in vibration-based SHM of engineering structures during the last two decades, because of their ability to cope with uncertainty and noise [4]. Among these techniques, feed-forward artificial neural networks (ANNs) are the most common ones.

In Yun et al. [5], the joint damage in a steel frame was estimated using an ANN model and employing modal data as input. Three techniques were used to improve the

performance of ANNs: substructural identification, noise injection learning, and the data perturbation scheme. Lee et al. [6] presented an ANN-based method for a structural damage assessment at element level considering the modeling errors in the finite element (FE) model that was used to generate training samples. The mode shape variations before and after damage were inputted to the ANNs as they are less sensitive to the modeling errors than the mode shapes themselves. Li et al. [7] proposed a damage identification method utilizing changes in frequency response functions (FRFs) and ANNs. Principal component analysis (PCA) techniques were employed to extract damage features and obtain suitable patterns for ANN inputs. Osornio-Rios et al. [8] used multiple signal classification (MUSIC) fused with ANNs for damage detection, localization, and quantification in a truss-type structure. The vibration signals obtained from the structure were first preprocessed using the MUSIC algorithm and then inputted to an ANN to evaluate the health state of the structure. Bandara et al. [9] introduced a structural damage identification procedure by combining FRFs, ANNs, and PCA. PCA was used to compress the initial FRF data and transform it into new damage indices. An ANN was then utilized to locate and quantify the damage. Another example is the method proposed by Abdeljaber and Avci [10]. This method uses self-organizing maps to extract damage indices from the vibration response of the structure. The damage indices are then processed using an ANN to conduct damage detection.

Other machine learning techniques such as support vector machine (SVM) [11–13], fuzzy neural network (FNN) [14, 15], optimization algorithms such as genetic algorithm (GA) [16–19], multiverse optimizer (MVO) algorithm [20], and grey wolf optimization (GWO) algorithm [21] have also been used in SHM and damage detection of civil structures.

When using traditional machine learning algorithms, the data samples need to be preprocessed to extract certain features representing the main characteristics of the data. The features are then inserted into the algorithm. The efficiency of these algorithms highly depends on the selection of extracted features, though it is not a trivial task to choose the right group of features representing the main properties of the input data for the machine learning algorithm to perform well [22]. As a subset of machine learning, deep learning-based approaches were introduced to address this problem. Among deep learning-based algorithms, convolutional neural networks (CNNs) have recently gained popularity in damage detection applications due to their great success in feature extraction and classification taking raw data as input, which is a result of using convolution layers. Abdeljaber et al. [23] used one-dimensional (1D) CNNs to detect loosened bolts in a steel frame. Lin et al. [24] proposed the use of CNNs for feature extraction and classification to detect structural damage. Gulgec et al. [25] designed a CNN topology to classify simulated damaged and healthy cases and localize the damage. Ma et al. [26] offered a transfer learning-CNN based on AlexNet for bearing fault diagnosis in noisy environment. Cofre-Martel et al. [27] presented a CNN-based approach to localize and quantify structural damage. Li et al. [28] employed a fully

convolutional neural network together with a Naïve Bayes data fusion to recognize cracks in concrete bridges. Guo et al. [29] combined wavelet transform and deformable CNN for bearing fault diagnosis.

Most of the deep learning-based methods proposed for SHM purposes are based on supervised learning algorithms. These algorithms need labeled data from healthy and various damaged states of the monitored structure which makes them impractical for civil infrastructures, because data from different damaged states of these structures are not usually accessible. Unsupervised deep learning-based approaches have recently been proposed to overcome this limitation. In Rafiei and Adeli [30], a deep restricted Boltzmann machine was used to extract features from a set of preprocessed signals obtained from the healthy state of a small-scale reinforced concrete building and another set of preprocessed signals with unknown health states. Synchrosqueezed wavelet transform and fast Fourier transform were used to process raw acceleration signals. The extracted features are then used to assess the health conditions of the unknown state. Silva et al. [31] presented a damage detection method based on deep PCA. This approach was applied on a progressively damaged prestressed concrete bridge and a three-span suspension bridge. Hsu et al. [32] trained an autoencoder network using natural frequencies extracted from vibration signals acquired from a dam under different environmental conditions to continuously monitor the health state of the dam. Wang and Cha [33] presented an unsupervised deep learning-based method for structural damage detection which uses a deep autoencoder with a one-class SVM to detect the presence of damage in civil structures using data from the healthy state of the structure as training samples. Ma et al. [34] used variational autoencoders (VAEs) for damage detection and localization task of a bridge under a moving vehicle. The presented method did not need the baseline data; VAE was trained using only data from a damaged structure.

This paper proposes a new unsupervised deep learning-based algorithm for structural damage detection based on CAEs. The main objective of the proposed method, which makes it different from the current researches, is to identify and quantify structural damage using raw vibration signals from the structure and training the CAE network using only signals acquired from the structure in the healthy state. Also, unlike the majority of existing methods that use complex algorithms, the proposed method is simple and easy to implement reducing the need for experts to interpret the results. The CAE is chosen to take advantage of high feature extraction capability of convolution layers and at the same time use the advantages of an autoencoder as an unsupervised algorithm that does not need data from damaged states in the training phase and thus can be used for health monitoring of real-life civil structures. Applications on the two numerical models of IASC-ASCE benchmark structure and a grid structure located at the University of Central Florida, as well as the full-scale Tianjin Yonghe Bridge, prove the efficiency of the proposed algorithm in assessing the global health state of the structures and quantifying the damage.

The rest of this paper is organized as follows: A brief overview of autoencoders is presented in Section 2, the proposed methodology is introduced in Section 3, applications on the above-mentioned structures and obtained results are described in Section 4, and finally, the last section includes concluding remarks regarding this paper.

2. Overview of Autoencoders

An autoencoder is a kind of feed-forward ANN that is basically used to form useful representations of the input data by combining their features in nonlinear ways [35]. Like other neural networks, learning in autoencoders is usually done using back-propagation and gradient descendant-based optimizers. Autoencoders are trained in an unsupervised manner and thus do not need damaged samples in the training phase.

