DOI: 10.1111/mice.12517

ORIGINAL ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Structural health monitoring using extremely compressed data through deep learning

Mohsen Azimi | **Gokhan Pekcan**

Department of Civil and Environmental Engineering, University of Nevada Reno, Reno, NV, USA

Correspondence
Gokhan Pekcan, Department of Civil and Environmental Engineering, University of Nevada, Reno, Mail Stop 258, Reno, NV 89557, USA.
Email: pekcan@unr.edu

Funding information
National Science Foundation, Grant/Award Number: 1646420

Abstract

This study introduces a novel convolutional neural network (CNN)-based approach for structural health monitoring (SHM) that exploits a form of measured compressed response data through transfer learning (TL)-based techniques. The implementation of the proposed methodology allows damage identification and localization within a realistic large-scale system. To validate the proposed method, first, a well-known benchmark model is numerically simulated. Using acceleration response histories, as well as compressed response data in terms of discrete histograms, CNN models are trained, and the robustness of the CNN architectures is evaluated. Finally, pretrained CNNs are fine-tuned to be adaptable for three-parameter, extremely compressed response data, based on the response mean, standard deviation, and a scale factor. The performance of each CNN implementation is assessed using training accuracy histories as well as confusion matrices, along with other performance metrics. In addition to the numerical study, the performance of the proposed method is demonstrated using experimental vibration response data for verification and validation. The results indicate that deep TL can be implemented effectively for SHM of similar structural systems with different types of sensors.

1 | INTRODUCTION

It is essential to monitor critical civil infrastructure during their life cycle. For many decades, operational methods for assessing the health and status of structural systems have primarily been through visual inspections. While this approach remains essential for system health assessment, it presents significant limitations that hinder the detection of various types and extent of damage after short- or long-term disruptive events. On the other hand, with the ongoing advancements in low-cost sensor technologies and high-performance computing, continuous and effective structural health monitoring (SHM) has become more feasible. In particular, extensive research has been conducted on vibration-based damage identification, and significant progress has been made in this area. A broad range of techniques, algorithms, and methods are developed to solve various problems encountered in different structures with varying complexities. Comprehensive state-of-the-art reviews of various aspects of SHM are available in a number of publications (Carden & Fanning, 2004; Doebling, Farrar, Prime, & Shevitz, 1996; W. Fan & Qiao, 2011; Kong, Cai, & Hu, 2017; Sinou, 2009; Sohn et al., 2003). However, to this date, two of the major impediments to the widespread application of SHM for civil infrastructure systems have been the lack of (1) efficient and reliable methodologies coupled with signal processing of large amounts of heterogeneous response data and (2) inexpensive sensors. With the recent technological advancements, a relatively large number of sensors and sensor networks can now be deployed that produce large volumes of response data. While the acquired data may not always provide sufficient or suitable information for conventional SHM methods, data-driven machine-learning (ML) techniques have been proposed by researchers to assess the global health condition of host structures (Bao, Chen et al., 2019; Khoa, Alamdari, Rakotoarivelo, Anaissi, & Wang, 2018; Sen & Nagarajaiah, 2018).

Data-driven damage detection methods may be implemented as in pattern recognition problems for which neural networks (NNs) can be used due to their adaptive learning and fault tolerance capabilities. One of the challenges for conventional NNs is that they require significant training datasets and computation costs. Such limitations were recently addressed by substituting deep learning (DL) tools for feature extraction and classification tasks in damage detection problems using raw and processed signals without any hand-designed features (Bao, Tang, Li, & Zhang, 2019; Tang, Chen, Bao, & Li, 2019). At the center of the state-of-the-art in DL with big-data lie convolutional neural networks (CNNs) that have the ability to learn from any type of large size data from multiple observables (sensors). CNNs have been used for classification purposes such as electrocardiogram signals (Kang & Cha, 2018; Kiranyaz, Ince, & Gabbouj, 2016) and images (Scherer, Müller, & Behnke, 2010); however, they are still new in SHM (Rafiei & Adeli, 2017; Rafiei, Khushefati, Demirboga, & Adeli, 2017). Other successful applications of CNNs in SHM are available in the literature, which includes damage detection of steel frames (Abdeljaber, Avci, Kiranyaz, Gabbouj, & Inman, 2017), pavement and concrete crack detection (Cha, Choi, Suh, Mahmoudkhani, & Büyüköztürk, 2018; Zhang et al., 2019), and overall system condition assessment (Khodabandehlou, Pekcan, & Fadali, 2019). The scale of attention toward the use of DL methods in damage detection, particularly CNN-based models with the ability to learn hierarchical features, proves the great potential of such strategies in tackling SHM problems.

The response data used for SHM purposes is almost exclusively recorded in the time domain; however, some of the researchers proposed the use of transformed data from the time domain to frequency or time-frequency domains to identify damaged structures from intact ones (Pan, Azimi, Gui, Yan, & Lin, 2017; Pan, Azimi, Yan, & Lin, 2018). Kaveh and Dadras (2018) improved a nature-inspired algorithm and applied it to vibration-based structural damage localization task, in order to minimize the differences between the vibration characteristics of FE model and real structure. A similar approach was also taken by Oh, Kim, Kim, Park, and Adeli (2017) to develop a sustainable strain sensing model for high-rise buildings. Tsogka, Daskalakis, Comanducci, and Ubertini (2017) developed a computationally efficient technique based on the stretching method for long-term monitoring of large structures using vibration response data. Recently, Roohi, Hernandez, and Rosowsky (2019) proposed a nonlinear model-data fusion algorithm for state estimation in nonlinear hysteretic structural systems. The main feature of the proposed algorithm is that it is designed to be physically realizable as a nonlinear structural model, which makes it appealing for vibration-based monitoring applications. The approach was employed for seismic monitoring of experimental and real-world large-scale instrumented buildings (Hernandez, Roohi, & Rosowsky, 2018; Roohi et al., 2019). Yao, Yi, Qu, and Li (2018) proposed a new blind identification method based on sparse component analysis through time-frequency method, which was experimentally evaluated using measured acceleration data from the sensors installed on the Yonghe Bridge. Pan, Azimi, Gui et al. (2017) and Pan, Azimi, Yan et al. (2018) proposed ML-based methods for vibration-based damage detection for large-scale bridge structures utilizing time-frequency approaches. Li, Park, and Adeli (2017), Amezquita-Sanchez and Adeli (2015), and Amezquita-Sanchez, Park, and Adeli (2017) took the advantages of wavelet and Hilbert transforms for modal parameter identification. Yuen and Huang (2018) improved the Bayesian substructure identification method for inverse problems in SHM of large-scale structures. Huang, Beck, and Li (2018) introduced a multitask sparse Bayesian learning approach by using multiple groups of measurements based on the sparseness similarity, which can be used as a tool to reconstruct the lost data during wireless data transmission, as well as in structural stiffness estimation. These methods are developed for feature extraction to identify the presence, location, and type of damage in structural systems. A detailed breakdown of the various methods including those derived from statistical and probabilistic approaches is presented in previous studies (W. Fan & Qiao, 2011; Kong et al., 2017).

Recently, a type of self-powered sensor which acquires strain or acceleration response data in a compressed form has been introduced (Alavi, Hasni, Jiao, Borchani, & Lajnef, 2017; Aono et al., 2017; Hasni et al., 2017). In these sensors, a series of memory cells store the duration of measured responses cumulatively provided that the preselected thresholds are defined and discretized. In other words, the duration of each event corresponds to the number of exceedance when the input signal exceeds the predefined thresholds. Therefore, the recorded response data is in the form of histogram of events (compressed), instead of time-history of responses. To use this form of discrete, compressed, and limited response data for damage detection purposes, new approaches must be defined. It is further suggested that statistical properties of fitted continuous distributions such as mean and standard deviation (STD) provide extremely compressed representation of the response data. These quantities can be used to identify the type, location, and extent of damage using DL, which is the main goal of this study.

The present study can be viewed in three phases. First, condition of a SHM benchmark problem (Caicedo, Dyke, & Johnson, 2004; Johnson, Lam, Katafygiotis, & Beck, 2004) was evaluated using a CNN model based on measured acceleration response data acquired by conventional

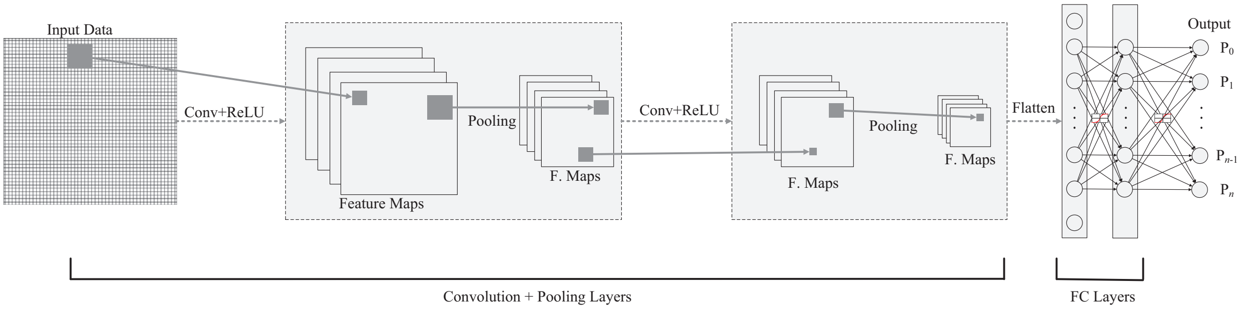


FIGURE 1 A typical convolutional neural network

Micromachined Microelectromechanical Systems (MEMS) sensors. This model and its performance were used as reference for the evaluation of the methods proposed in this paper. In the second phase, all of the acceleration time-history data were converted to discrete histograms (compressed data) that would have been recorded by self-powered sensors, and system condition assessment was then carried out using a CNN model as in the first phase. In the third phase, a transfer learning (TL)-based method was developed to extract information from discrete response histograms (compressed data), and to utilize three-parameter representation of vibration response data (extremely compressed) for subsequent system evaluations. For this purpose, 70% of the discrete histograms were treated as the source domain for which the "pretrained" CNN models were developed. Subsequently, each model is "fine-tuned" using the fitted smooth distributions (target domain) obtained from the remaining 30% of the histograms, which are essentially defined by only three parameters; mean, variance, and a scale factor. Finally, it is noted that system evaluation in each phase was performed using 90% of available sensor data for training and the remaining 10% for testing of CNN. The strategy was evaluated and verified using a SHM benchmark model with different damage patterns. The numerical simulations of structural models were carried out using Matlab®, while the CNN models were created and trained using Python libraries that provide high-performance DL algorithms, particularly when using Tensorflow-GPU, which makes training process significantly faster through GPU-based parallel computation. Subsequently, the proposed method and CNN models are used with experimentally recorded response data from two different SHM benchmark structures to further validate the accuracy and robustness.

2 | CNN

CNNs belong to a group of artificial neural networks (ANNs), which are well known for their applications in the classification of images, audio, and text. The main difference between a typical neural network and a CNN is that NN uses general multiplication while CNN uses convolution that makes it a perfect tool for processing data in a vector or grid form such as pictures/images. It was demonstrated that in applications for SHM, time-history of vibration response data could be converted into grid-like 2D matrices (Khodabandehlou et al., 2019). Furthermore, time-history of vibration response data obtained from sensors could be treated as distinct images. Clearly, various forms of response data representations (time-history, discrete histograms or continuous distributions) can be treated as such. Figure 1 shows the architecture of a CNN for which the inputs are images of specific sizes. Different configurations can be considered for a CNN depending on the performance and the cost of computation. Different filters are trained and applied to the input images in order to automatically extract features. The fully connected (FC) layers link the output of convolution layers to the output layer with specified activation functions.

