Engineering Structures 172 (2018) 13–28

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct

Structural damage identification based on autoencoder neural networks and deep learning

Chathurdara Sri Nadith Pathirage[a], Jun Li[b], Ling Li[a,*], Hong Hao[b,c], Wanquan Liu[a], Pinghe Ni[b,d]

[a] School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Kent Street, Bentley, WA 6102, Australia
[b] Centre for Infrastructural Monitoring and Protection, School of Civil and Mechanical Engineering, Curtin University, Kent Street, Bentley, WA 6102, Australia
[c] School of Civil Engineering, Guangzhou University, Guangzhou 510006, China
[d] Department of Civil and Environmental Engineering, Hong Kong Polytechnic University, Hung Hom, Hong Kong

ARTICLE INFO

Keywords:
Autoencoders
Deep learning
Deep neural networks
Structural damage identification
Pre-training

ABSTRACT

Artificial neural networks are computational approaches based on machine learning to learn and make predictions based on data, and have been applied successfully in diverse applications including structural health monitoring in civil engineering. It is difficult to optimize the weights in the neural networks that have multiple hidden layers due to the vanishing gradient issue. This paper proposes an autoencoder based framework for structural damage identification, which can support deep neural networks and be utilized to obtain optimal solutions for pattern recognition problems of highly non-linear nature, such as learning a mapping between the vibration characteristics and structural damage. Two main components are defined in the proposed framework, namely, dimensionality reduction and relationship learning. The first component is to reduce the dimensionality of the original input vector while preserving the required necessary information, and the second component is to perform the relationship learning between the features with the reduced dimensionality and the stiffness reduction parameters of the structure. Vibration characteristics, such as natural frequencies and mode shapes, are used as the input and the structural damage are considered as the output vector. A pre-training scheme is performed to train the hidden layers in the autoencoders layer by layer, and fine tuning is conducted to optimize the whole network. Numerical and experimental investigations on steel frame structures are conducted to demonstrate the accuracy and efficiency of the proposed framework, comparing with the traditional ANN methods.

1. Introduction

Civil infrastructure including bridges and buildings etc., are crucial for a society to well function. They may deteriorate progressively and accumulate damage during their service life due to fatigue, overloading and extreme events, such as strong earthquake and cyclones. Structural Health Monitoring (SHM) provides practical means to assess and predict the structural performance under operational conditions. It is usually referred as the measurement of the critical responses of a structure to track and evaluate the symptoms of operational incidents, anomalies, and deterioration that may affect the serviceability and safety [1]. Numerous efforts have been devoted to develop vibration based structural damage identification methods by using vibration characteristics of structures [2]. These methods are based on the fact that changes in the structural physical parameters, such as stiffness and mass, will alter the structural vibration characteristics as well, i.e. natural frequencies and mode shapes. Structural damage identification based on changes in vibration characteristics of structures can be formulated as a pattern-recognition problem.

One of the most significant challenges associated with the vibration based methods is that they are susceptible to uncertainties in the damage identification process, such as, finite element modelling errors, noises in the measured vibration data and environmental effect etc. Artificial intelligence techniques, such as Artificial Neural networks (ANN) [3] and Genetic Algorithms (GA) [4], and Swarm Intelligence methods [5,6] are computational approaches based on machine learning to learn and make predictions based on data, and have been applied successfully in diverse applications including SHM in civil engineering. Yun et al. [7] estimated the structural joint damage from modal data via an ANN model. Noise injection learning with a realistic noise level for each input component was found to be effective in better understanding the noise effect in this work. Later, the mode shape differences or the mode shape ratios before and after damage were used as the input to the neural networks to reduce the effect of the modelling errors in the baseline finite element model [8]. Measured frequency response functions (FRF) were analysed by using Principal Component

* Corresponding author.
E-mail address: L.Li@curtin.edu.au (L. Li).

https://doi.org/10.1016/j.engstruct.2018.05.109
Received 20 January 2018; Received in revised form 22 April 2018; Accepted 30 May 2018
0141-0296/

Analysis (PCA) for data reduction, and the compressed FRFs represented by the most significant components were then used as the input to ANN for structural damage detection [9]. Ni et al. [10] investigated the construction of appropriate input vectors to neural networks for hierarchical identification of structural damage location and extent from measured modal properties. The neural network is first trained to locate the damage, and then re-trained to evaluate the damage extent with several natural frequencies and modal shapes. Yeung and Smith [11] generated the vibration feature vectors from the response spectra of a bridge under moving traffic as the input to neural networks for examination. It was shown that the sensitivity of the neural networks maybe adjusted so that a satisfactory rate of damage detection is achieved even in the presence of noisy signals. Bakhary et al. [12] proposed a statistical approach to account for the effect of uncertainties in developing an ANN model. Li et al. [13] used pattern changes in frequency response functions and ANN to identify structural damage. Later, Bandara et al. [14] used PCA to reduce the dimension of the measured FRF data and transformed it as new damage indices. ANN was then employed for the damage localization and quantification. Dackermann et al. [15] utilized cepstrum based operational modal analysis and ANN for damage identification of civil engineering structures. The damages in the joints of a multi-storey structure can be identified effectively.

In general, neural networks are particularly applicable to problems where a significant amount of information is available, but an explicit algorithm for processing them is difficult to specify. The weights associated with the mapping functions that make the neural networks exhibit desired behavior are obtained from training a large amount of data. Back propagation based on gradient descent method is one of the most traditional training algorithms, which has been found to be effective providing: (1) Initial weights are close enough to a good solution; (2) Computers are fast enough; and (3) Data sets are big enough. However, it is difficult to optimize the weights in the networks that have multiple hidden layers due to the vanishing gradient issue and convergence to local minima [16]. This problem has been a bottleneck for ANN with shallow architecture models. For network models with a deep structure, the major difficulty has been to optimize the weights of the hidden layers that are close to the input layer.

Hinton and Salakhutdinov [17] introduced the concept of deep learning to reduce the dimensionality of data and tackle the above three limitations. Deep neural networks have attracted wide-spread attention, mainly since they outperform alternative machine learning methods such as support vector machine and kernel machines in numerous important applications. The original applications mainly focused on face detection, objective recognition, speech recognition and detection, and natural language processing [18,19]. Recently it has been developed for fault detection and diagnosis in mechanical engineering [20,21]. A study on using 1-D Convolutional Neural Networks for detecting the structural damage has been conducted in 2017 [22]. It should be noted that sensors have to be placed on all the joints in a space frame structure to detect the damage in that work. Cha et al. [23] proposed a vision-based method using a deep architecture of convolutional neural networks for detecting concrete surface cracks without calculating the defect features. Nadith et al. [24] explored using the Autoencoders model to perform the feasibility study on pattern recognition for structural health monitoring with numerical simulations only. No system uncertainties and measurement noises have been considered.