The main objective of most autoencoders is to compress the data and reduce their dimensionality in a way that only their main features are preserved (feature extraction). Compressed data can be analyzed easier and faster and with less computational burden. Figure 1 shows an example of an autoencoder network. As can be seen in this figure, an autoencoder consists of two parts, namely, encoder and decoder. The encoder part lowers the dimensions of the input data and outputs their compressed representations, while the decoder part tries to rebuild the input data from compressed representations, so the output of an autoencoder is supposed to be the same as its input. The size of the compressed data can be controlled by the number of neurons in the last layer of the encoder.

In order to use the advantages of both autoencoders and CNNs, CAEs are used in this study, which usually use convolution and pooling layers to extract the key features of the input data and compress them (encoding) and deconvolution and unpooling layers to reconstruct the original data from the compressed form (decoding) [36]. Figure 2 shows an illustrative example of the structure of a two-dimensional (2D) CAE. The CAEs employed in this study are trained using data from the healthy state of the structure. After training, the encoder can extract features from a damaged state that are different from those of the healthy state of the structure.

3. Methodology

The proposed damage detection methodology contains three steps: in the first step, the acceleration data from the structure are preprocessed and made ready to enter the CAE; then, the network is trained using data from the healthy state of the structure; at the end, in the third step, the encoder part of the trained network is used to compress the input data by extracting their main features and the damage is detected by comparing the features extracted from a damaged state data with those extracted from the healthy state data. These steps are shown in Figure 3 and explained in the following subsections.

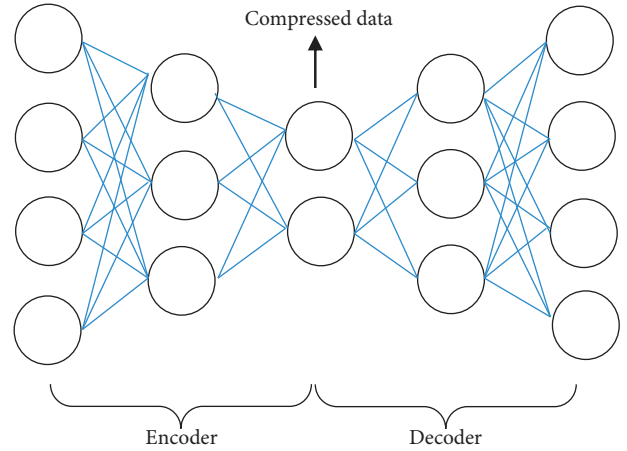


FIGURE 1: An example of an autoencoder network.

3.1. First Step: Preprocessing. In this step, the data from n accelerometers installed on the structure are concatenated to form an n -column matrix. Then, the values in this matrix are normalized between -1 and 1 . This matrix is divided into smaller matrices which can be used as inputs to the network.

3.2. Second Step: Training of the CAEs. In the second step, the CAEs are trained using preprocessed healthy data. In this study, all convolution layers of a CAE use Rectified Linear Unit (ReLU) activation function, except the last one that uses the Sigmoid function. Mean Square Error (MSE) is chosen for reconstruction loss function which is minimized using Adam optimizer.

3.3. Third Step: Data Compression and Damage Detection. In the last step, the encoder part of the network trained in the second step is used to compress input matrices into 1D vectors. Data from healthy and unknown states of the structure are fed into the CAE separately to form reference and test vectors, respectively. Every vector V is then converted to a unit vector UV , by dividing its components by its norm as shown in the following equations:

$$V = \{v_1, v_2, v_3, \dots, v_n\}, \quad (1)$$

$$UV = \{v'_1, v'_2, v'_3, \dots, v'_n\}, \quad (2)$$

where

$$v'_i = \frac{v_i}{\sqrt{v_1^2 + v_2^2 + \dots + v_n^2}}. \quad (3)$$

The distance between a unit test vector $UU = \{u'_1, u'_2, u'_3, \dots, u'_n\}$ and a unit reference vector UV , each of length n , can be calculated using

$$D = \sqrt{\sum_{i=1}^n (u'_i - v'_i)^2}. \quad (4)$$

As the damage in the structure increases, the vectors extracted from its acceleration data take distance from the reference vectors.

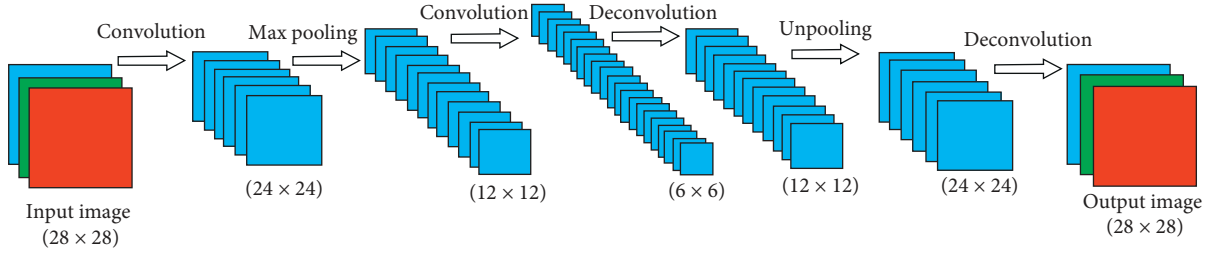


FIGURE 2: An example of the structure of a 2D CAE.

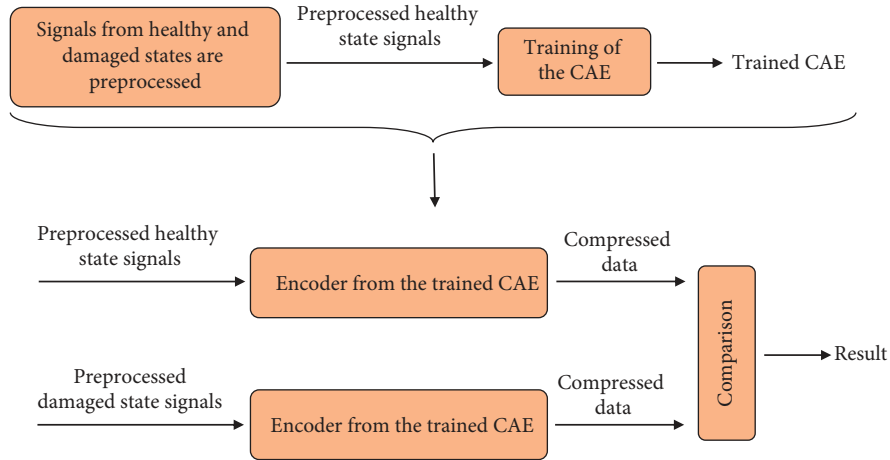


FIGURE 3: Procedure of the proposed damage detection method.