The most important segment of a CNN is the convolution and pooling layers. For a 1D convolution with a single channel, the operating function for discrete data can be written as (Avci, Abdeljaber, Kiranyaz, & Inman, 2017):

$$F(i) = \sum_{n=1}^{\upsilon_k} S(i+n)K(n) \tag{1}$$

For a multichannel scenario, convolution is applied on each channel in parallel, and the results are added up. The "stride" parameter is the step length between each gap, which is equal to one for all the convolution layers in this study. Each convolution layer is followed by a pooling layer. Using the pooling layers, the large feature map is down-sampled in a way that the small variance of the input data is not reflected on the outputs. In this study, the max-pooling layers reduce the dimensions by half. Activation functions are used in order to activate nonlinear mapping for an NN. The rectified linear units (ReLUs) (Glorot, Bordes, & Bengio, 2011; Nair & Hinton, 2010) are half-rectified activations functions for the convolutions layers that can increase the training speed. Therefore, the gradients on the right-side of the function can be retained with cheaper computations without vanishing. The main issue with the ReLU is that some weights on the left-side can vanish during the training and never activate again. Therefore, in this study, the leaky ReLU function (Maas, Hannun, & Ng, 2013) is used after each batch normalization (BN), which increases the range of the ReLU function on the left-side by introducing a small slope to the function; $\alpha = 0.01$ in this study.

2.1 | Regularization

In order to avoid overfitting and premature convergence, one method of generalization is to add dropout layers to the network architecture, which drop out a percentage of data at each training step. As the input data flow into a CNN, the values are adjusted based on weights, which makes data too big or too small (vanishing gradient problem). Therefore, the algorithm needs to fit such unstable distributions in each training iteration, which results in a low learning rate. This problem can be alleviated with a cheaper computational cost if each mini-batch is normalized based on mean or variance. Thus, the outputs have a similar distribution, which makes the training easier and with fast convergence. BN was introduced first by researchers at Google (Ioffe & Szegedy, 2015) to solve the internal covariate shift. Nowadays, BN is used in the majority of CNN configurations because: (a) the required iterations are reduced and the training of network is much faster, which enables training deeper networks; (b) higher learning rates are allowed, which is not always possible with gradient descent; (c) more activation functions become viable, since they are regulated by BN; (d) better overall results can be achieved. BN uses weights without adding bias term, and it is added before calling the activation functions (Cooijmans, Ballas, Laurent, Gülçehre, & Courville, 2016). Therefore, dropout can be removed from the network architecture since BN does have regularization effect, as well (Lin, Nie, & Ma, 2017). BN procedure is presented in Equations (2)–(5).

$$\mu_D = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2}$$

$$\sigma_D^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_D)^2 \tag{3}$$

$$\hat{x}_i = \frac{x_i - \mu_D}{\sqrt{(\sigma_D^2 + \epsilon)}} \tag{4}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{5}$$

where ϵ is a very small value, 10^{-6} in this study, to avoid the denominator becoming zero. Using BN, in each training step, the mean and variance of each mini-batch of data are calculated, and the original data is shifted and scaled to have zero mean and one variance. In order to keep the model flexible, γ and β parameters are introduced.

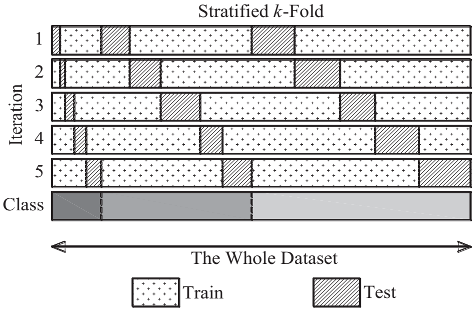


FIGURE 2 Visualization of stratified *k*-fold cross-validation with $k = 5$

2.2 | Training and cross-validation

The designed networks are trained using the Adam stochastic gradient descent optimization method (Kingma & Ba, 2014) and the categorical cross-entropy loss as the objective function that measures the performance of a classification method, for which the output is a probability value. For multiclass classification, it is defined as (Avci et al., 2017):

$$H(p,q) = -\sum_{i=1}^{M} p(x_i)\log(q(x_i)) \tag{6}$$

where, H is the cross-entropy loss; p and q are true and predicted the probability for an observation, respectively; M is the number of total classes. For every training iteration, the data batch size is 128, which shows the number of steps in a single epoch. The learning rate of Adam algorithm is selected to be $lr = 0.00005$, and the parameters for the weight of the momentum and decay of learning rate are selected as $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

In order to achieve more generalized relationships, the *k*-fold cross-validation is used in this study (Figure 2). Using the *k*-fold cross-validation, (a) the entire dataset is split into *k* folds; (b) for each fold, a model is trained based on the $k-1$ folds, and the effectiveness is validated for the *k*th fold; (c) by recording the prediction accuracy from each run, average *k* recorded accuracy serves as the performance. In order to ensure that each fold is a proper representative of the entire dataset, data is rearranged using stratification. Data samples are shuffled before splitting into specified batches.

The details of the reference CNN model that is used to compare the accuracy using MEMS and self-powered sensors, as well as training the "pretrained" networks, are given in Table 1, which are slightly modified for different cases depending on the database size and difficulty of the problem. Following similar studies in literature, the architecture of the reference CNN model was optimized starting from fewer convolution layers and by trial and error. The flowchart of this model is presented in Figure 3.

TABLE 1 The configuration of the reference CNN architecture

Layer	Type	Kernel number	Kernel size	Stride	Padding	Activation
1	Convolution	16	3	1	Same	ReLU
2*	Convolution	16	3	1	Same	ReLU
3**	Convolution	16	3	1	Same	ReLU
4	Batch normalization	–	–	–	–	–
5	Advanced activation	–	–	–	–	Leaky ReLU
6	Max. pooling	–	2	–	Valid	–
7	Convolution	32	3	1	Same	ReLU
8*	Convolution	32	3	1	Same	ReLU
9**	Convolution	32	3	1	Same	ReLU
10	Batch normalization	–	–	–	–	–
11	Advanced activation	–	–	–	–	Leaky ReLU
12	Max. pooling	–	2	–	Valid	–
13	Convolution	64	3	1	Same	ReLU
14*	Convolution	64	3	1	Same	ReLU
15**	Convolution	64	3	1	Same	ReLU
16	Batch normalization	–	–	–	–	–
17	Advanced activation	–	–	–	–	Leaky ReLU
18	Max. pooling	–	2	–	Valid	–
19	Flatten	–	–	–	–	–
20	Batch normalization	–	–	–	–	–
21	Dense	–	–	–	–	–
22	Dense	–	–	–	–	–
23	Softmax	–	–	–	–	Softmax

Note: * and ** are used for Case 3 and Case 8, respectively.

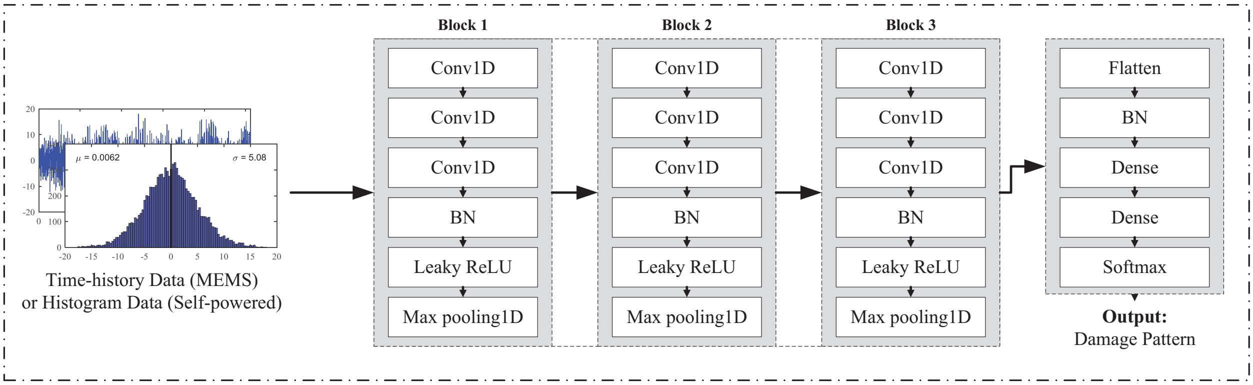


FIGURE 3 The reference 1D-CNN configurations (two separate models for two types of input data/sensors)

3 | TRANSFER LEARNING

For a deep CNN, as the depth of the networks increases, the training procedure takes much longer time due to the complexity of the network and a larger number of training parameters requires more data. TL provides a powerful tool to reduce the dependency of DL techniques on the size of available data. By definition, TL transfers some level of knowledge from a source domain to the target domain for similar data types (Pan & Yang, 2010). Application of TL is an emerging area in vision-based SHM, and novel studies are being carried out in classification problems in vision-based SHM (Gao & Mosalam, 2018); however, TL applications in vibration-based SHM are not significant.

Following the state-of-the-art in vision-based SHM, during the TL process, feature extraction layers of a pretrained CNN are frozen and the classification layers are fine-tuned with a lower computational cost. Figure 4 shows different TL strategies that can be used based on the data size in the target domain and its similarity to the source domain. In this figure,

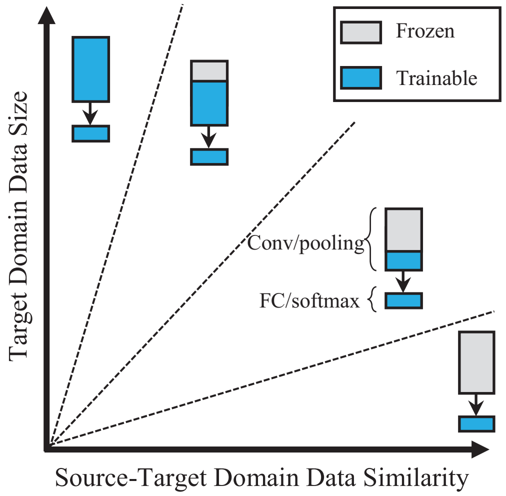


FIGURE 4 Different TL strategies based on target domain data size and its similarity to source domain data

which is simplified according to the reference CNN model, the convolution and pooling layers are shown with larger rectangles and the FC layers with smaller rectangles. For these four approaches, the trainable layers are shown with lightly shaded boxes. Another common way in TL is to replace the fully connected (FC) layers of a pretrained network with new layers based on the complexity of the classification problem; in this study, the FC layers are replaced with two successive BN-Dense-Leaky ReLU layers that are followed by the Softmax layer. Therefore, the weights in the convolutional layers are fixed and skipped during the training, and the output of the "Flatten" layer is considered as the training data for fine-tuning the classification layers' weights.

The general practice in vision-based SHM using TL is to fine-tune and improve the flexibility of a pretrained DL model that has already been created based on an image dataset. The same approach is considered in this study; however, pretraining and fine-tuning are done on the same structure response, but each with different data domain.