It has been demonstrated that deep learning based methods are favorable in optimizing networks with multiple hidden layers [25]. Autoencoders are unsupervised training models. The aim of an autoencoder is to learn a representation for a set of data, usually for the purpose of dimensionality reduction. Deep autoencoder is utilized for effective feature learning through hierarchical non-linear mappings via the multiple hidden layers of the model [26]. The training of autoencoders is usually performed in two stages: pre-training and fine-tuning. The pre-training is usually performed layer by layer and multiple simple autoencoders are used to initialize the layer weights that are close enough to a good solution. The fine-tuning is performed to optimize the multiple layers of the whole network together with respect to the final objective function. Autoencoders have been used in the "deep architecture" approaches [17,27–30] with unsupervised learning algorithms.

This paper proposes an autoencoder based framework for structural damage identification, which can be utilized to learn optimal solutions for pattern recognition problems of highly non-linear nature, such as learning a mapping between the vibration characteristics and structural damages. The proposed framework consists of two main components, namely, dimensionality reduction and relationship learning. The first component is to reduce the dimensionality of the original input vector while preserving the necessary information required, and the second component is to perform the relationship learning between the features with the reduced dimensionality and the stiffness reduction parameters of the structure. Vibration characteristics, such as natural frequencies and mode shapes, are used as the input and the structural damages are considered as the output vector. A pre-training scheme is performed to train the hidden layers in the autoencoders layer by layer, and fine tuning is conducted to optimize the whole network. Numerical studies and experimental validations on steel frame structures are conducted to demonstrate the accuracy and efficiency of the proposed framework, comparing with the traditional ANN methods.

2. Autoencoder based framework for structural health monitoring

An autoencoder based framework, which can support deep neural networks, is proposed for structural health monitoring. A typical autoencoder model will be briefly described in Section 2.1, and the proposed Autoencoder based framework will be presented in Section 2.2. The proposed framework will be applied for structural damage identification, which is a pattern recognition problem based on the fact that the changes in structural physical material properties, i.e. stiffness, will alter the structural vibration characteristics, i.e. natural frequencies and mode shapes. In this study, natural frequencies and mode shapes serve as the input to the proposed framework and the output will be the elemental stiffness reduction parameters representing structural health conditions. Training methods for the proposed framework will also be described in Section 2.2.

2.1. Autoencoder

A traditional autoencoder [26] consists of two core segments: encoder and decoder with a single hidden layer.

Encoder: The deterministic mapping $f(\bar{x})$, which transforms a d-dimensional input vector $\bar{x} \in \Re^d$ into a r-dimensional hidden representation $\bar{h} \in \Re^r$, is called an encoder. Its typical form is an affine mapping followed by a nonlinear transformation, which can be expressed as follows

$$\bar{h} = f(\bar{x}) = \Phi(W\bar{x} + \bar{b}) \tag{1}$$

where $W \in \Re^{rxd}$ denotes the mapping weight matrix of the encoder, $\bar{b} \in \Re^r$ is the bias vector and Φ is the activation function, which is usually a squashing non-linear function and could be a sigmoid function or hyperbolic tangent function: $\Phi(\bar{x}) = sigmoid(\bar{x}) = 1/1 + e^{-\bar{x}}$ or $\Phi(\bar{x}) = \tanh(\bar{x}) = (e^{\bar{x}} - e^{-\bar{x}})/(e^{\bar{x}} + e^{-\bar{x}})$. A non-squashing linear function, such as $\Phi(\bar{x}) = purelin(\bar{x}) = \bar{x}$, can also be used to output real values that do not fall into a specific range, where "*purelin*" is a linear transfer function.

Decoder: The mapping $g(\bar{h})$, which transforms the hidden representation $\bar{h}$ (observed in the step described above) back into a reconstructed vector $\bar{z} \in \Re^d$ in the input space, is called a decoder. The typical form of a decoder is also an affine mapping optionally followed by a squashing nonlinearity

$$\bar{z} = g(\bar{h}) = \Phi(\widehat{W}\bar{h} + \widehat{\bar{b}}) \tag{2}$$

where $\widehat{W} \in \Re^{dxr}$ is the weight matrix of the decoder, $\widehat{\bar{b}} \in \Re^{d}$ is the bias vector and Φ is the activation function described above.

To optimize the parameters W, $\bar{b}$, $\widehat{W}$, $\widehat{\bar{b}}$, usually the mean squared error is employed as the cost function as follows

$$[W^*, \bar{b}^*, \widehat{W}^*, \widehat{\bar{b}}^*] = \boldsymbol{argmin}_{W,\bar{b},\widehat{W},\widehat{\bar{b}}} \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\|g(f(x^{(i)}))-x^{(i)}\|^2\right) \tag{3}$$

where m is the number of samples, $x^{(i)}$ is the ith input, $f(\cdot)$ and $g(\cdot)$ mappings are the encoder and decoder functions respectively. The nonlinearity of the activation function shown in Eq. (3) is difficult to solve, thus the gradient descent algorithm is commonly employed.A typical autoencoder neural network in Eq. (3) tries to reconstruct the input. However if a response distinct from the input is used as the output of the decoder $g(\cdot)$, it can be considered as a kind of non-linear regression technique.

2.2. The proposed framework

Autoencoders can be used for various tasks, such as effective feature learning, dimensionality reduction and nonlinear regression etc. [27,28]. These functions are explored in the proposed framework for structural health monitoring to learn a compressed feature representation and form a nonlinear regression for accurate and robust structural damage detections.

The recent demonstrations of the potential of deep learning algorithms were achieved despite the serious challenge of training models with many layers of adaptive parameters. In general all instances of deep learning, the objective function is a highly non-convex function of the parameters, with the potential for many distinct local minima in the model parameter space. Hence the optimization algorithm may not be guaranteed to arrive at even a local minimum in a reasonable amount of time, but it often finds a very low value of the cost function quickly enough to be useful provided with a decent initialization for weights. The principal difficulty is that not all of these minima provide equivalent generalization errors but the weight initialization method for deep architectures. The standard training schemes (based on random initialization) tend to place the parameters in regions of the parameters space that generalize poorly—as was frequently observed empirically but rarely reported [25].