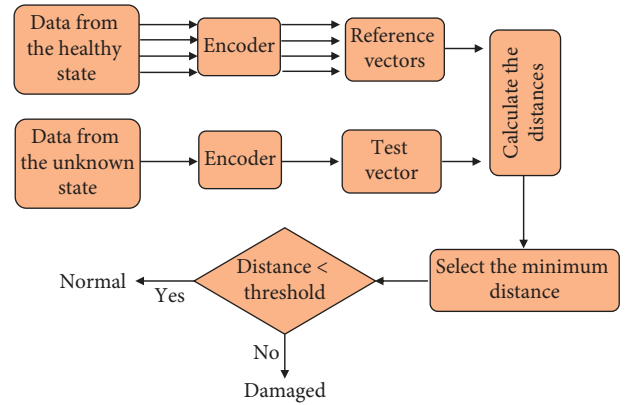
When data from the healthy state of a structure are fed to a CAE in the form of n 2D matrices, the encoder produces n 1D vectors from them (reference vectors). In the same way, we will have m 1D vectors for m matrices from an unknown state of the structure (test vectors). The distances between each of these m vectors and every one of the reference vectors are computed and among the n obtained values, the minimum value is chosen as the final distance of that test vector from reference vectors. If this distance is greater than a specific threshold, the related test vector is classified as damaged. Figure 4 shows the classification process for $m = 1$. The damage percentage for an unknown state of a structure is computed as follows:

$$\text{damage percentage} = \frac{\text{number of damaged vectors}}{m} \times 100. \quad (5)$$

For any structure of interest, the aforementioned threshold is determined in a way that the resulting damage percentage for the healthy state data is reasonably low.

4. Case Studies and Results

To study the efficiency of the proposed method, it is applied for damage detection in two numerical models and a real-world structure. All calculations are implemented in Python

FIGURE 4: Classification of the vector obtained from a 2D matrix from an unknown state of the structure ($m = 1$).

3.6.9. The damage detection steps for each of the structures along with the obtained results are discussed in this section.

4.1. Phase I IASC-ASCE Benchmark Structure. Phase I SHM benchmark structure was established by the International Association for Structural Control (IASC)-American Society of Civil Engineers (ASCE) Structural Health Monitoring Task group in order to provide a uniform test case for validating various structural damage detection techniques. Simulated acceleration data from an analytical model

developed for this experimental structure, published by this task group in 2004, were used in this study. The benchmark frame is a four-story, two-bay by two-bay model structure built at the University of British Columbia. It has a $2.5\text{ m} \times 2.5\text{ m}$ plan and is 3.6 m tall. Two diagonal braces were installed on each floor of each exterior face. Figure 5 shows the diagram of this structure. Two FE models were developed based on this building. The second one, which is considered in this study, is a 120-degree-of-freedom (DOF) model. In addition to the healthy state, six damage patterns were studied for this structure. These patterns are described in Table 1 and shown graphically in Figure 6 [37].

4.1.1. Preprocessing. In this study, 2 accelerometers per floor are considered to record acceleration data in y direction at a sampling frequency of 200 Hz under independent loading in y direction at each floor. For each state, the data from all 8 sensors are concatenated to form an 8-column matrix. After normalization, these matrices are divided into smaller 128×8 matrices. In the end, the healthy state matrices are shuffled before being used to train the network.

4.1.2. Training of the CAE. The number of 128×8 matrices from the healthy state data needed in the training stage for the network to perform well is obtained equal to 2000 through trial and error. The encoder part of the network used for this structure has two 1D convolution layers, with filter counts of 4 and 1, filter size of 5, and the same padding. A 1D max pooling layer with a window size of 4 is used after each convolution layer. The decoder part uses three 1D convolution layers with filter counts of 1, 4, and 8, filter size of 5, and the same padding and 1D upsampling layers with a window size of 4 after the first two convolution layers.

4.1.3. Data Compression and Damage Detection. The encoder part of the CAE trained in the previous step is used to compress the input data by removing redundant information and keeping only the most important features. 1000 matrices from the healthy state data are inputted to the encoder and outputted in the form of 1D vectors of length 8 which are used as reference vectors. A parametric study is then performed in order to find the minimum number of input matrices from an unknown state needed to reach a suitable damage percentage. Damage detection is carried out with (a) 50, (b) 100, (c) 200, (d) 300, (e) 400, and (f) 500 input matrices from each state (the healthy state and the six damaged states). These matrices are fed to the encoder to obtain test vectors of the same length (length 8). It is worth mentioning that the healthy state matrices used to gain the test vectors are different from those employed for training or acquiring reference vectors.

After the distances are calculated, the mean plus 1.6 standard deviations of the set of distances obtained for healthy state data is set as the damage threshold. The damage percentage for each state is calculated by dividing the number of vectors whose distances are greater than this threshold by the total number of vectors. The damage

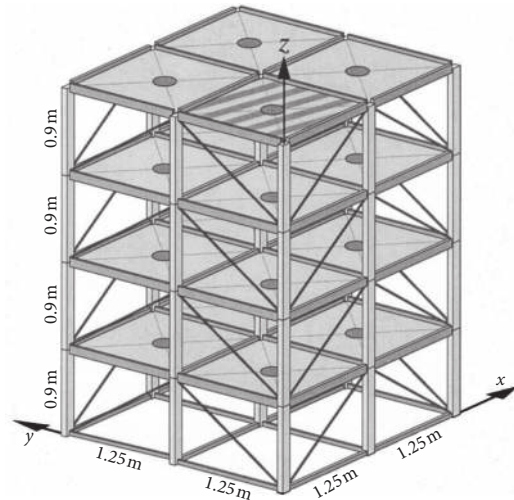


FIGURE 5: Diagram of IASC-ASCE benchmark structure [37].

percentage in the structure under the six damage patterns and for the healthy state is shown in Table 2 for cases (a) to (f).