As it was mentioned earlier and shown in Figure 5, 70% of the discrete histogram data (compressed) was taken as the source domain to build the pretrained CNN models, and the remaining 30% of the data (extremely compressed; target domain) was fitted smooth distributions to derive three parameters; mean, variance, and a scale factor. By freezing the first layers of the pretrained network, the knowledge (CNN's weights) from the source domain is used for adapting the mid- to high-level classification tasks in the target domain using the three parameters of the smooth distributions. It should be noted that the tenfold cross-validation technique was used for all of the models. The architectures of the CNN models for the pretrained and fine-tuned models are provided in Figure 6. The convolutional blocks are similar for all of the models.

For each damage pattern in this study, 1,000 uniquely labeled data samples were numerically generated using different input excitation signals. Therefore, the total number of data samples is 9,000 that include the intact structure as well. Subsequently, the discrete histograms of the entire data were obtained to simulate the response data that would have been recorded using self-powered sensors. In addition, smooth distributions with the same number of bins (as in discrete histograms) were derived based on the mean, variance, and scale factor parameters. For each fold of the tenfold cross-validation, 10% of the data samples were selected for testing. Therefore, all of the 1,000 data samples appeared in the testing phase (e.g., confusion matrices). Only 70% of the histogram data were used as the source data domain to train and test the "pretrained" CNN models for TL, and the remaining 30%, that is, smooth distributions, were used as the target data domain to "fine-tune" the "pretrained" models in the previous step. The tenfold cross-validation method was used for training and testing of all of the models. No identical data samples were observed during the numerical data generation.

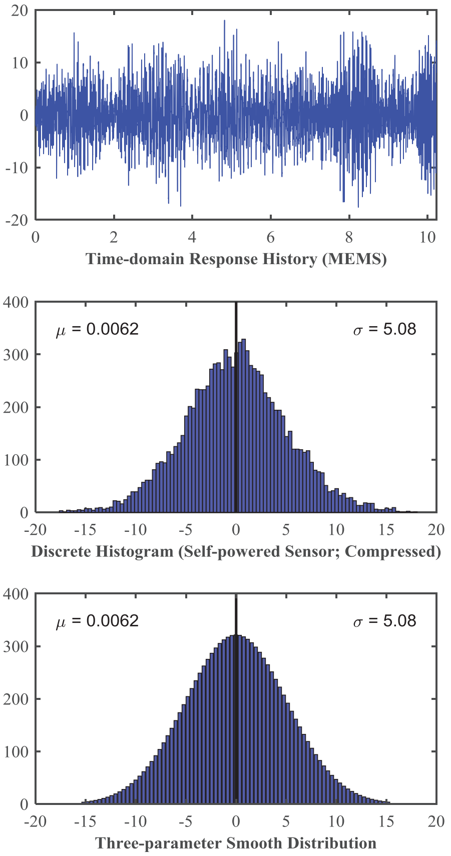


FIGURE 5 Data compression in the form of histograms (level crossing cumulative time counting) using self-powered sensors

4 | NUMERICAL VERIFICATION

Figure 7 shows the four-story quarter-scale steel frame model known as the IASC-ASCE structural health monitoring

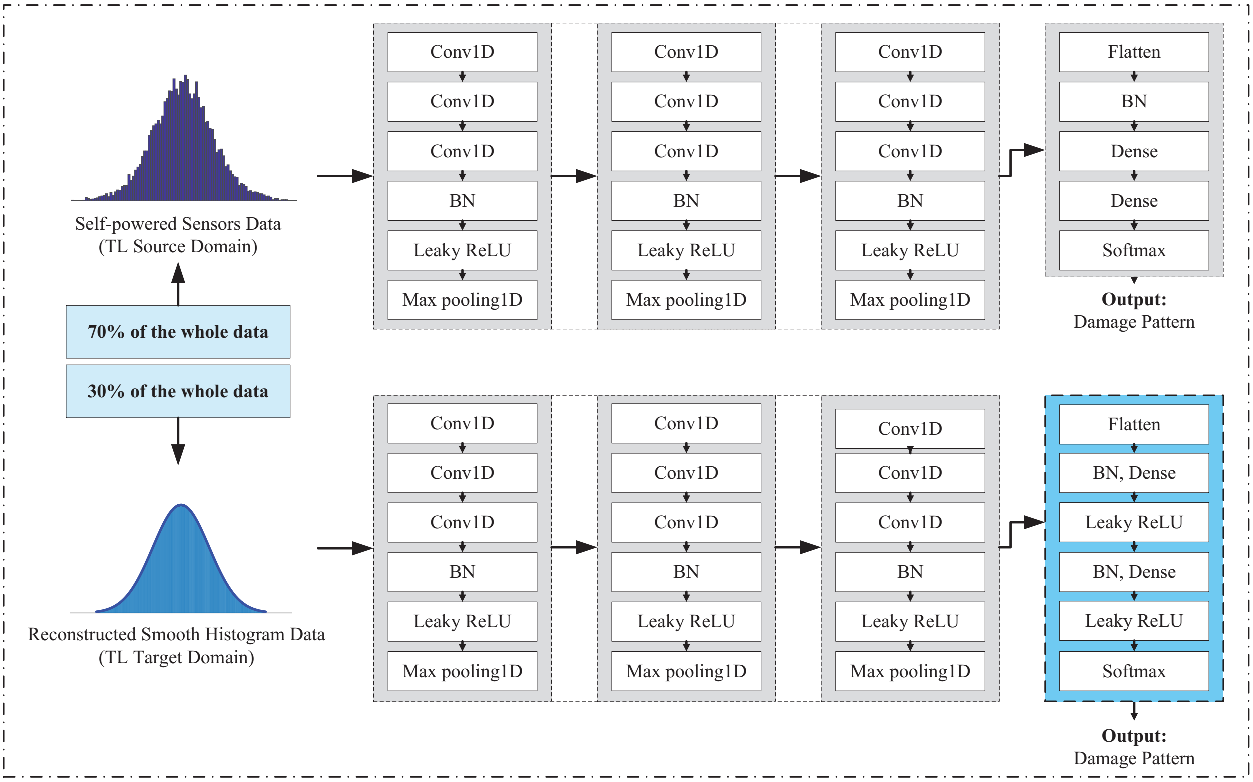


FIGURE 6 TL procedure and CNN configurations of the pretrained (top) and fine-tuned (bottom) models; tenfold cross-validation is used for each model

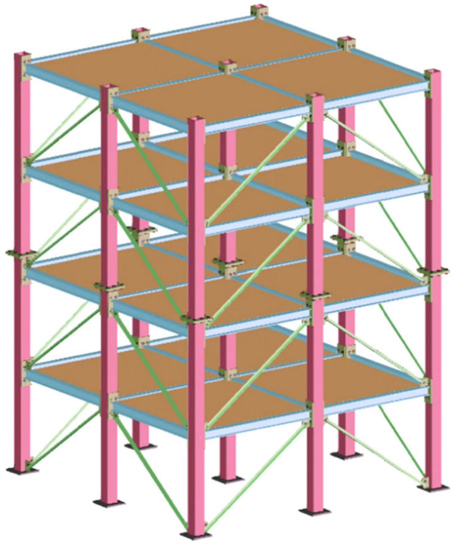
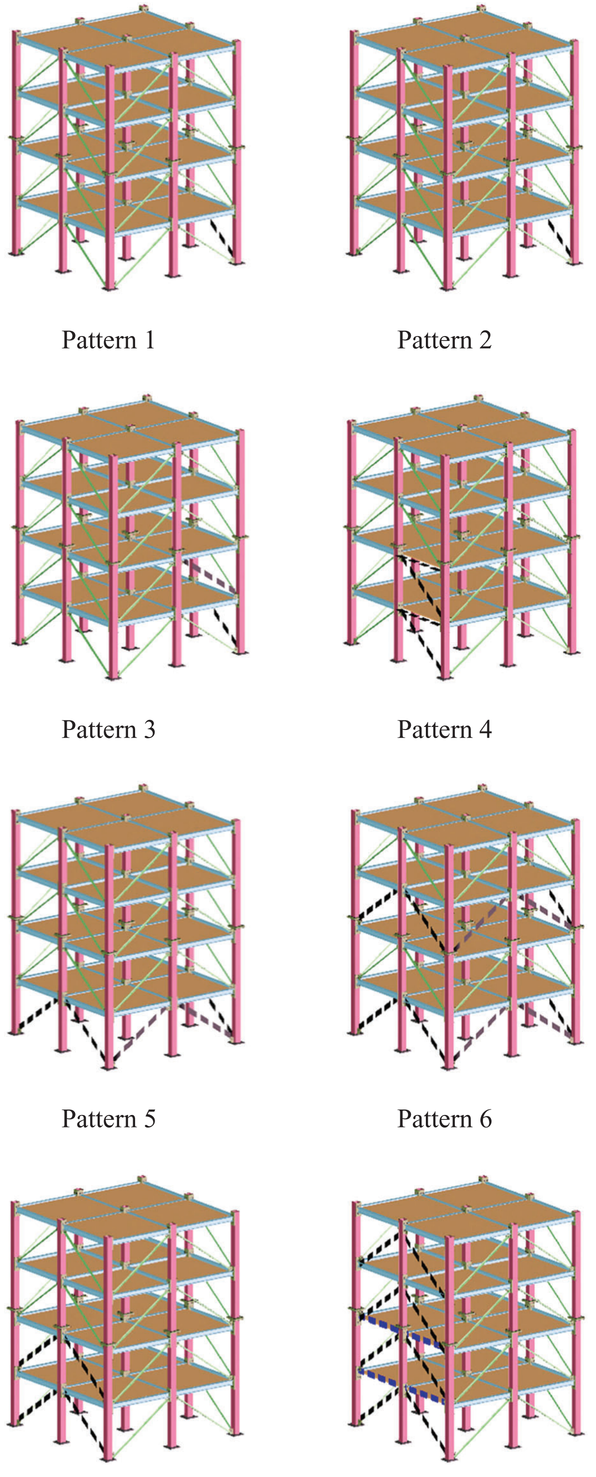


FIGURE 7 The analytical model of the IASC-ASCE SHM benchmark (Johnson et al., 2004)

benchmark problem (Johnson et al., 2004). It is 3.6 m tall, and the plan is 2.5 m × 2.5 m. The sections of the members are hot rolled, and columns have stronger bending stiffness in the *x*-direction. Each floor has a pair of diagonal bracing members that can be removed to simulate damage. Each floor has four slabs that are 800 kg on the first floor, 600 kg on the second and third floors, and 400 kg on the fourth floor. Two different finite element models were developed to simulate the behavior of the benchmark problem (Caicedo et al., 2004; Johnson et al., 2004). The Euler–Bernoulli beams are used for columns and floor beams, and no bending stiffnesses are considered for braces. The first model assumes that each floor has three degrees of freedom (12-DOF shear frame), while the second model consists of 120-DOF that allow the floor slab to have out-of-plane motion and rotation in order to simulate the effects of modeling errors (Caicedo et al., 2004). In other words, for the 120-DOF model, the nodes of each floor have the same horizontal translational and vertical rotation, and the out-of-plane DOF of floors, such as vertical movements and rolling, are active. Sixteen accelerometers; four per floor (two in each direction), are assumed to measure the responses at the center column of the external frames.