Hence the concept of layer wise pre-training of the network is introduced to find the weights that are close to the optimal. In this paper a set of simple autoencoders were used to perform this task. One of the claims of this paper is that powerful unsupervised and semi-supervised (or self-taught) learning is a crucial component in building successful learning algorithms for deep architectures aimed at approaching optimal solutions. If gradients of a criterion defined at the output layer become less useful as they are propagated backwards to lower layers, it is reasonable to believe that an unsupervised learning criterion defined at the level of a single layer could be used to move its parameters in a favourable direction. It would be reasonable to expect this if the single-layer learning algorithm discovered a representation that captures statistical regularities of the layer's input [27]. Also layer wise pre-training could be a way to naturally decompose the problem into sub-problems associated with different levels of abstraction. It is known that unsupervised learning algorithms can extract salient information about the input distribution. This information can be captured in a distributed representation, i.e., a set of features which encode the salient factors of variation in the input. A one-layer unsupervised learning algorithm could extract such salient features, but because of the limited capacity of that layer, the features extracted on the first level of the architecture can be seen as low-level features. It is conceivable that learning a second layer based on the same principle but taking as input the features learned with the first layer could extract slightly higher-level features. In this way, one could imagine that higher-level abstractions that characterize the input could emerge. In the latter stage, layers are pre-trained on performing the mapping between the learned salient features to the output. Note how in this process all learning could remain local to each layer, therefore side-stepping the issue of gradient diffusion that might be hurting gradient-based learning of deep neural networks, when we try to optimize a single global criterion.

The objective of the proposed framework is to learn the relationship between the structural vibration characteristics, i.e. natural frequencies and mode shapes, and the physical properties of structures, such as stiffness. Therefore the input to the framework are the modal information such as frequencies and mode shapes, while the elemental stiffness reduction parameters of structures are the output vector. The input feature vector including possibly many orders of natural frequencies and mode shapes is usually high dimensional. Learning a relationship directly from a high dimensional input will very likely be less accurate than using compressed features, since the high dimensional input feature may contain unnecessary information due to the redundancy in the data, as well as uncertainties such as measurement noise and finite element modelling errors. Therefore, structural damage identification in this study is performed in two main components in the proposed framework as shown in Fig. 1. The first component is to reduce the dimensionality of the original input vector while preserving necessary information required, and the second component is to perform the relationship learning between the compressed features with reduced dimensionality and the structural stiffness reduction parameters. Each component is defined with a specific objective optimization function, which will be described in the following sections.

The proposed framework is shown in Fig. 1. As mentioned above, there are two components in this framework, namely the dimensionality reduction and the relationship learning. In the dimensionality reduction component, an autoencoder based model with a deep architecture and nonlinear activation units is proposed to perform the nonlinear dimensionality reduction. A lower dimensional feature vector learned from this process is obtained to represent the given high dimensional data. It is worth noting that a pre-training scheme is conducted for training the first component. The quality of the dimensionality reduction process is evaluated by using the mean squared error (MSE) and the regression value (R-value) on the reconstruction accuracy of the original input feature. In the relationship learning component, a simple autoencoder model with a single hidden layer and nonlinear activation units is utilized to perform this regression task. MSE and R-Value are also used to evaluate the quality of the predictions

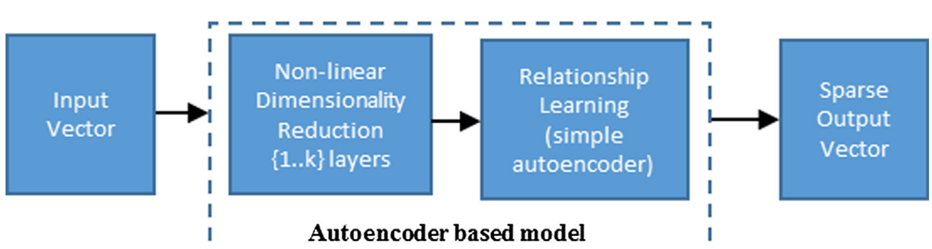

Fig. 1. Framework of Autoencoder based deep neural networks.

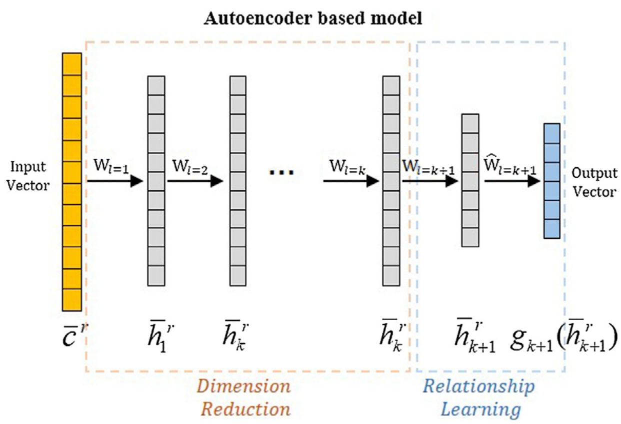

Fig. 2. Framework of the proposed Autoencoder based model.

on structural stiffness parameters.

These two components are employed sequentially and fine-tuning of the whole network is conducted to perform the joined optimization on the final objective function for learning the relationship from the original input vector to the final output. In this manner, the proposed framework is capable of retaining only the required information to establish the relationship between the learned compressed features and the final output in the form of stiffness reduction parameters. The detailed formulations of these two components are presented in the following sections.

2.2.1. Dimensionality reduction

An autoencoder model with a deep neural network architecture is trained for the dimensionality reduction, where the *1st* hidden layer is defined to perform the feature fusion of both the frequencies and mode shapes from the structure while the subsequent *2nd* to kth hidden layers further compress the features, as shown in Fig. 2. One can visualize this model as the encoding architecture of a typical deep autoencoder [29], but not strictly the generic deep autoencoder model with the decoding structure.

$\bar{c}^r$ represents the combined high dimensional input vector, including n structural natural frequencies and the corresponding $n \times t$ mode shape values

$$\bar{c}^r = [q_1^r, q_2^r, \dots, q_i^r, m_1^{q_1 r}, m_2^{q_1 r} \dots, m_t^{q_i r}, \dots, m_j^{q_i r}]^T \tag{4}$$

where q_i^r is the ith $(i = 1 \dots n)$ natural frequency included in the rth sample; $m_j^{q_i r}$ denotes the jth $(j = 1 \dots t)$ mode shape value corresponding to the ith frequency and t is the number of measurement points for describing a mode shape. A layer-wise pre-training scheme [29] is performed for all the layers of this dimensionality reduction component with the following cost function

$$J_{cost}^p(W, \bar{b}) = J_{MSE}^p(W, \bar{b}) + \lambda J_{weight}^p(W, \bar{b}) \tag{5}$$

with

$$J_{MSE}^p(W, \bar{b}) = \sum_{\tau=1}^{N} \|\bar{h}_{p-1}^r - g_p(f_p(\bar{h}_{p-1}^r))\|_2^2 \tag{6}$$

$$J_{weight}^p(W) = \frac{1}{2} \sum_{l=p-1}^{p} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (w_{ji}^{(l)})^2 \tag{7}$$