From the table, it is obvious that at least 400 matrices (case (e)) are needed for appropriate damage detection. Considering case (e), the damage percentage obtained for the healthy state is 5.5% which can be neglected. As expected, the highest damage values belong to patterns (2) and (1), respectively. The 3rd pattern has a lower damage value compared to patterns (4) and (5) and finally, the 6th pattern has the lowest damage percentage as it should be. It can be seen from the column labeled “case (e)” in Table 2 that the obtained result for every pattern is in proportion to the inflicted damage and a higher level of damage results in a higher damage percentage; this shows the reliability and efficiency of the proposed method in determining the severity of damage in the structure.

4.2. A Bridge Health Monitoring (BHM) Benchmark Model. The BHM benchmark model discussed in this paper is a grid structure located at the University of Central Florida that was developed to evaluate the reliability of SHM techniques. Figure 7 shows a scheme of this structure. It has two 5.49 m (18 ft) girders in longitudinal direction and seven 1.83 m (6 ft) transverse beams at 0.91 m (3 ft) spacing to provide lateral stability. These members all have the same cross section of $S3 \times 5.7$. The bridge has also six 1.07 m (42 in) columns with a $W12 \times 26$ cross section which are all fixed to the ground. A numerical model was also developed based on this structure for practitioners to test their health monitoring methodologies using static or dynamic tests on this model [38]. Here, we use acceleration data from this numerical model to evaluate the proposed damage detection methodology. Six damage patterns were studied for this purpose and six accelerometers were considered to record structural response under dynamic excitation. Table 3 describes these damage patterns. Also, the damaged nodes together with considered sensor placement are shown in Figure 8.

TABLE 1: Description of the damage patterns considered for IASC-ASCE benchmark structure [37].

Damage patterns	Description
1	The braces in the first story have all lost their stiffness.
2	The braces in the first and third stories have all lost their stiffness.
3	The north brace on the west face of the building, located in the first story, has lost its stiffness.
4	The north brace on the west face located in the first story and the west brace on the north face located in the third story have lost their stiffness.
5	In addition to damage in pattern (4), the beam-column connection noted in Figure 6 is partially unscrewed and thus can only transfer forces and does not sustain any bending moments.
6	The north brace on the west face located in the first story has lost one-third of its stiffness.

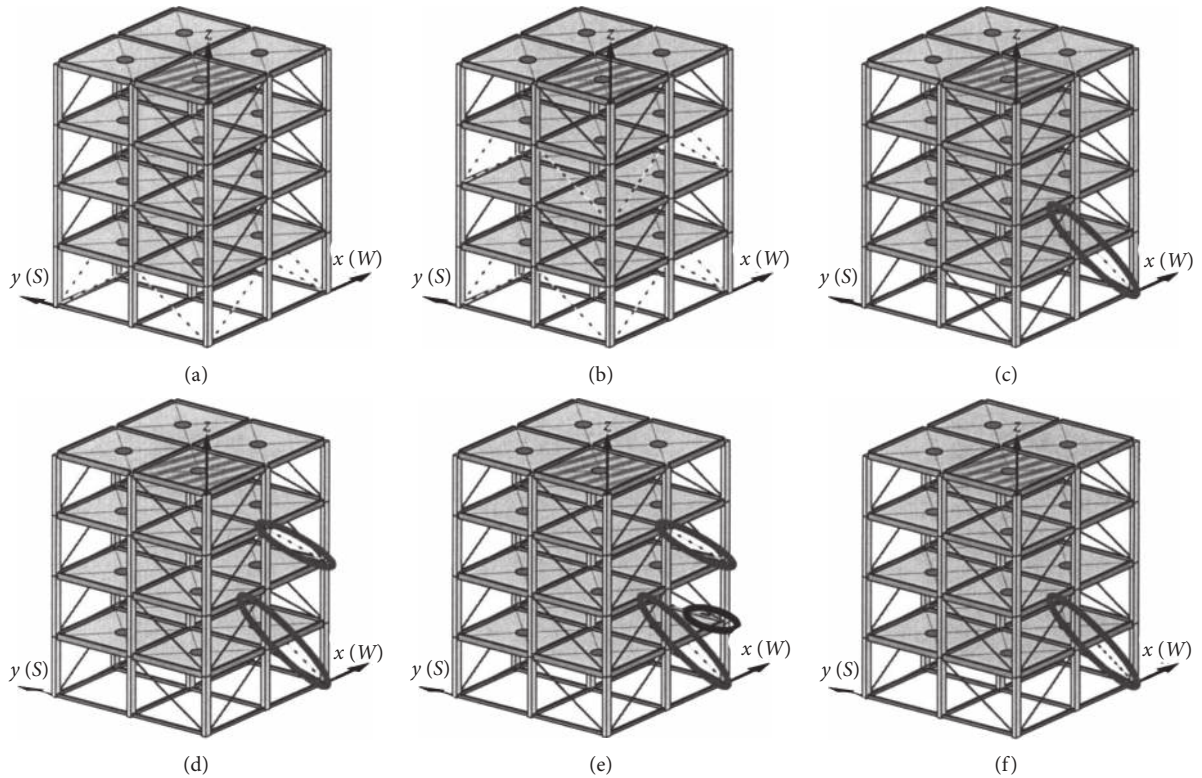


FIGURE 6: The six damage patterns considered for the IASC-ASCE benchmark structure [37]: (a) no stiffness in 1st floor braces, (b) no stiffness in 1st floor braces and no stiffness in 3rd floor braces, (c) no stiffness in one 1st floor brace, (d) no stiffness in one 1st floor brace and no stiffness in one 3rd floor brace, (e) no stiffness in one 1st floor brace and no stiffness in one 3rd floor brace beam-column connection weakened, and (f) 2/3 stiffness in one 1st floor brace.

TABLE 2: Damage percentage in IASC-ASCE benchmark structure under different damage patterns for cases (a) to (f).