In addition to Intact case, a total of eight damage patterns, P1–P8, are considered, which may be detectable using simple to complex algorithms. They are defined to validate the ability of the proposed methods in detecting various damage scenarios. The eight damage patterns are shown in Figure 8, some of which are given in Johnson et al. (2004). In damage patterns P1 and P2, one of the braces of the first floor is damaged by 30% and 100%, respectively. Damage pattern P3 is characterized with two complete brace failures on the first and the second floor as shown in the figure.



Pattern 1

Pattern 2

Pattern 3

Pattern 4

Pattern 5

Pattern 6

Pattern 7

Pattern 8

FIGURE 8 Damage patterns for the SHM benchmark problem

In damage pattern P4, two braces of the first and second floors are fractured; in addition, two of the floor beams are released from both end connections. In damage pattern P5, all of the braces from the first floor are removed. Similarly, in damage pattern P6, all the braces are removed from the first and third floors. For damage pattern P7, brace members at the first two floors in the west are removed. Finally, in the damage pattern P8, all of the braces are failed on one side, in addition to the floor beams of the frame in the first two floors. The MATLAB code for the structural model was obtained from the NEES Database for Structural Control and Monitoring Benchmark Problems (Dyke, 2011a).

The input ambient (wind) excitations are defined using independent filtered Gaussian white noise (Johnson et al., 2004), the noise level that is a percentage of RMS of the signal, is selected to be 10%. For each data sample, the sampling rate and duration are 100 Hz and 10 s, respectively. Therefore, 1,000 data samples are generated for each damage pattern for which, the signals contain 1,000 data points per sensor. A total of 16 sensors (four per floor level) were assumed to be located at the center of each floor beam along the perimeter. For one of the cases, the shaker-induced vibration is used for damage detection (Johnson et al., 2004). The damage pattern groups and the associated datasets are summarized in Table 2.

5 | RESULTS AND DISCUSSION

This section presents comparison and performance of the proposed SHM alternatives: (a) CNN using time-domain response data from conventional sensors (MEMS) and also using response data in the form of discrete histograms from self-powered sensors (compressed data) as depicted in Figure 3, and (b) CNN and TL using response data represented in terms of three statistical parameters only (extremely compressed data) as depicted in Figure 6.

5.1 | Damage identification using MEMS and self-powered sensor data

In order to describe and compare the performance of the classification models during the training and testing phases, training history and the confusion matrices are provided for each considered case in Table 2. Figures 9 and 10 present the training epoch with respect to the accuracy and loss history curves. The mean values from all of the folds are shown with solid lines (accuracy = ascending with left vertical axis, and loss = descending with right vertical axis), and the shaded areas around each curve represent the envelop ranges. From the curves for each training, it can be seen that using the MEMS raw acceleration response data, the training is fast and they converge quickly, except for Cases 1–3. This observation is deemed reasonable since the damage patterns 1 and 2 share similar characteristics in these groups. It is obvious that the automatic feature extraction using DL classification approach yields an approximately perfect classification based

TABLE 2 Datasets and damage detection goals for the SHM benchmark problem

Case	Excitation	Model	Damage patterns	Mass distribution	Dataset size[a]
1	Wind	12 DOF	Intact,1,2	Symmetric	1,000 × 16 × 3,000
2	Wind	12 DOF	Intact,3,4	Symmetric	1,000 × 16 × 3,000
3	Wind	120 DOF	Intact,1,2,3,4	Symmetric	1,000 × 16 × 5,000
4	Wind	12 DOF	Intact,5,6	Symmetric	1,000 × 16 × 3,000
5	Wind	120 DOF	Intact,5,6	Symmetric	1,000 × 16 × 3,000
6	Shaker	12 DOF	Intact,5,6	Symmetric	1,000 × 16 × 3,000
7	Wind	120 DOF	Intact,7,8	Symmetric	1,000 × 16 × 3,000
8	Wind	120 DOF	Intact,1,2,3,4,5,6,7,8	Symmetric	1,000 × 16 × 9,000

[a]Datasets were generated using MATLAB code. The three dimensions of the datasets indicate the time-history signal length, number of sensors or channels, and number of simulated samples, respectively.

on the MEMS (time-domain) data for this specific benchmark problem.

On the other hand, the accuracy and loss history plots associated with the self-powered sensor data (discrete histograms) demonstrate that the main features can be extracted using the CNNs, despite the compressed nature of the data. These features allow faster convergence during the training phase in comparison to that associated with the MEMS data, and result in significantly higher levels of accuracy in both training and testing phases. Nonetheless, the training phase with an achieved 100% accuracy level does not indicate a perfect classification model. Therefore, the corresponding confusion matrices are used as another measure for determining the classification accuracy (diagonal elements) as well as misclassifications (off-diagonal elements) for each individual damage pattern. In the confusion matrices, all of the integers of the diagonal elements indicate the correctly classified data samples, and the off-diagonal elements are misclassified. The sum of all the integers is equal to the total number of the datasets since the *k*-fold cross-validation method is used (i.e., 10% of the available response data used as test dataset per fold). The ratio of each integer to the total number of response data is shown in percentage. The overall accuracy score is shown in the lower right corner of the matrices, while the prediction accuracy scores for the individual damage patterns are shown in the last column and row. P1–P8 class names indicate the pattern number for each damage class in confusion matrices.

For Case 1, using both MEMS and self-powered sensor data, P2 is identified accurately, but there is an overlap between P1 and Intact patterns. The overall score for the self-powered-based models is higher for Case 1, particularly considering the computational costs due to the dimensions of the input data and the epochs. It is noted that the architecture of the models remained the same for both sensor data types. Similar results were obtained for Case 2 for which the overall scores are almost 99.9%. The training history indicates a faster convergence when self-powered sensor data are used. A similar conclusion can be made when all of the first four patterns are combined as Case 3. For Cases 4 through 6, the self-powered sensors data does not lead to high accuracies in comparison to MEMS data that result in classification scores of 100%. However, both models accurately classify patterns Intact, P7, and P8 (Case 7). Finally, for Case 8 that includes all of the eight damage patterns, the overall scores of 96.5% and 98.1% are achieved for CNN models based on the data from MEMS and self-powered sensors, respectively, which indicates the feasibility of using CNNs for damage detection using the self-powered sensor data (Figure 11).

5.2 | Damage identification using extremely compressed data using TL-based approach

Using the same learning rate as specified for the training phase of the source domain, the CNN models were fine-tuned according to Figure 6. In order to compare the training and testing phases for the source and target domains, the categorical cross-entropy loss and the corresponding accuracy history are provided for all of the eight cases in Figure 12. The mean values of the *k*-fold cross-validation results are plotted in these figures. The accuracy curves for the source domain (70% of self-powered sensors data in the form of discrete histograms; compressed) are shown with dashed lines and those for target domain (30% of the self-powered sensors data in form of three-parameter distributions; extremely compressed) are presented with solid lines. The faster convergence of the training phase of the target domain with the DL models that include the frozen CNN layers indicates that the convolution layers carry important information from the source domain while the classification mainly relies on the extremely compressed response data. Furthermore, the classification performance for the fine-tuned models is shown in Figure 13. Again, the mean accuracy scores are obtained from the *k*-fold cross-validation. From the accuracy scores of the test sets for the source and target domains, it is observed that the three parameters; mean (μ), variance (σ), and a scale factor (SF), are sufficient for identification of damage classes for which the network is trained through TL-based fine-tuning. An exception for Case 3 is noted in which the use of

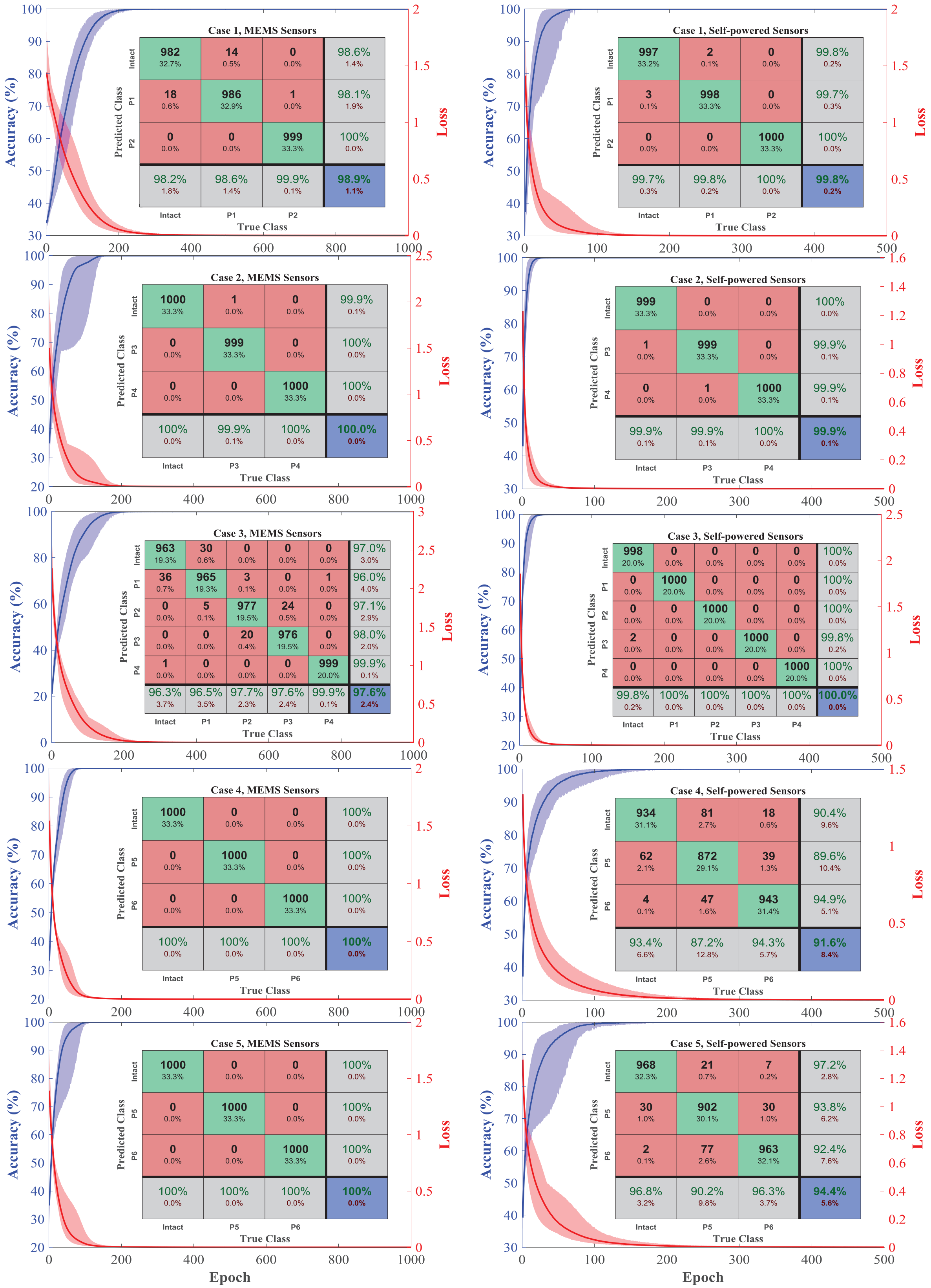


FIGURE 9 Training accuracy/loss history and the corresponding total confusion matrices of the test datasets: Cases 1–5