where $p = \{1 \dots k\}$ with k being the number of layers in the dimensionality reduction component, N is the number of data samples involved in the training, and g_p and f_p are the decoder and encoder functions of the *p-th* layer, respectively. $\bar{h}_{p-1}^r$ is the lower dimensional representation that is established in the $(p-1)^{th}$ layer for the rth sample where $\bar{h}_0^r = \bar{c}^r$. $w_{ji}^{(l)}$ represents the weighting coefficient in the weighting matrix $W^{(l)}$, and s_l denotes the number of neural units in the *l-th* layer. Encoder function f_p is set to be "tanh" (hyperbolic tangent) since the value 0 is contained in its activation region thus supports a sparse representation of the input when the activation of a hidden unit becomes 0. Decoder function g_p is set to be "purelin" since it needs to reconstruct the real values of the input. The factor "1/2" in Equation (7) is used to eliminate "2" when taking the gradient of the mean square error. This is to have a clear derivate of the cost function. Hence having multiplication factor on the cost will not change the optimal solution reached via the optimizing method. Furthermore, the L2-weight decay term denoted in Eq. (7) is added on the cost function as shown in Eq. (5) to avoid over-fitting in the overall training process. The motivation behind L2 (or L1) regularization on network weights in all layers is that by restricting the weights, constraining the network, it is less likely to over fit. Also this mechanism of regularization helps to constrain the number of hidden nodes in a layer during the pre-training by pushing the weights to zero (there by the making the inputs to a node close to zero, making the neuron's response less significant). In addition, L2-weight decay constraints the hidden nodes of a layer [29] thus allowing the model to utilize hidden layers with the number of hidden nodes same as its input. A typical autoencoder without L2-weight decay would learn the identity mapping from its input to output. The burden of choosing a suitable number of hidden nodes for each layer is handled to a certain extent with the introduction of this constraint. The optimal parameter value for λ in Eq. (5) is chosen via utilizing a validation dataset [31]. The compressed representation features learned in the kth layer $\bar{h}_k^r$, is then fed to a nonlinear relationship learning component that will be described in the following section.

2.2.2. Relationship learning

The relationship learning component, as shown in Fig. 2, is defined to perform the regression task utilizing the low dimensional feature learned at the kth layer, which is a better feature representation than the original input to predict the structural stiffness reduction parameters as the final output. A simple autoencoder model with only one hidden layer and a hyperbolic tangent activation function is defined to perform this task. The cost function for this model is defined as

$$J_{cost}^{k+1}(W, \bar{b}) = J_{MSE}^{k+1}(W, \bar{b}) + \lambda J_{weight}^{k+1}(W, \bar{b}) \tag{8}$$

with

$$J_{MSE}^{k+1}(W, \bar{b}) = \sum_{\tau=1}^{N} \|\bar{o}^r - g_{k+1}(f_{k+1}(\bar{h}_k^r))\|_2^2 \tag{9}$$

where g_{k+1} and f_{k+1} are respectively the decoder and the encoder functions of the $(k + 1)$th layer, $\bar{h}_k^r$ is the low dimensional representations obtained at the kth layer (also the last layer) of the dimensionality reduction component for the rth sample, and $\bar{o}^r$ is the labeled output vector, namely the pre-defined stiffness reduction parameters of the rth sample. The output of this relationship learning component is the predicted structural stiffness reduction parameters.

2.2.3. Fine-tuning

Once the optimal mapping weight coefficients and bias parameters of all the hidden layers are obtained with the pre-training scheme, the whole network is fine-tuned to optimize all the layers as a whole with the following cost function

$$J_{cost}^F(W, \bar{b}) = J_{MSE}^F(W, \bar{b}) + \lambda J_{weight}^F(W, \bar{b}) \tag{10}$$

with

$$J_{MSE}^F(W, \bar{b}) = \sum_{\tau=1}^{N} \|\bar{o}^r - p(\bar{c}^r)\|_2^2 \tag{11}$$

where $p(\bar{c}^r) = g_{k+1}(f_k(f_{k-1}(f_{k-2}(\dots(\bar{c}^r)))))$ is the predicted output vector through the activations of all the layers in both the dimensionality reduction and relationship learning components. The layer-wise pre-training and fine-tuning of the whole network are performed to

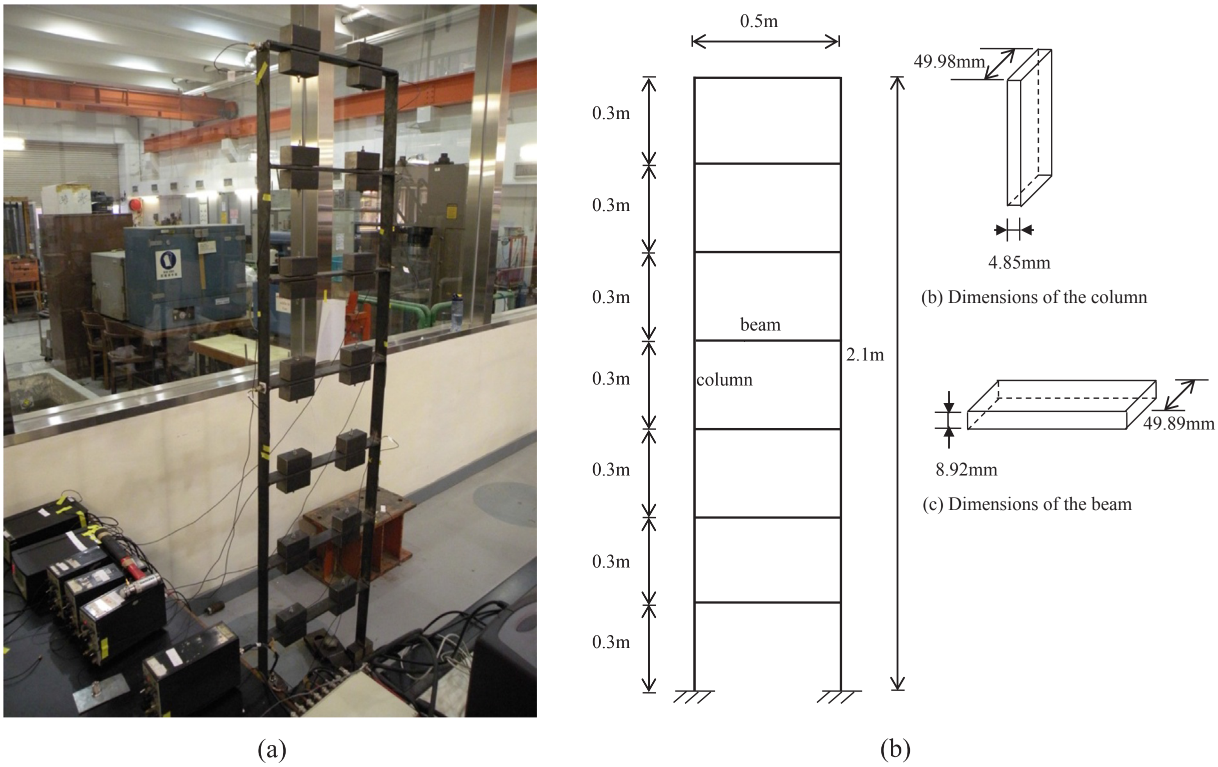

Fig. 3. Laboratory model and dimensions of the steel frame structure: (a) Steel frame model; (b) Dimensions.

improve the training efficiency and achieve a better accuracy of the proposed framework.