Damage pattern	Damage percentage (%)					
	Case (a)	Case (b)	Case (c)	Case (d)	Case (e)	Case (f)
Healthy	5	5	4.5	5	5.5	5.52
1	84	76	79.5	76.6	78.25	78.2
2	90.5	79	86.5	86.2	87.37	86.24
3	30	29	29	28.6	28.42	27.4
4	38.5	28	27	28.54	30.67	29.36
5	27.5	33	33	31.74	34.05	35.16
6	17	14	13	16	17.55	17.16

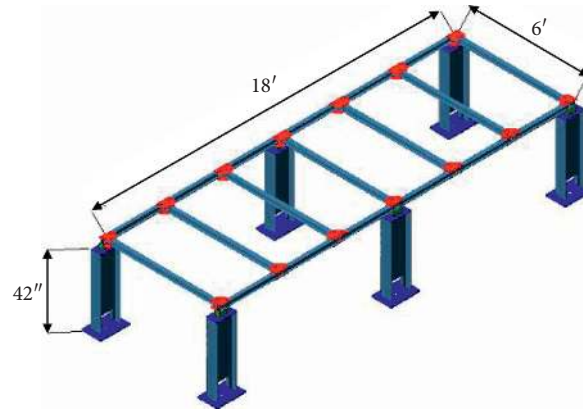


FIGURE 7: Scheme of the BHM benchmark structure [38].

TABLE 3: Damage patterns considered for the BHM benchmark structure [38].

Damage patterns	Description
1	The connection of the transverse member into node 7 does not sustain the major moment any longer.
2	In addition to pattern (1), the connection plates at node 7 are removed.
3	The beam-column connection in node 19 has lost its moment releases and has turned into a fixed connection.
4	Similar to damage pattern (3) but for nodes 19 and 38.
5	Reduction of 10% on the spring constant in the springs located at the supports of the bridge.
6	Reduction of 20% on the spring constant in the springs located at the supports of the bridge.

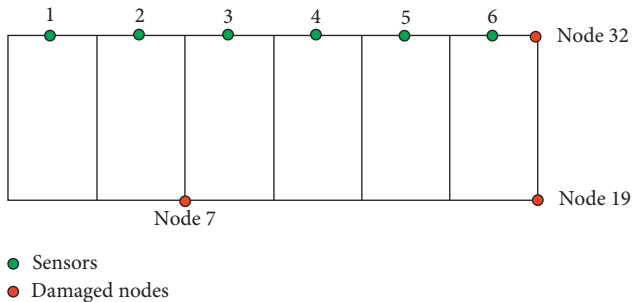


FIGURE 8: The sensor placement and the damaged nodes related to the BHM benchmark structure.

4.2.1. *Preprocessing.* To record acceleration data, a 10 KN dynamic load is applied to the model and its response is recorded by the six sensors at a sampling frequency of 500 Hz. The data from these sensors are concatenated and then divided into 128×6 matrices after normalization. The healthy state matrices are then shuffled before being used in the training phase.

4.2.2. *Training of the CAE.* The CAE developed for this structure has two 1D convolution layers in the encoder part with filter counts of 4 and 1, filter size of 4, and the same padding. Each of these two layers is followed by a 1D max pooling layer with a window length of 4. The decoder has three 1D convolution layers with filter counts of 1, 4, and 6, filter size of 4, and the same padding. An upsampling layer with a window length of 4 is placed after each one of the first two convolution layers. The number of matrices from the healthy state data needed to train this network is determined

by trial and error. 2000 128×6 matrices are used for this purpose.

4.2.3. *Data Compression and Damage Detection.* 1000 matrices from the healthy state data are fed to the encoder to form reference vectors of length 8. As with the case of the IASC-ASCE benchmark structure, a parametric study is performed at this point to determine the minimum number of input matrices from an unknown state needed for proper damage detection. Based on the results, at least 700 matrices from each state must be employed for this purpose and the damage detection process is described here accordingly. 700 matrices from the healthy state data that are not used in the training phase or for extracting reference vectors and 700 matrices from each damaged state data are inputted to the encoder separately and outputted as test vectors of length 8. The distances from each test vector to all the reference vectors are computed and the minimum of the obtained values is chosen as the final distance. The mean plus 1.4 standard deviations of the set of distances obtained for healthy state test vectors is chosen as the damage threshold. The damage percentage in the structure is then calculated using equation (5).

Table 4 shows the damage percentage in the bridge for the healthy and the six damaged states. As can be seen in the table, the amount of damage is 8.82% for the healthy state which is negligible. Releasing the major moment of a transverse member in one node increases the value to about 20%. When connection plates in this node are removed as well, the damage percentage will be increased to about 23%. In a similar way, when the reduction on the spring constant goes from 10% in damage pattern (5) to 20% in damage

TABLE 4: Damage percentage in the BHM benchmark structure under different damage patterns.

Damage pattern	Healthy	1	2	3	4	5	6
Damage percentage (%)	8.82	20.47	23.52	28.6	20.47	21.83	37.76



FIGURE 9: Tianjin Yonghe bridge [41].

pattern (6) the amount of damage increases from 21.83% to 37.76%. The stiffness of the structure under damage pattern (4), in which two columns are fixed to the deck, is more than that under damage pattern (3), in which only one column is fixed to the deck; thus, the amount of damage under pattern (4) is obtained less than that under pattern (3) (20.47% and 28.6%, respectively). Accordingly, even though the CAE is trained using only the healthy state data, the algorithm can successfully assign a reasonable damage percentage in all cases and performs well in damage quantification in the structure.

4.3. Tianjin Yonghe Bridge. The Tianjin Yonghe Bridge is one of the oldest cable-stayed bridges constructed in mainland China (Figure 9). It is composed of a 260 m long main span and two side spans of length 25.15 + 99.85 m and width 11 m. It also has two 60.5 m tall towers. After 19 years of operation, cracks with a maximum size of 2 cm were observed in the midspan girder, so the bridge was repaired and rehabilitated in 2007 [39, 40].

During rehabilitation, an SHM system was designed and installed on the bridge by the Center of Structural Monitoring and Control (SMC) at the Harbin Institute of Technology. In order to record acceleration time series, 14 uniaxial accelerometers were installed on the deck and one biaxial accelerometer was installed on top of the south tower. Figure 10 shows the placement of these sensors on the bridge [40].

In August 2008, the bridge was inspected and two damage patterns were observed. The side spans were seriously cracked and the piers were damaged causing the structure to lose a part of its vertical support. During this period, the installed SHM system recorded acceleration data from the healthy to the very damaged state [42].