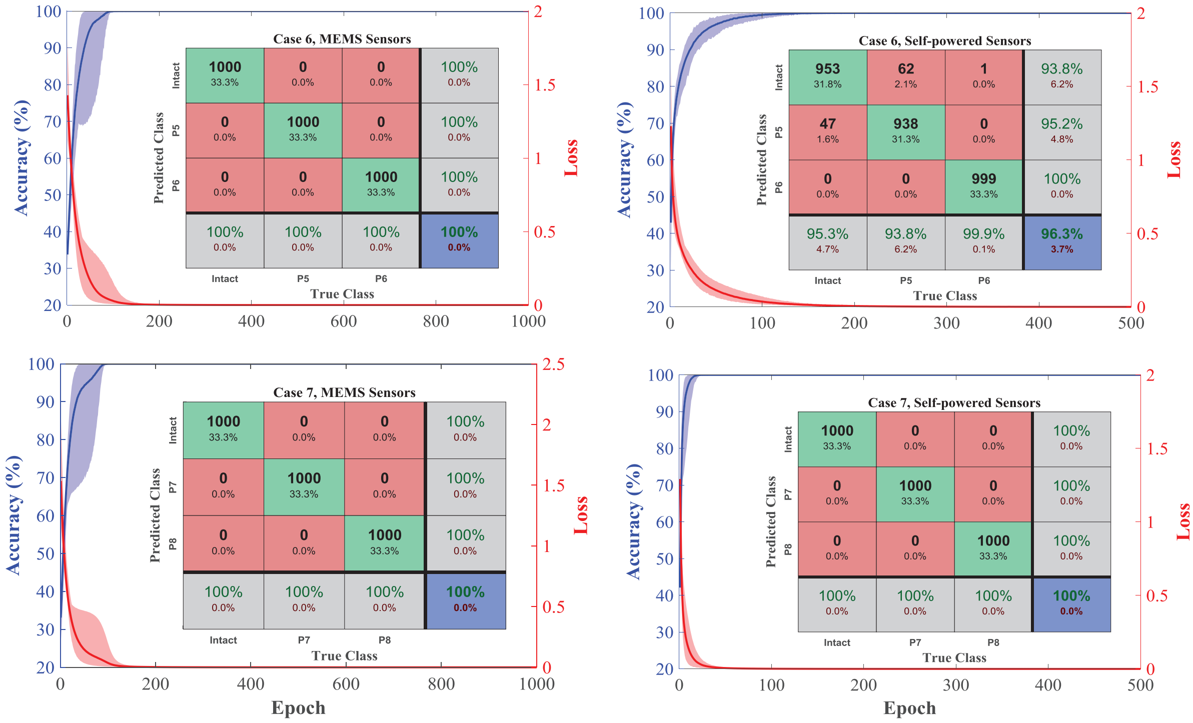
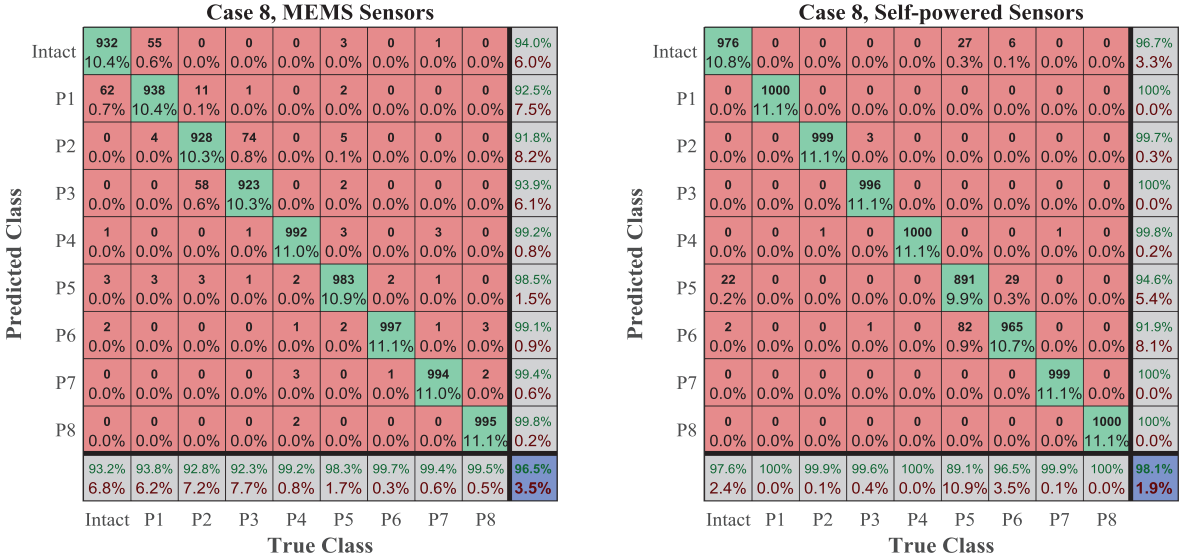


FIGURE 10 Training accuracy/loss history and the corresponding total confusion matrices of the test datasets: Cases 6–7



Case 8, MEMS Sensors

Predicted Class \ True Class	Intact	P1	P2	P3	P4	P5	P6	P7	P8	
Intact	932 10.4%	55 0.6%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	1 0.0%	0 0.0%	94.0% 6.0%
P1	62 0.7%	938 10.4%	11 0.1%	1 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	92.5% 7.5%
P2	0 0.0%	4 0.0%	928 10.3%	74 0.8%	0 0.0%	5 0.1%	0 0.0%	0 0.0%	0 0.0%	91.8% 8.2%
P3	0 0.0%	0 0.0%	58 0.6%	923 10.3%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	93.9% 6.1%
P4	1 0.0%	0 0.0%	0 0.0%	1 0.0%	992 11.0%	3 0.0%	0 0.0%	3 0.0%	0 0.0%	99.2% 0.8%
P5	3 0.0%	3 0.0%	3 0.0%	1 0.0%	2 0.0%	983 10.9%	2 0.0%	1 0.0%	0 0.0%	98.5% 1.5%
P6	2 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	2 0.0%	997 11.1%	1 0.0%	3 0.0%	99.1% 0.9%
P7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	1 0.0%	994 11.0%	2 0.0%	99.4% 0.6%
P8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	995 11.1%	99.8% 0.2%
	93.2% 6.8%	93.8% 6.2%	92.8% 7.2%	92.3% 7.7%	99.2% 0.8%	98.3% 1.7%	99.7% 0.3%	99.4% 0.6%	99.5% 0.5%	96.5% 3.5%

Case 8, Self-powered Sensors

Predicted Class \ True Class	Intact	P1	P2	P3	P4	P5	P6	P7	P8	
Intact	976 10.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	27 0.3%	6 0.1%	0 0.0%	0 0.0%	96.7% 3.3%
P1	0 0.0%	1000 11.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P2	0 0.0%	0 0.0%	999 11.1%	3 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
P3	0 0.0%	0 0.0%	0 0.0%	996 11.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
P4	0 0.0%	0 0.0%	1 0.0%	0 0.0%	1000 11.1%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	99.8% 0.2%
P5	22 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	891 9.9%	29 0.3%	0 0.0%	0 0.0%	94.6% 5.4%
P6	2 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	82 0.9%	965 10.7%	0 0.0%	0 0.0%	91.9% 8.1%
P7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	999 11.1%	0 0.0%	100% 0.0%
P8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1000 11.1%	100% 0.0%
	97.6% 2.4%	100% 0.0%	99.9% 0.1%	99.6% 0.4%	100% 0.0%	89.1% 10.9%	96.5% 3.5%	99.9% 0.1%	100% 0.0%	98.1% 1.9%

FIGURE 11 The total confusion matrices of the test datasets: Case 8

compressed response data results in lower accuracy score. Further model evaluation metrics are provided in the following section.

5.3 | Evaluation metrics

In addition to confusion matrices, the accuracy, precision, recall, and F1 scores were calculated for the test datasets from all folds and the box plots are shown in Figures 14–16. The metrics are defined as follows, and more details are available from scikit-learn documentation (Pedregosa et al., 2011).

$$\text{Accuracy}\ (y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i) \tag{7}$$

$$\text{Precision}\ (y, \hat{y}) = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall}\ (y, \hat{y}) = \frac{TP}{TP + FN} \tag{9}$$

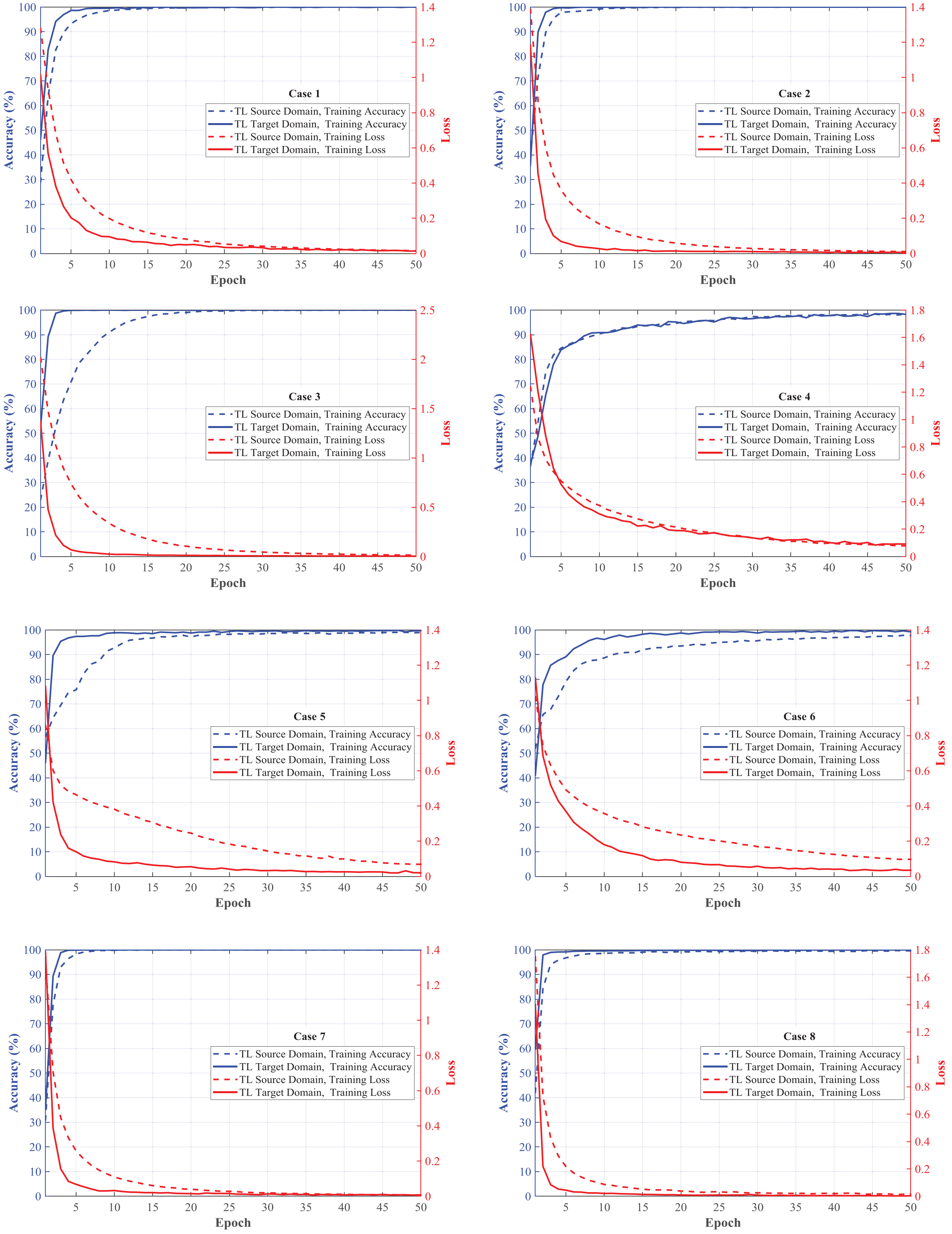


FIGURE 12 Training history of the source and target domain datasets

$$F_\beta = (1+\beta^2)\frac{P \times R}{\beta^2 P + R}, (F_1 = F_{\beta=1}) \quad (10)$$

where, $1(x)$ is the indication function; $\hat{y}$ and y are the predicted true labels; TP is the number of true positives; FP is the number of false positives; and FN is the number of false negatives.