3. Numerical studies

In this section, the numerical model, data generation and pre-processing, and performance evaluation of the proposed framework will be presented. The accuracy and efficiency of using the proposed framework for structural damage identification will be evaluated with simulation data generated from a numerical finite element model. Both the uncertainties in the finite element modelling and measurement noise effect in the data will be considered.

3.1. Numerical model

A seven-storey steel frame structure is fabricated in the laboratory and the dimensions of the frame are shown in Fig. 3. The column of the frame has a total height of 2.1 m with 0.3 m for each storey. The length of the beam is 0.5 m. The cross-sections of the column and beam elements are measured as 49.98 mm × 4.85 mm and 49.89 mm × 8.92 mm, respectively. The measured mass densities of the column and beam elements are 7850 kg/m^3 and 7734.2 kg/m^3, respectively. The initial Young's modulus is taken as 210 GPa for all members. The connections between column and beam elements are continuously welded at the top and bottom of the beam section. Two pairs of mass blocks with approximately 4 kg weight each, are fixed at the quarter and three-quarter length of the beam in each storey to simulate the mass from the floor of a building structure. The bottoms of the two columns of the frame are welded onto a thick and solid steel plate which is fixed to the ground.

Fig. 4 shows the finite element model of the whole frame structure. It consists of 65 nodes and 70 planar frame elements. The weights of steel blocks are added at the corresponding nodes of the finite element model as concentrated masses. Each node has three DOFs (two translational displacements x, y and a rotational displacement θ), and the system has 195 DOFs in total. The translational and rotational restraints at the supports, which are Nodes 1 and 65, are represented initially by a large stiffness of 3×10^9 N/m and 3×10^9 N·m/rad, respectively. The initial finite element model updating has been conducted to minimize the discrepancies between the analytical finite element model and the experimental model in the laboratory. The detailed model updating process can be found in [32]. This updated finite element model is taken as the baseline model for generating the training, validation and testing data.

3.2. Data generation and pro-processing

Modal analysis is performed by using the baseline model to generate the input and output data to train the proposed framework. The first seven frequencies and the corresponding mode shapes at 14 beam-column joints are obtained. The elemental stiffness parameters are normalized to the range between 0 and 1, where 1 denotes the intact state and 0 denotes the completely damaged state. For example, if the stiffness parameter of a specific element is equal to 0.9, it means 10% stiffness reduction is introduced in this element. 12,400 data samples are generated from the baseline model including both single and multiple damage cases. In single element damage cases, the stiffness parameter for each element varies from 1, 0.99, 0.98,… to 0.7 while the rest of elements are intact. 30 data sets are generated for such scenarios when a local damage is introduced in a specific element. With 70 elements in the finite element model, 2100 single damage cases are simulated. In multiple element damage cases, the stiffness parameters of randomly selected two or three elements out of 70 elements are changed with stiffness reductions randomly defined between 0 and 30%, while the other elements are undamaged. 10,300 multiple damage cases with different damaged elements and patterns are simulated in total. The first seven frequencies and the corresponding mode shapes at 14 beam-column joints are taken as the input, and the pre-defined elemental stiffness reduction parameters as considered as the labelled output. These input and output data are used for the training and

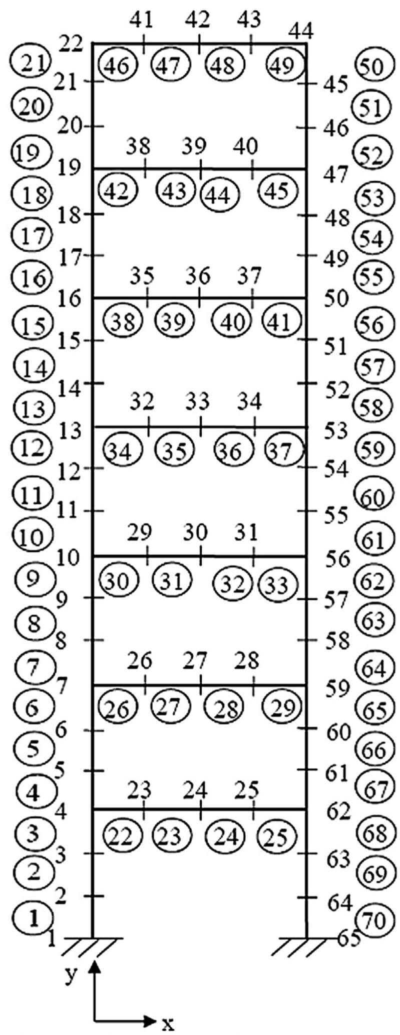

Fig. 4. Finite element model of the steel frame structure.

Table 1
Performance evaluation results for Scenario 1 in the numerical study.

Methods	MSE	R-Value	Optimization method	Training time (Hours)
ANN	6.2e − 04	0.652	SGD	5
ANN	4.1e − 04	0.824	SCG	1
The proposed approach	2.5e − 04	0.921	SCG	1.5

validation of the proposed framework.

To investigate the effectiveness and robustness of using the proposed framework for structural damage identification, the measurement noise and the uncertainty effect in the finite element modelling are included in the datasets. The following scenarios are defined in the numerical studies

(1) Scenario 1: No measurement noise and uncertainty. No noise effect in the vibration characteristics and uncertainties in the finite element modelling are considered;
(2) Scenario 2: Measurement noise effect. White noises are added on the input vectors, specifically, 1% noise in the frequencies and 5% in the mode shapes, considering structural frequencies are usually measured more accurately than mode shapes [33];
(3) Scenario 3: Uncertainty effect. 1% uncertainty is included in the elemental stiffness parameters to simulate the finite element modelling errors;
(4) Scenario 4: Both the measurement noise and uncertainty effect defined in Scenarios 2 and 3 are considered.

Since frequencies and mode shapes of the input feature $\bar{c}^r$ are measured in different scales, they are normalized separately to the range from -1 to $+1$. This range is chosen due to the active range of the hyperbolic tangent function. Considering that structural damages are usually observed at a few number of elements, sparse output vector is defined by using 0 for the intact state and 1 for the fully damage state. The output is also scaled to the range from -1 to $+1$ to serve the operating range of the used linear activation function in the final output layer. The performance evaluation of the proposed framework based on the pre-processed datasets will be described in the following section.