4.3.1. Preprocessing. In this study, the data from the sensors embedded downstream of the deck (7 sensors), recorded on January 17, February 3, March 19, May 5, May 18, June 7, and June 16, are used to validate the proposed SHM method. The data recorded on January 17 2008 are considered as healthy state data; afterward, the amount of damage gradually increases. For each day, the sensors have recorded the data for 24 hours with a sampling frequency of 100 Hz;

therefore, a total of 864×10^4 samples are available for each sensor on each day. Figure 11 shows an example of the recorded data on January 17 and June 16. 360×10^4 of the samples are used to form a 3600000×7 matrix which is then normalized and divided into 128×7 matrices. These matrices are then shuffled before being fed to the CAE.

4.3.2. Training of the CAE. 1000 128×7 matrices from the data recorded on January 17 (the healthy state data) are used to train the CAE (the number of matrices needed at this stage is obtained by trial and error). The encoder part of the CAE used for this structure has three 1D convolution layers with filter counts of 64, 64, and 1, filter length of 16, and the same padding. A 1D max pooling layer is placed after each one of these convolution layers. The window length of these max pooling layers is 4, 2, and 2, respectively. The decoder has four 1D convolution layers with filter counts of 1, 64, 64, and 7, filter length of 16, and the same padding. The first three convolution layers are followed by a 1D upsampling layer. The window length in these upsampling layers is equal to 2, 2, and 4, respectively.

4.3.3. Data Compression and Damage Detection. 1000 128×7 matrices from the healthy state data are fed to the encoder part of the CAE trained in the second step to obtain 1000 vectors of length 8 (reference vectors). In addition, 600 other healthy matrices that are not used in the training phase and 600 matrices from the data recorded on the rest of the days are inputted to the encoder part to extract test vectors (the minimum number of matrices from each state needed to extract test vectors is identified in a parametric study, like in the case of IASC-ASCE benchmark structure). The minimum of the distances from each test vector to the reference vectors is considered as the final distance between that test vector and reference vectors. The mean plus 1.6 standard deviations of the set of distances obtained for healthy state test vectors is chosen as the damage threshold. The amount of damage on each of the considered days is then computed using equation (5). Table 5 shows the results of this computation. As can be seen, the amount of damage on January 17 and February 3 is 5.9% and 6.95%, respectively, which is a low and negligible amount. However, the damage percentage gradually increases until it reaches

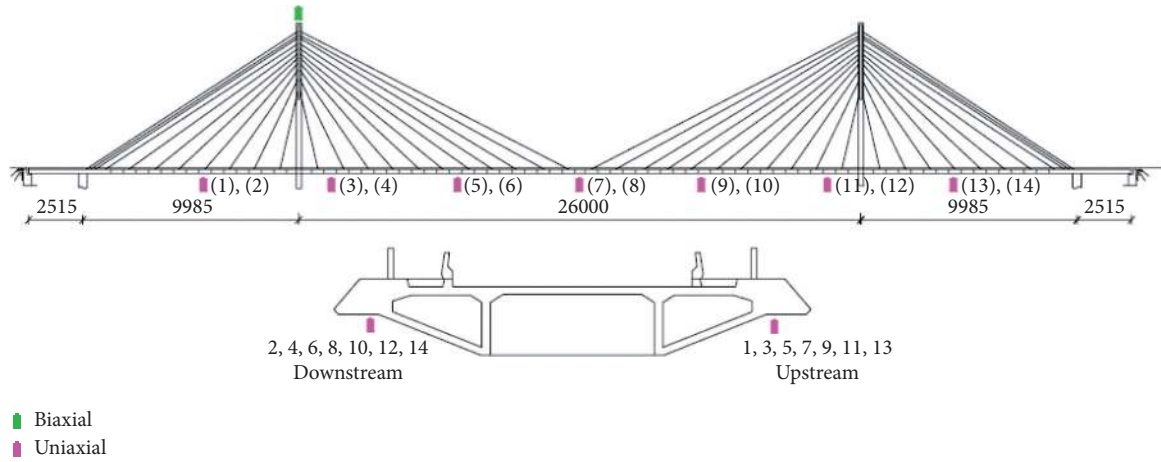


FIGURE 10: Placement of the accelerometers installed on Tianjin Yonghe Bridge as part of the SHM system [43].

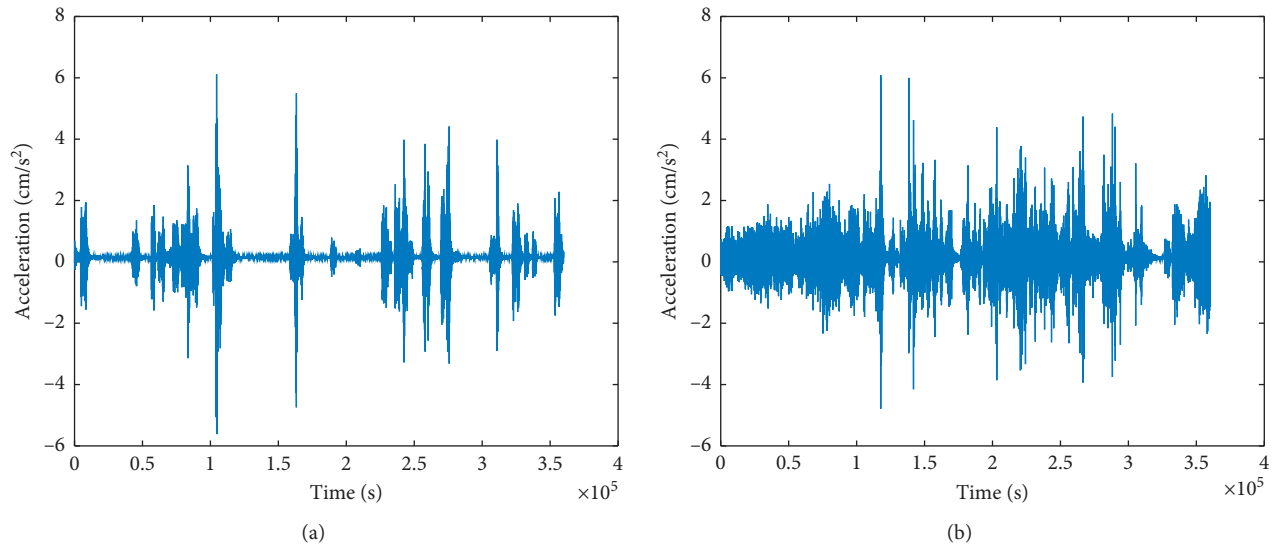


FIGURE 11: Acceleration data recorded by sensor number 2 on (a) January 17 and (b) June 16, 2008.