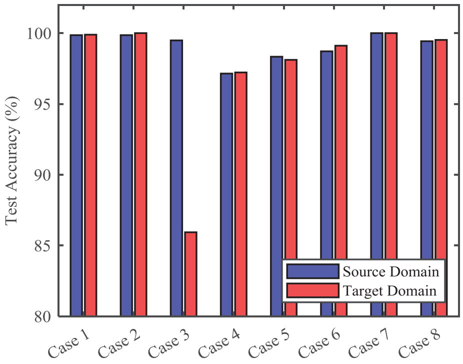


FIGURE 13 Mean accuracy scores of the test subsets of the pretrained (source domain) and fine-tuned (target domain) models

P and R indicate precision and recall scores, respectively. Figures 14–16 show the box plots of the four mentioned scores for the reference CNN model that uses MEMS data for classification of the defined patterns. For Cases 2 and 4–7, the obtained accuracy, precision, recall, and F1 scores are approximately 100%, and for the Cases 1, 3, and 8, the mean scores fall between 95% and 99%. Using the self-powered sensors data (discrete histograms), similar results are achieved, with the exception of Case 4, for which the scores fall between 85% and 95% with the mean values of approximately 91%. In addition, 50% of the *k*-folds for Cases 5, 6, and 8 have test scores between 91% and 96%, 95% and 98%, and 97% and 99%, respectively. For Cases 1, 2, and 7 using the fined-tuned models based on the compressed data nearly perfect scores are obtained; however, the variance for Case 3 is larger with precision scores between 50% and 100%.

In order to further investigate and verify the performance of each CNN model, the receiver operating characteristic (ROC) is used to compare the accuracy of the proposed methods with respect to each damage class (Pan, Azimi, Gui et al., 2017; Pan, Azimi, Yan et al., 2018). More information on ROC curves is presented by (J. Fan, Upadhye, & Worster, 2006). The ROC curve that is closer to the upper-left corner of the plot demonstrates a higher accuracy. In addition to ROC curves, the area under the curve (AUC) indicates the level of accuracy in identifying damage location as well as the performance in the presence of noise in the measured response data. The maximum value for AUC is 1, which corresponds to a perfect classifier, and an acceptable value is AUC ≥ 75% (Pan, Azimi, Yan, et al., 2017). The performance curves that factor in scores of the negative classes for a multiclass classification can be defined as follows:

$$\begin{aligned}\text{Score } s_{\text{diff}} &= \text{scores (positive class)}\\ &\quad -\max\text{(scores (negative classes))}\end{aligned} \tag{11}$$

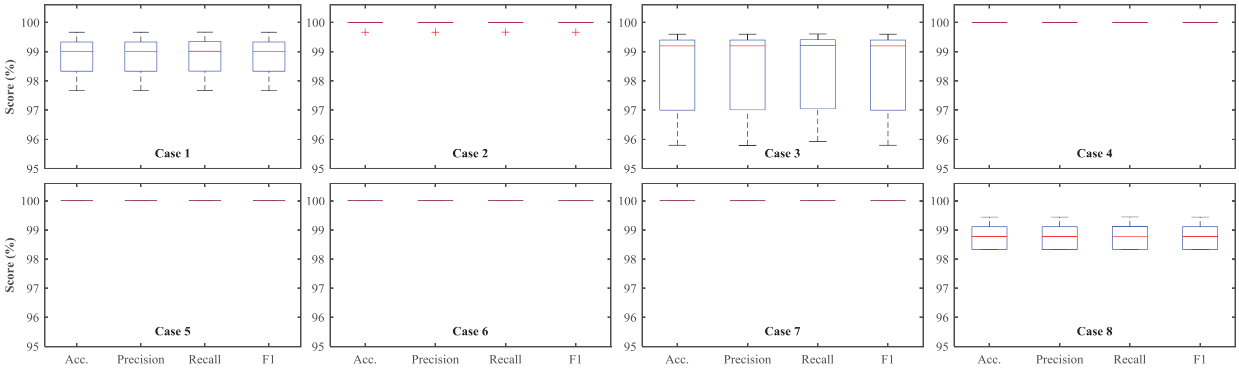


FIGURE 14 Accuracy, precision, recall, and F1 scores based on the MEMS data

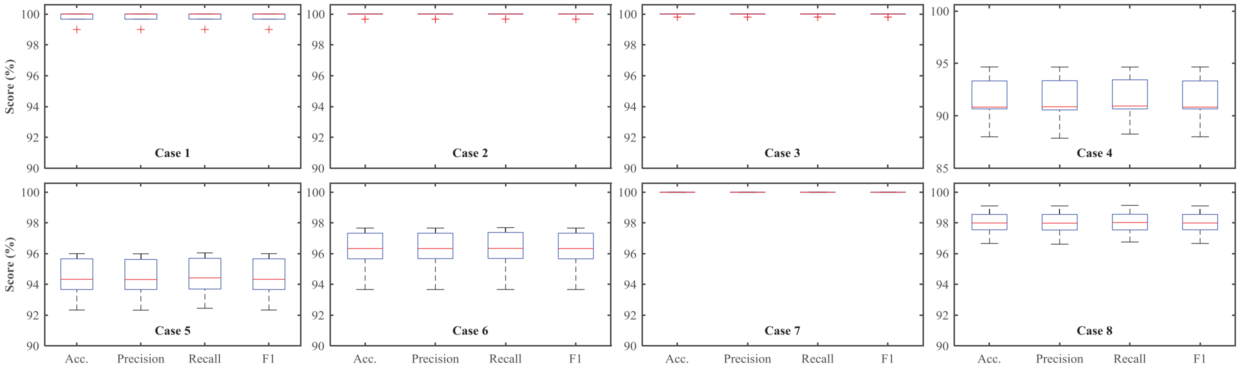


FIGURE 15 Accuracy, precision, recall, and F1 scores based on the self-powered data

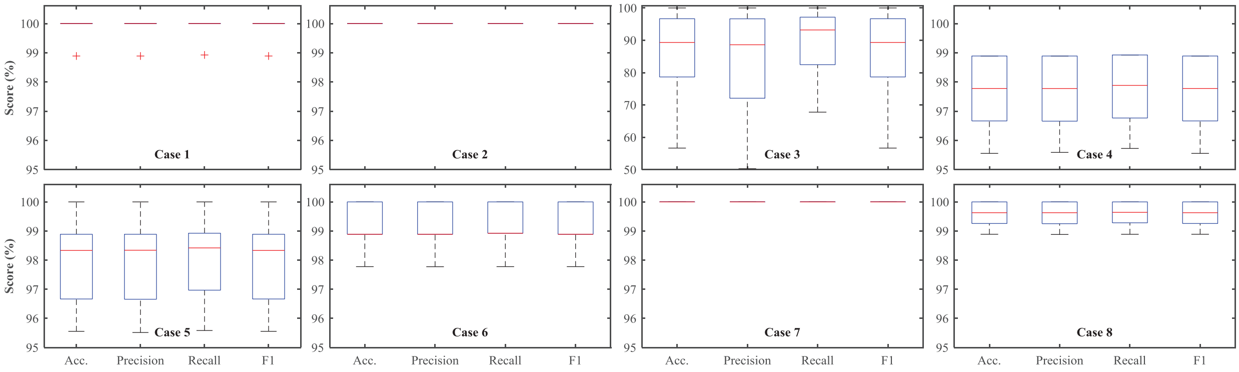


FIGURE 16 Accuracy, precision, recall, and F1 scores based on the extremely compressed data using TL

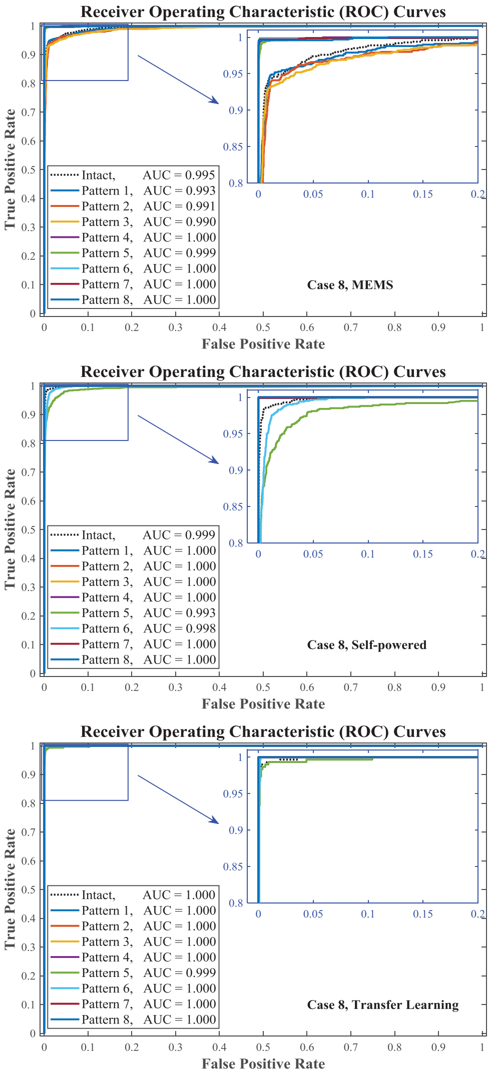


FIGURE 17 Performance curves of the three different SHM approaches

where, *scores*(*) is the prediction probabilities for each instance of the test datasets (Chollet, 2015). Figure 17 shows the ROC curves and the AUC of each damage pattern, which is obtained from all the *k*-fold test data sets. It can be seen from the ROC curves associated with the CNNs using the MEMS and self-powered sensors data result in good classification. The other performance curves using the three-parameter extremely compressed response data and TL-based method give the AUC of approximately one for all damage patterns except P5.