3.3. Performance evaluation of the proposed framework

Four different scenarios, as described in Section 3.2, are considered in the performance evaluation of the proposed framework against the traditional ANN. It should be noted that when comparing the performances of using the proposed approach and ANN for structural damage identification, the same datasets are used. For this numerical study, 2 hidden layers are used in the dimensionality reduction component with 100 neurons each, and one hidden layer with 80 neurons is used in the relationship learning component. Thus a deep neural network with 3 layers in total is used in this numerical study. The number of entries in the original input vector is 7 frequencies plus 14×7 mode shape functions, that is, 105 in total. 70 elemental stiffness parameters are included in the final output vector. The selection of the number of hidden layers and neurons is based on the complexity of the target problem. A deeper neural network would be used for a more complex problem. It is not necessary to use the same number of input nodes in each layer since it is a question of mapping the problem complexity to model complexity. A over complex model that has higher complexity than the data does not mean that it will always over fit the data provided that we have enough data and strong regularization techniques in place. Also as explained above, pre-training helps to regularize the network and place the weights in close to optimal regions at the same time. Usually the guideline is, when a set of data is given, data augmentation methods can be used to expand the dataset in a meaningful way. Then it is advised to start from the simplest model and step forward while increasing the complexity of the model gradually. It is also important to utilize regularization techniques while increasing the number of nodes and layers to avoid the model from being over fitted on data. Hyperbolic tangent and linear functions are employed respectively as the encoder and decoder functions of the autoencoders in the pre-training stage. After the pre-training, hyperbolic tangent activation functions are used for all the hidden layers while the last layer uses a linear function to predict the elemental stiffness reductions accurately.

It is noted that the selection of the optimal class of ANN models for a given set of training data has been studied [34,35]. In this paper, in order to perform a fair comparison between the proposed approach and ANN methods, 3 hidden layers with the same number of neurons in each layer is defined for an ANN model. Hence in contrast to the proposed approach no specific layer wise pre-training is performed on ANN model. Two commonly used optimization methods, namely, Stochastic Gradient Descent (SGD) and Scaled Conjugate Gradient (SCG), are used for training the ANN model, respectively. Nearly all the deep learning systems are powered by one very important algorithm: stochastic gradient descent (SGD), which is an extension of the gradient descent algorithm [36]. A recurring problem in machine learning is that large training sets are necessary for good generalization, but large training sets are computationally more expensive. The cost function used by a

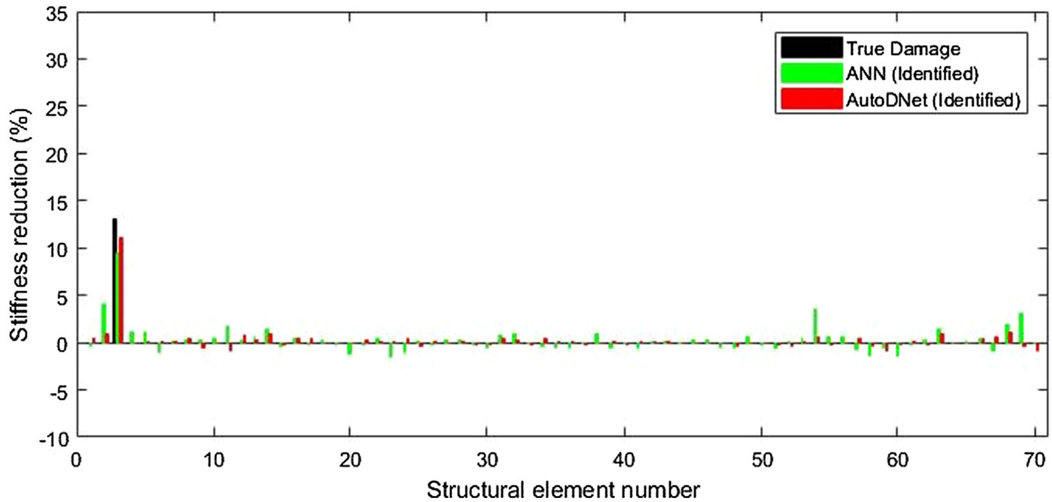

Fig. 5. Damage identification results of a single damage case from ANN and the proposed approach for Scenario 1.

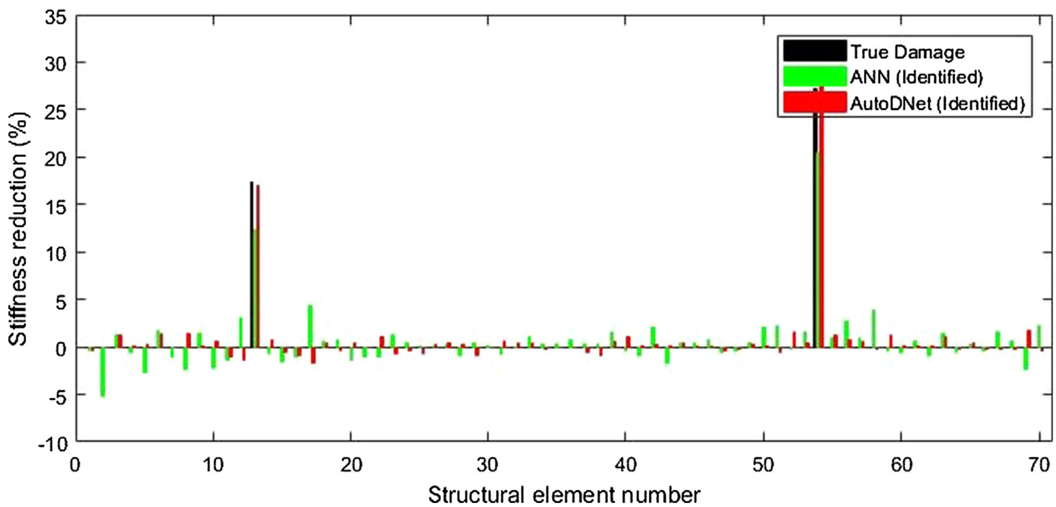

Fig. 6. Damage identification results of a multiple damage case from ANN and the proposed approach for Scenario 1.