TABLE 5: Damage percentage in Tianjin Yonghe Bridge on the seven considered days.

Date	Jan. 17	Feb. 3	Mar. 19	May. 5	May. 18	Jun. 7	Jun. 16
Damage percentage (%)	5.9	6.95	16.12	29.24	36.56	40.3	46.34

46.34% on June 16. According to Table 5, the obtained results are consistent with the expected overall condition of the bridge knowing that the damage is propagating through the structure which proves the success of the proposed algorithm in damage quantification in this bridge as a real-life structure.

5. Conclusions

In this paper, a new unsupervised deep-learning-based method was proposed to detect structural damage. The objective was to provide a practical way for global health monitoring of civil structures using raw acceleration data. CAEs with simple structures were used for this purpose. The proposed method was validated through applications on

numerical models of IASC-ASCE benchmark structure and a BHM benchmark model and also on the full-scale Tianjin Yonghe Bridge.

The CAEs were trained using only healthy state data; no information from the damaged states is needed during the training phase. After training, the encoder part is used for data compression and damage detection is done by simply comparing the compressed data from a damaged state with the compressed data from the healthy state. Applications on the three above-mentioned structures show that the proposed method can successfully detect and quantify damage in all damage scenarios considered for the structures. The CAEs used here have a simple structure and provide an optimal way for realizing the severity of damage in civil structures, which proves that using complex machine

learning-based algorithms is not necessary for the purposes of this paper.

Data Availability

The data supporting this study are from previously reported studies and datasets, which have been cited.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Sun, L. Liang, M. Li, and X. Li, "Bridge performance evaluation via dynamic fingerprints and data fusion," *Journal of Performance of Constructed Facilities*, vol. 33, no. 2, p. 04019004, 2019.
- [2] M. Abè, "Structural monitoring of civil structures using vibration measurement Current practice and future," in *Artificial Intelligence in Structural Engineering*, pp. 1–18, Springer, Berlin, Germany, 1998.
- [3] J. P. Amezquita-Sanchez and H. Adeli, "Signal processing techniques for vibration-based health monitoring of smart structures," *Archives of Computational Methods in Engineering*, vol. 23, no. 1, pp. 1–15, 2016.
- [4] H. Adeli and S. L. Hung, *Machine Learning-Neural Networks, Genetic Algorithms and Fuzzy Systems*, John Wiley & Sons, New York, NY, USA, 1995.
- [5] C.-B. Yun, J.-H. Yi, and E. Y. Bahng, "Joint damage assessment of framed structures using a neural networks technique," *Engineering Structures*, vol. 23, no. 5, pp. 425–435, 2001.
- [6] J. J. Lee, J. W. Lee, J. H. Yi, C. B. Yun, and H. Y. Jung, "Neural networks-based damage detection for bridges considering errors in baseline finite element models," *Journal of Sound and Vibration*, vol. 280, no. 3–5, pp. 555–578, 2005.
- [7] J. Li, U. Dackermann, Y.-L. Xu, and B. Samali, "Damage identification in civil engineering structures utilizing PCA-compressed residual frequency response functions and neural network ensembles," *Structural Control and Health Monitoring*, vol. 18, no. 2, pp. 207–226, 2011.
- [8] R. A. Osornio-Rios, J. P. Amezquita-Sanchez, R. J. Romero-Troncoso, and A. Garcia-Perez, "MUSIC-ANN analysis for locating structural damages in a truss-type structure by means of vibrations," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 9, pp. 687–698, 2012.
- [9] R. P. Bandara, T. H. T. Chan, and D. P. Thambiratnam, "Frequency response function based damage identification using principal component analysis and pattern recognition technique," *Engineering Structures*, vol. 66, pp. 116–128, 2014.
- [10] O. Abdeljaber and O. Avci, "Nonparametric structural damage detection algorithm for ambient vibration response: utilizing artificial neural networks and self-organizing maps," *Journal of Architectural Engineering*, vol. 22, no. 2, p. 04016004, 2016.
- [11] H. Pan, M. Azimi, G. Gui, F. Yan, and Z. Lin, "Vibration-based support vector machine for structural health monitoring," in *Proceedings of the International Conference on Experimental Vibration Analysis for Civil Engineering Structures*, San Diego, CA, USA, July 2017.
- [12] G. Gui, H. Pan, Z. Lin, Y. Li, and Z. Yuan, "Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection," *KSCSE Journal of Civil Engineering*, vol. 21, no. 2, pp. 523–534, 2017.
- [13] R. Ghiasi, P. Torkzadeh, and M. Noori, "A machine-learning approach for structural damage detection using least square support vector machine based on a new combinational kernel function," *Structural Health Monitoring: An International Journal*, vol. 15, no. 3, pp. 302–316, 2016.
- [14] C. M. Wen, S. L. Hung, C. S. Huang, and J. C. Jan, "Unsupervised fuzzy neural networks for damage detection of structures," *Structural Control and Health Monitoring*, vol. 14, no. 1, pp. 144–161, 2007.
- [15] S.-F. Jiang, C.-M. Zhang, and S. Zhang, "Two-stage structural damage detection using fuzzy neural networks and data fusion techniques," *Expert Systems with Applications*, vol. 38, no. 1, pp. 511–519, 2011.
- [16] K. Boonlong, "Vibration-based damage detection in beams by cooperative coevolutionary genetic algorithm," *Advances in Mechanical Engineering*, vol. 6, p. 624949, 2014.
- [17] A. Greco, A. Pluchino, F. Cannizzaro, S. Caddemi, and I. Calio, "Closed-form solution based genetic algorithm software: application to multiple cracks detection on beam structures by static tests," *Applied Soft Computing*, vol. 64, pp. 35–48, 2018.
- [18] A. Greco, D. D'Urso, F. Cannizzaro, and A. Pluchino, "Damage identification on spatial Timoshenko arches by means of genetic algorithms," *Mechanical Systems and Signal Processing*, vol. 105, pp. 51–67, 2018.
- [19] A. Yu, A. Ji, and S. Sun, "An improved multi-objective genetic algorithm and data fusion in structural damage identification," *International Journal of Security and Networks*, vol. 14, no. 2, pp. 95–102, 2019.
- [20] P. Ghannadi and S. S. Kourehli, "Multiverse optimizer for structural damage detection: numerical study and experimental validation," *The Structural Design of Tall and Special Buildings*, vol. 29, no. 13, p. e1777, 2020.
- [21] P. Ghannadi, S. S. Kourehli, M. Noori, and W. A. Altabay, "Efficiency of grey wolf optimization algorithm for damage detection of skeletal structures via expanded mode shapes," *Advances in Structural Engineering*, vol. 23, no. 13, pp. 2850–2865, 2020.
- [22] S. Pouyanfar, S. Sadiq, Y. Yan et al., "A survey on deep learning: algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–36, 2018.
- [23] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *Journal of Sound and Vibration*, vol. 388, pp. 154–170, 2016.
- [24] Y.-Z. Lin, Z.-H. Nie, and H.-W. Ma, "Structural damage detection with automatic feature-extraction through deep learning," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 12, pp. 1025–1046, 2017.
- [25] N. S. Gulgec, M. Takáč, and S. N. Pakzad, "Convolutional neural network approach for robust structural damage detection and localization," *Journal of Computing in Civil Engineering*, vol. 33, no. 3, p. 04019005, 2019.
- [26] P. Ma, H. Zhang, W. Fan, C. Wang, G. Wen, and X. Zhang, "A novel bearing fault diagnosis method based on 2D image representation and transfer learning-convolutional neural network," *Measurement Science and Technology*, vol. 30, no. 5, p. 055402, 2019.
- [27] S. Cofre-Martel, P. Kobrich, E. L. Droguett, and V. Meruane, "Deep convolutional neural network-based structural damage localization and quantification using transmissibility data,"