6 | EXPERIMENTAL VERIFICATION

The proposed CNN models were further used to validate the application of the method for real-world problems. Therefore, the experimentally recorded vibration response data from two benchmark problems have been used (Figure 18). The first problem is the IASC-ASCE SHM benchmark building that is used to develop the numerical procedure in this study (Caicedo et al., 2004; Johnson et al., 2004). The damage classes for this building are defined as follows: (a) fully braced configuration; (b) missing all east side braces; (c) removed braces on all floors in one bay on SE corner; (d) removed braces on 1st and 4th floors in one bay on SE corner; (e) removed braces on 1st floor in one bay on SE corner; (f) removed braces on all floors on east face and 2nd floor braces on north face; (g) all braces removed on all faces; (h) configuration 7, plus loosened bolts on all floors—both ends of beams on east face, north side; (i) configuration 7, plus loosened bolts on floors 1 and 2—both ends of beams on east face, north side (Dyke, 2011b). For this benchmark structure, for each damage pattern, the dataset has a size of 60,000 × 16, which was split into data samples with the size of 1,000 × 16. The second structure is a steel frame with 5 × 6 bolted joints, which was recently tested at Qatar University Grandstand Simulator (QUGS) (Abdeljaber et al., 2018). The damage classes for the QUGS benchmark frame were defined based on the location of the loosened bolts (30 joints, 30 damage classes). Therefore, using 30 sensors deployed at each joint location on the horizontal beam flanges, the dataset is a matrix with the size of 262,144 × 30 for each damage scenario. The CNNs were trained using data samples with the size of

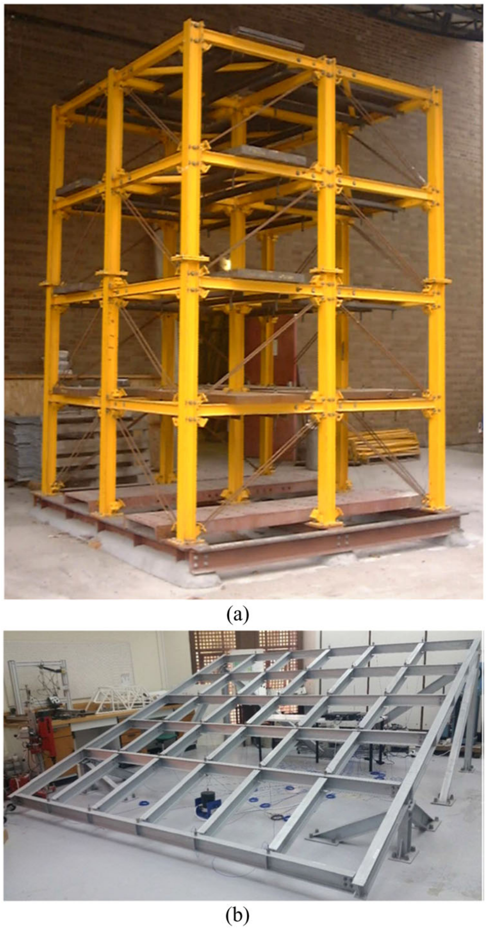


FIGURE 18 (a) IASC-ASCE SHM benchmark building (photo courtesy Prof. Carlos Ventura, UBC), and (b) QUGS structural steel frame (photo courtesy Prof. Onur Avci, QU)

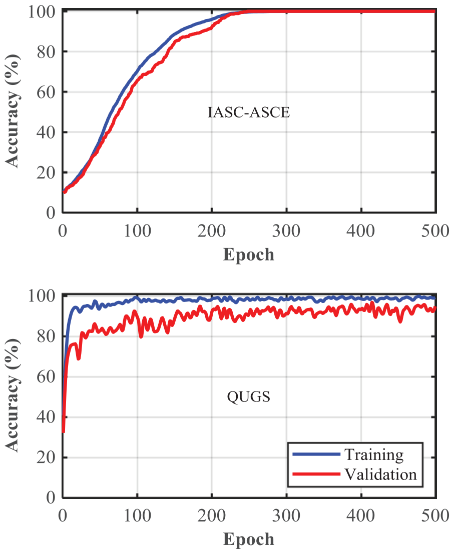
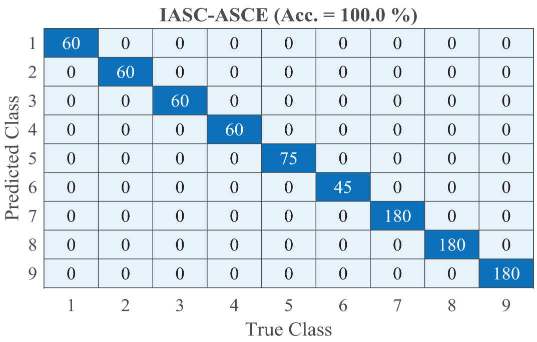


FIGURE 19 Training and validation histories of the experimental case studies (mean of fivefolds)



	1	2	3	4	5	6	7	8	9
1	60	0	0	0	0	0	0	0	0
2	0	60	0	0	0	0	0	0	0
3	0	0	60	0	0	0	0	0	0
4	0	0	0	60	0	0	0	0	0
5	0	0	0	0	75	0	0	0	0
6	0	0	0	0	0	45	0	0	0
7	0	0	0	0	0	0	180	0	0
8	0	0	0	0	0	0	0	180	0
9	0	0	0	0	0	0	0	0	180

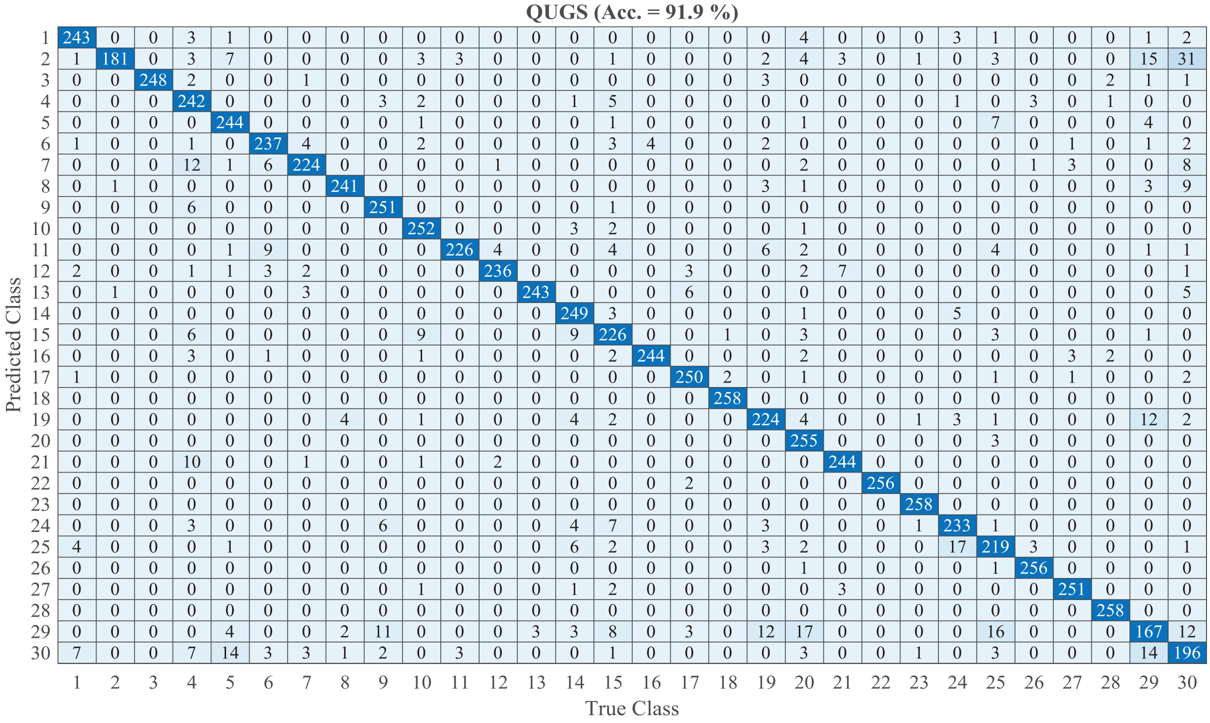
FIGURE 20 The total confusion matrix of the IASC-ASCE experimental case study (sum of fivefolds)

1,024 × 30. Further details regarding the experimental setup and the datasets are available from the original publications.

Before using the proposed method, the time-history response data from both structures were converted into discrete histograms to simulate the self-powered sensors that were mentioned earlier. Two dropout layers were added to the reference CNN architecture after the dense layers to avoid overfitting. In addition, in order to show the training phase performance, 10% of each training subset of fivefolds was used as the validation subset. The training and validation histories are shown in Figure 19 for both cases, and the confusion matrices are given in Figures 20 and 21. Due to the large number of damage classes, only the number of data samples (observation) is shown in the confusion matrices. For the IASC-ASCE SHM benchmark building frame, perfect classification scores were obtained using the proposed method, while for the QUGS benchmark steel frame, the accuracy level of 91.9% is achieved.

7 | SUMMARY AND CONCLUSION

In this paper, the use of CNN coupled with deep TL is introduced and demonstrated for general applications in vibration-based SHM on several benchmark structural systems. Three types of sensor data were considered: (1) time-domain data acquired with conventional accelerometers, (2) compressed data in the form of discrete histograms of response thresholds, acquired by self-powered sensors, and (3) extremely compressed three-parameter data in the form of continuous distributions. First, reference CNN models were developed and implemented using response data from conventional accelerometers and compressed data from self-powered sensors. Subsequently, a new TL-based strategy is proposed. The proposed method implements TL to fine-tune the pretrained CNNs that use the self-powered sensors data so that extremely compressed three-parameter response data can be used effectively. The extremely compressed data consist of three parameters as; mean, variance, and a scale factor for the measured



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	243	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	3	1	0	0	0	1	2
2	1	181	0	3	7	0	0	0	0	3	3	0	0	0	1	0	0	0	2	4	3	0	1	0	3	0	0	0	15	31
3	0	0	248	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	2	1	1
4	0	0	0	242	0	0	0	0	3	2	0	0	0	1	5	0	0	0	0	0	0	0	0	1	0	3	0	1	0	0
5	0	0	0	0	244	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	7	0	0	0	4	0
6	1	0	0	1	0	237	4	0	0	2	0	0	0	0	3	4	0	0	2	0	0	0	0	0	0	0	1	0	1	2
7	0	0	0	12	1	6	224	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	1	3	0	0	8
8	0	1	0	0	0	0	0	241	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	3	9
9	0	0	0	6	0	0	0	0	251	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	252	0	0	0	3	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	1	9	0	0	0	0	226	4	0	0	4	0	0	0	6	2	0	0	0	0	4	0	0	0	1	1
12	2	0	0	1	1	3	2	0	0	0	0	236	0	0	0	0	3	0	0	2	7	0	0	0	0	0	0	0	0	1
13	0	1	0	0	0	0	3	0	0	0	0	0	243	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	5
14	0	0	0	0	0	0	0	0	0	0	0	0	0	249	3	0	0	0	0	1	0	0	0	5	0	0	0	0	0	0
15	0	0	0	6	0	0	0	0	0	9	0	0	0	9	226	0	0	1	0	3	0	0	0	0	3	0	0	0	1	0
16	0	0	0	3	0	1	0	0	0	1	0	0	0	0	2	244	0	0	0	2	0	0	0	0	0	0	3	2	0	0
17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	250	2	0	1	0	0	0	0	1	0	1	0	0	2
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	258	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	4	0	1	0	0	0	4	2	0	0	0	224	4	0	0	1	3	1	0	0	0	12	2
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	0	0	0	0	3	0	0	0	0	0
21	0	0	0	10	0	0	1	0	0	1	0	2	0	0	0	0	0	0	0	0	244	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	256	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	258	0	0	0	0	0	0	0
24	0	0	0	3	0	0	0	0	6	0	0	0	0	4	7	0	0	0	3	0	0	0	1	233	1	0	0	0	0	0
25	4	0	0	0	1	0	0	0	0	0	0	0	0	6	2	0	0	0	3	2	0	0	0	17	219	3	0	0	0	1
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	256	0	0	0	0
27	0	0	0	0	0	0	0	0	0	1	0	0	0	1	2	0	0	0	0	0	3	0	0	0	0	0	251	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	258	0	0
29	0	0	0	0	4	0	0	2	11	0	0	0	3	3	8	0	3	0	12	17	0	0	0	0	16	0	0	0	167	12
30	7	0	0	7	14	3	3	1	2	0	3	0	0	0	1	0	0	0	0	3	0	0	1	0	3	0	0	0	14	196

FIGURE 21 The total confusion matrix of the QUGS experimental case studies (sum of fivefolds)

responses. In order to evaluate the performance of the proposed models, accuracy and loss histories, confusion matrices, performance curves along with the model evaluation metrics are presented for combinations of damage patterns.