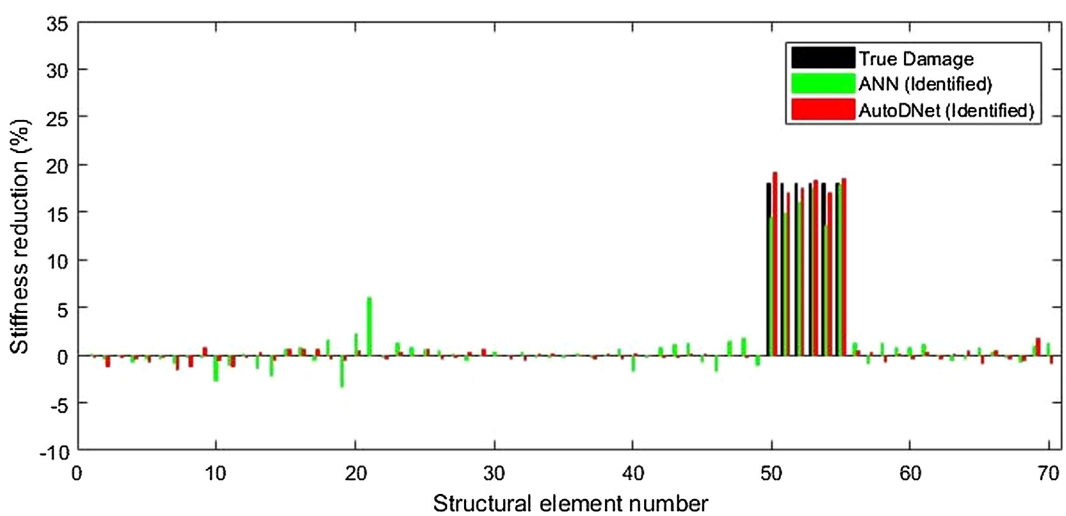

Fig. 7. Damage identification results of another multiple damage case from ANN and the proposed approach for Scenario 1.

machine learning algorithm often decomposes as a sum over training examples of some per-example loss function. As the training set size grows to billions of samples, the time to take a single gradient step becomes considerably long. The insight of SGD is that the gradient is an expectation and it may be approximately estimated using a small set of samples. Specifically, on each step of the algorithm, a mini batch of samples can be used, which is typically chosen to be a relatively small number of examples, ranging from 1 to a few hundred drawn uniformly

Table 2
Performance evaluation results for Scenario 2 in the numerical study.

Methods	MSE	R-Value	Optimization Method	Training Time (Hours)
ANN	6.7e−04	0.578	SGD	5
ANN	4.9e−04	0.711	SCG	1.5
The proposed approach	3.7e−04	0.794	SCG	2

from the training set. Hence it is plausible to fit a training set with billions of samples using updates computed on only a hundred samples. The estimate of the gradient is formed as using examples from the mini batch. The SGD algorithm then follows the estimated gradient downhill. Since SGD is a first order algorithm, it may suffer in efficiency and accuracy. In contrast Scaled Conjugate Gradient (SCG) is a supervised learning algorithm and can also be used for training the network based on conjugate directions [37]. Note that two vectors (u, v) are said to be conjugate in $u^T Av = 0$ and each step is moved in a direction conjugate to the all previous step. This direction is found from the residual and the director of the previous steps. SCG algorithm finds the search direction and the step size by using information of a second order Taylor expansion of the error function. It is fully-automated, includes no critical user- dependant parameters, and avoids a line search per iteration by using a Levenberg-Marquardt approach in order to scale the step size. Hence it yields a significant speed up in training while reaching the optimal in a short time comparatively to SGD algorithm.

70%, 15% and 15% samples randomly selected from the generated datasets are used for training, validation and testing, respectively. MSE and R-Value are used to assess the quality of the damage predictions through the networks. All the numerical computations are conducted on a desktop computer with an Intel i7 processor, 16 GB RAM and the graphics card NVidia 1080 Ti GTX by using GPU for parallel computing.

3.3.1. Scenario 1: No measurement noise and uncertainties

In this scenario, the datasets without measurement noise nor uncertainty effect are used. The performances of using ANN and the proposed framework are compared by examining the MSE values and R-Values on the test datasets.

The performance evaluation are shown in Table 1. It can be observed that ANN with SGD performs worse than the two other methods while consuming more time for training. It shows the inefficiency in utilizing the first order method for training. In contrast, ANN with SCG using the second order information for training the network consumes less time while performs better than SGD. A further problem with a backpropagation based training scheme such as SGD is that if more hidden layers are involved, it might become hard to achieve a satisfactory accuracy [17]. R-Values obtained from ANN with SGD and SCG are 0.652 and 0.824, respectively. The proposed approach shows a significant improvement in the regression compared with ANN methods with a smaller MSE value and a better R-Value. A little more training time is required for the proposed approach compared to ANN with SCG because pre-training is not used in the ANN model. To further demonstrate the quality of the damage identification, several single damage and multiple damage identification results are presented below. Since ANN with SCG generally performs better than ANN with SGD in terms of both the accuracy and the training efficiency, the proposed approach will only be compared with ANN with SCG in such intuitive comparisons.

The damage identification of a single damage case resulted from ANN with SCG and the proposed approach are shown in Fig. 5. It can be seen that the proposed approach provides more accurate damage identification than ANN. The damage location is well identified, and the identified stiffness reduction at the damaged element is very close to the actual value. The identified stiffness reduction values at the non -damage elements are very close to zero. The proposed approach is also evaluated against ANN (SCG) with multiple structural damage cases, and the identification results for two types of multiple damage cases are shown in Figs. 6 and 7. It can be seen clearly that the proposed approach work very well in multiple damage cases too. Damage locations are accurately detected and the identified stiffness reductions are very close to the actual values with very small false identifications. In contrary, ANN is not working very well in identifying multiple damages. Significant errors appear in the predicted damage extents, since there are no well-defined layer wise objectives for ANN model.

3.3.2. Scenario 2: Measurement noise effect

In this scenario, measurement noise is added into the data with 1% random noise in the frequencies and 5% in modes shapes. The accuracies of ANN and the proposed approach for damage identification in such scenario are investigated. The noisy data are used for the training, validation and testing of the neural networks. The performance evaluation of the proposed approach against the ANN model is shown in Table 2. A lower MSE value and a significantly higher R-Value from the proposed approach are observed than those from ANN models. It demonstrates the robustness and effectiveness of the proposed approach when the noise effect is included in the measurements. The damage identification results for two multiple damage cases are shown in Figs. 8 and 9. It can be seen that ANN with SCG may fail to identify the damages effectively, while the proposed approach can reliably and