- Shock and Vibration*, vol. 2019, Article ID 9859281, 27 pages, 2019.
- [28] G. Li, Q. Liu, S. Zhao, W. Qiao, and X. Ren, "Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive Bayes data fusion based on a visual detection system," *Measurement Science and Technology*, vol. 31, no. 7, p. 075403, 2020.
- [29] J. Guo, X. Liu, S. Li, and Z. Wang, "Bearing intelligent fault diagnosis based on wavelet transform and convolutional neural network," *Shock and Vibration*, vol. 2020, Article ID 6380486, 14 pages, 2020.
- [30] M. H. Rafiei and H. Adeli, "A novel unsupervised deep learning model for global and local health condition assessment of structures," *Engineering Structures*, vol. 156, pp. 598–607, 2018.
- [31] M. Silva, A. Santos, R. Santos, E. Figueiredo, C. Sales, and J. C. W. A. Costa, "Deep principal component analysis: an enhanced approach for structural damage identification," *Structural Health Monitoring*, vol. 18, no. 5-6, pp. 1444–1463, 2019.
- [32] T.-Y. Hsu, A. Valentino, A. Liseikin et al., "Continuous structural health monitoring of the Sayano-Shushenskaya Dam using off-site seismic station data accounting for environmental effects," *Measurement Science and Technology*, vol. 31, no. 1, p. 015801, 2019.
- [33] Z. Wang and Y.-J. Cha, "Unsupervised deep learning approach using a deep auto-encoder with an one-class support vector machine to detect structural damage," *Structural Health Monitoring*, vol. 20, no. 1, pp. 406–425, 2020.
- [34] X. Ma, Y. Lin, Z. Nie, and H. Ma, "Structural damage identification based on unsupervised feature-extraction via variational auto-encoder," *Measurement*, vol. 160, p. 107811, 2020.
- [35] D. Charte, F. Charte, S. García, M. J. del Jesus, and F. Herrera, "A practical tutorial on autoencoders for nonlinear feature fusion: taxonomy, models, software and guidelines," *Information Fusion*, vol. 44, pp. 78–96, 2018.
- [36] L. Chen, Q. Guan, B. Feng, H. Yue, J. Wang, and F. Zhang, "A multi-convolutional autoencoder approach to multivariate geochemical anomaly recognition," *Minerals*, vol. 9, no. 5, p. 270, 2019.
- [37] E. A. Johnson, H. F. Lam, L. S. Katafygiotis, and J. L. Beck, "Phase I IASC-ASCE structural health monitoring benchmark problem using simulated data," *Journal of Engineering Mechanics*, vol. 130, no. 1, pp. 3–15, 2004.
- [38] J. L. Burkett, "Benchmark studies for structural health monitoring using analytical and experimental models," Masters Thesis, University of Central Florida, Orlando, FL, USA, 2005.
- [39] H. Li, S. Li, J. Ou, and H. Li, "Modal identification of bridges under varying environmental conditions: temperature and wind effects," *Structural Control and Health Monitoring*, vol. 17, no. 5, pp. 495–512, 2010.
- [40] S. Li, H. Li, Y. Liu, C. Lan, W. Zhou, and J. Ou, "SMC structural health monitoring benchmark problem using monitored data from an actual cable-stayed bridge," *Structural Control and Health Monitoring*, vol. 21, no. 2, pp. 156–172, 2014.
- [41] M. Kaloop, J. Hu, and E. Elbeltagi, "Adjustment and assessment of the measurements of low and high sampling frequencies of GPS real-time monitoring of structural movement," *ISPRS International Journal of Geo-Information*, vol. 5, no. 12, p. 222, 2016.
- [42] H. Babajanian Bisheh, G. Ghodrati Amiri, M. Nekooei, and E. Darvishan, "Damage detection of a cable-stayed bridge using feature extraction and selection methods," *Structure and Infrastructure Engineering*, vol. 15, no. 9, pp. 1165–1177, 2019.
- [43] M. R. Kaloop and J. W. Hu, "Stayed-cable bridge damage detection and localization based on accelerometer health monitoring measurements," *Shock and Vibration*, vol. 2015, Article ID 102680, 11 pages, 2015.