The results indicate that SHM strategies lead to high classification accuracies using conventional MEMS data. In addition, the results from the similar CNN models highlight the potentials of self-powered sensors for which, the time-domain information may not be recoverable. Owing to the mean accuracy scores of 90–100%, the TL-based fine-tuning of previously trained CNNs, allows effective use of response data that can be represented by only three parameters without considerable loss in accuracy for the specific benchmark problems considered in this study.

Finally, it is noted that the damage patterns are predefined and other potential damage scenarios may not be detected correctly. On the other hand, extensive numerical simulations using highly nonlinear and accurate finite element models can be used to derive response data for practically all plausible damage scenarios that can be used to train CNNs. However, it is also possible to implement an unsupervised clustering procedure that does not need labeled data in advance. Furthermore, hybrid and parallel CNN models can be developed for large-scale real-world applications in that each CNN may be designed for specific subgroups of damage patterns and for respective damage identification and localization tasks.

ACKNOWLEDGMENT

This study was funded in part by the National Science Foundation (NSF) through Award No.: 1646420.

REFERENCES

Abdeljaber, O., Avci, O., Kiranyaz, M. S., Boashash, B., Sodano, H., & Inman, D. J. (2018). 1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data. *Neurocomputing*, *275*, 1308–1317.

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, *388*, 154–170.

Alavi, A. H., Hasni, H., Jiao, P., Borchani, W., & Lajnef, N. (2017). Fatigue cracking detection in steel bridge girders through a self-powered sensing concept. *Journal of Constructional Steel Research*, *128*, 19–38.

Amezquita-Sanchez, J. P., & Adeli, H. (2015). Synchrosqueezed wavelet transform-fractality model for locating, detecting, and quantifying damage in smart highrise building structures. *Smart Materials and Structures*, *24*(6), 065034.

Amezquita-Sanchez, J. P., Park, H. S., & Adeli, H. (2017). A novel methodology for modal parameters identification of large smart structures using MUSIC, empirical wavelet transform, and Hilbert transform. *Engineering Structures*, *147*, 148–159.

Aono, K., Kondapalli, S. H., Lajnef, N., Pekcan, G., Faridazar, F., & Chakrabartty, S. (2017, July). *Self-powered sensors to facilitate infrastructural Internet-of-Things for smart structures*. Paper presented at the 13th International Workshop on Advanced Smart Materials and Smart Structures Technology, University of Tokyo, Japan.

Avci, O., Abdeljaber, O., Kiranyaz, S., & Inman, D. (2017). Structural damage detection in real time: Implementation of 1D convolutional neural networks for SHM applications. In C. Niezrecki (Series Ed.), *Conference Proceedings of the Society for Experimental Mechanics Series: Vol. 7. Structural health monitoring & damage detection* (pp. 49–54). Cham, Switzerland: Springer.

Bao, Y., Chen, Z., Wei, S., Xu, Y., Tang, Z., & Li, H. (2019). The state of the art of data science and engineering in structural health monitoring. *Engineering*, *5*(2), 234–242.

Bao, Y., Tang, Z., Li, H., & Zhang, Y. (2019). Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, *18*(2), 401–421.

Caicedo, J. M., Dyke, S. J., & Johnson, E. A. (2004). Natural excitation technique and eigensystem realization algorithm for phase I of the IASC-ASCE benchmark problem: Simulated data. *Journal of Engineering Mechanics*, *130*(1), 49–60.

Carden, E. P., & Fanning, P. (2004). Vibration based condition monitoring: A review. *Structural Health Monitoring*, *3*(4), 355–377.

Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 731–747.

Chollet, F. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., & Courville, A. (2016). Recurrent batch normalization. *Preprint arXiv:1603.09025*.

Doebling, S. W., Farrar, C. R., Prime, M. B., & Shevitz, D. W. (1996). *Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review*. Los Alamos, NM: Los Alamos National Laboratory.

Dyke, S. (2011a). *Report on the building structural health monitoring problem phase 1 analytical*. Retrieved from https://datacenterhub.org/resources/2806

Dyke, S. (2011b). *Report on the building structural health monitoring problem phase 2 analytical*. Retrieved from https://datacenterhub.org/resources/2810

Fan, J., Upadhye, S., & Worster, A. (2006). Understanding receiver operating characteristic (ROC) curves. *Canadian Journal of Emergency Medicine*, *8*(1), 19–20.

Fan, W., & Qiao, P. (2011). Vibration-based damage identification methods: A review and comparative study. *Structural Health Monitoring*, *10*(1), 83–111.

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research*, *15*, 315–323. Retrieved from http://proceedings.mlr.press/v15/glorot11a.html

Hasni, H., Alavi, A. H., Lajnef, N., Abdelbarr, M., Masri, S. F., & Chakrabartty, S. (2017). Self-powered piezo-floating-gate sensors for health monitoring of steel plates. *Engineering Structures*, *148*, 584–601.

Hernandez, E., Roohi, M., & Rosowsky, D. (2018). Estimation of element-by-element demand-to-capacity ratios in instrumented SMRF buildings using measured seismic response. *Earthquake Engineering & Structural Dynamics*, *47*(12), 2561–2578.

Huang, Y., Beck, J. L., & Li, H. (2018). Multitask sparse Bayesian learning with applications in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *34*(9), 732–754.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Preprint arXiv:1502.03167*.

Johnson, E. A., Lam, H.-F., Katafygiotis, L. S., & Beck, J. L. (2004). Phase I IASC-ASCE structural health monitoring benchmark problem using simulated data. *Journal of Engineering Mechanics*, *130*(1), 3–15.

Kang, D., & Cha, Y. J. (2018). Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. *Computer-Aided Civil and Infrastructure Engineering*, *33*(10), 885–902.

Kaveh, A., & Dadras, A. (2018). Structural damage identification using an enhanced thermal exchange optimization algorithm. *Engineering Optimization*, *50*(3), 430–451.

Khoa, N. L. D., Alamdari, M. M., Rakotoarivelo, T., Anaissi, A., & Wang, Y. (2018). Structural health monitoring using machine learning techniques and domain knowledge based features. In J. Zhou & F. Chen (Series Eds.), *Human–Computer Interaction Series: Human and machine learning* (pp. 409–435). Cham, Switzerland: Springer.

Khodabandehlou, H., Pekcan, G., & Fadali, M. S. (2019). Vibration-based structural condition assessment using convolution neural networks. *Structural Control and Health Monitoring*, *26*(2), e2308.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *Preprint arXiv:1412.6980*.

Kiranyaz, S., Ince, T., & Gabbouj, M. (2016). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, *63*(3), 664–675.

Kong, X., Cai, C.-S., & Hu, J. (2017). The state-of-the-art on framework of vibration-based structural damage identification for decision making. *Applied Sciences*, *7*(5), 497.

Li, Z., Park, H. S., & Adeli, H. (2017). New method for modal identification of super high-rise building structures using discretized synchrosqueezed wavelet and Hilbert transforms. *The Structural Design of Tall and Special Buildings*, *26*(3), e1312.

Lin, Y.-z., Nie, Z.-h., & Ma, H.-w. (2017). Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *32*(12), 1025–1046.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proceedings of ICML*, *30*, 3.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *Proceedings of ICML*, 807–814.

Oh, B. K., Kim, K. J., Kim, Y., Park, H. S., & Adeli, H. (2017). Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings. *Applied Soft Computing*, *58*, 576–585.

Pan, H., Azimi, M., Gui, G., Yan, F., & Lin, Z. (2017). Vibration-based support vector machine for structural health monitoring. In J. Conte, R. Astroza, G. Benzoni, G. Feltrin, K. Loh, B. Moaveni (Series Eds.), *Lecture Notes in Civil Engineering: Vol. 5. Experimental vibration analysis for civil structures. EVACES 2017* (pp. 167–178). Cham, Switzerland: Springer.

Pan, H., Azimi, M., Yan, F., & Lin, Z. (2018). Time-frequency-based data-driven structural diagnosis and damage detection for cable-stayed bridges. *Journal of Bridge Engineering*, *23*(6), 04018033.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *Structural Design of Tall and Special Buildings*, *26*(18), e1400.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2), 237–244.

Roohi, M., Hernandez, E. M., & Rosowsky, D. (2019). Nonlinear seismic response reconstruction and performance assessment of instrumented wood-frame buildings—Validation using NEESWood Capstone full-scale tests. *Structural Control and Health Monitoring*, *26*(9), e2373.

Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. *Artificial neural networks–ICANN 2010* (pp. 92–101). Cham, Switzerland: Springer.

Sen, D., & Nagarajaiah, S. (2018). Data-driven approach to structural health monitoring using statistical learning algorithms. In E. Ottaviano, A. Pelliccio, & V. Gattulli (Eds.), *Mechatronics for cultural heritage and civil engineering* (pp. 295–305). Cham, Switzerland: Springer.

Sinou, J.-J. (2009). *A review of damage detection and health monitoring of mechanical systems from changes in the measurement of linear and non-linear vibrations*. Hauppauge, NY: Nova Science Publishers.

Sohn, H., Farrar, C. R., Hemez, F. M., Shunk, D. D., Stinemates, D. W., Nadler, B. R., & Czarnecki, J. J. (2003). *A review of structural health monitoring literature: 1996–2001*. Los Alamos, NM: Los Alamos National Laboratory.

Tang, Z., Chen, Z., Bao, Y., & Li, H. (2019). Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Structural Control and Health Monitoring*, *26*(1), e2296.

Tsogka, C., Daskalakis, E., Comanducci, G., & Ubertini, F. (2017). The stretching method for vibration-based structural health monitoring of civil structures. *Computer-Aided Civil and Infrastructure Engineering*, *32*(4), 288–303.

Yao, X. J., Yi, T. H., Qu, C., & Li, H. N. (2018). Blind modal identification using limited sensors through modified sparse component analysis by time-frequency method. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 769–782.

Yuen, K. V., & Huang, K. (2018). Identifiability-enhanced Bayesian frequency-domain substructure identification. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 800–812.

Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Chen, C., Yang, G., … Qiu, S. (2019). Automated pixel-level pavement crack detection on 3D asphalt surfaces with a recurrent neural network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(3), 213–229.

How to cite this article: Azimi M, Pekcan G. Structural health monitoring using extremely compressed data through deep learning. *Comput Aided Civ Inf*. 2020;35:597–614. https://doi.org/10.1111/mice.12517