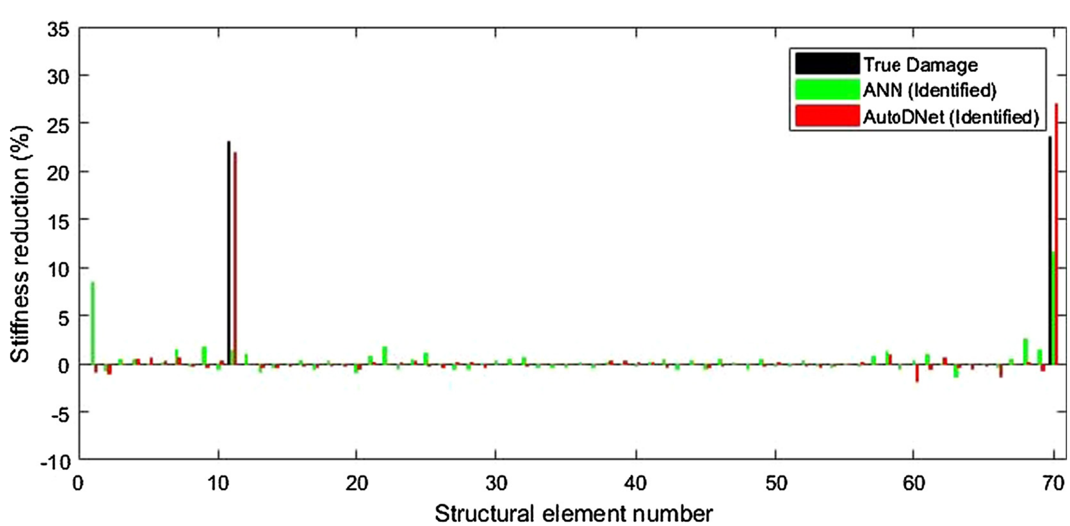

Fig. 8. Damage identification results of a multiple damage case from ANN and the proposed approach for Scenario 2.

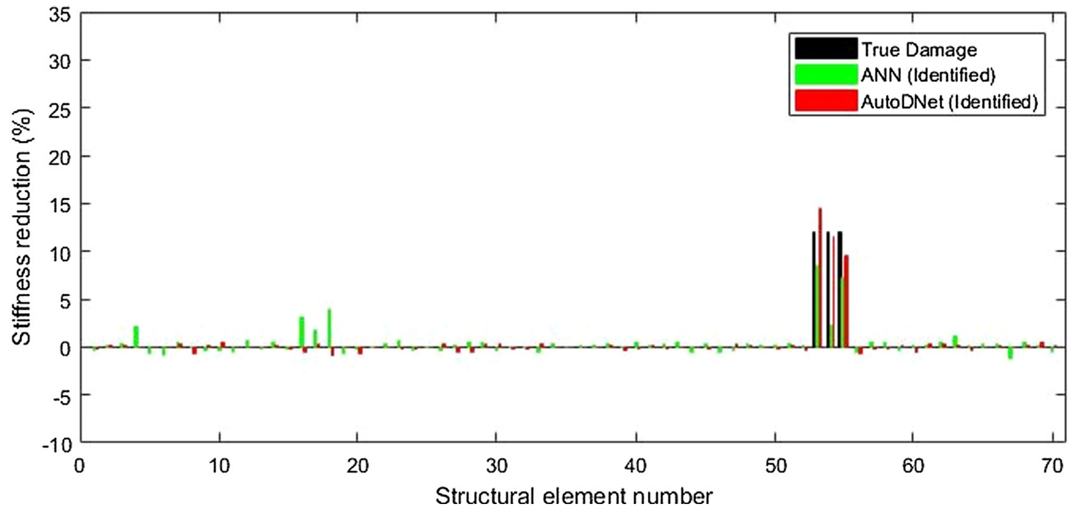

Fig. 9. Damage identification results of another multiple damage case from ANN and the proposed approach for Scenario 2.

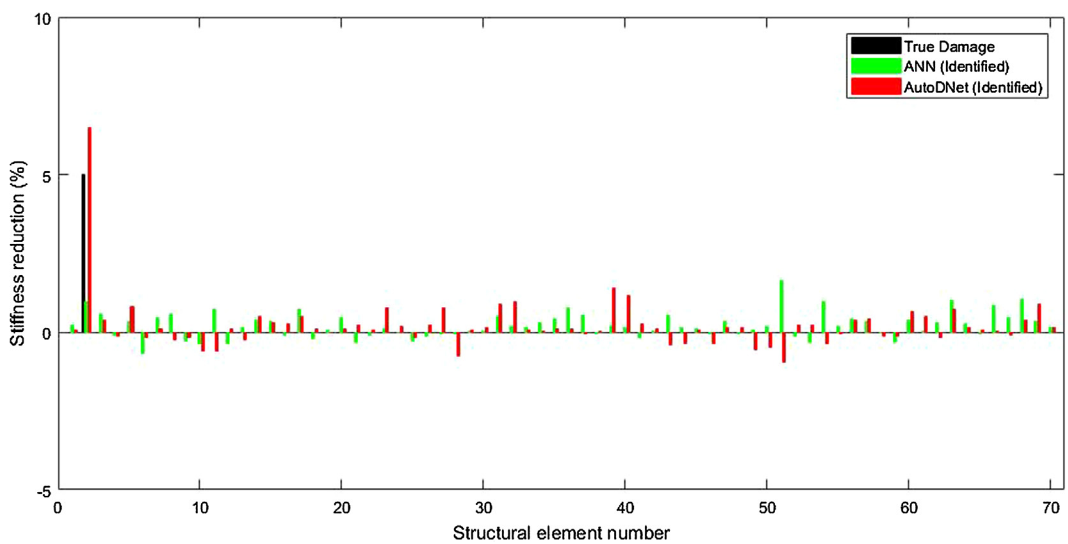

Fig. 10. Damage identification results of a minor damage case from ANN and the proposed approach for Scenario 2 with a higher noise level.

Table 3
Performance evaluation results for Scenario 3 in the numerical study.

Methods	MSE	R-Value	Optimization method	Training time (Hours)
ANN	6.5e−04	0.613	SGD	5
ANN	4.8e−04	0.729	SCG	2
The proposed approach	2.9e−04	0.83	SCG	2.5

accurately identify both the locations and magnitudes of structural damages when measurement noise is included in the data. the proposed approach outperforms ANN due to the utilizations of dimensionality reduction and relationship learning in the framework. A lower MSE value and a higher R-Value from the proposed approach are observed than those from ANN models. The damage identification results of two specific multiple damage cases are shown in Figs. 8 and 9. It can be found out from the identification results that ANN with SCG may fail to identify the damage effectively. However, the proposed approach can still reliably identify both the locations and magnitudes of pre-defined multiple structural damages when the noise is included in the data.

To investigate the effect of noise levels on the identification results, a higher level noise case, namely 2% in the frequencies and 10% in the mode shapes, is further considered. The identification results of a minor damage case with 5% stiffness reduction a single element are shown in Fig. 10. With this significant noise effect, the results demonstrate that ANN fails to identify the introduced damage, however, the proposed approach can still provide satisfactory identification results, though the pre-set damage severity is as small as 5%.

3.3.3. Scenario 3: uncertainty effect

Uncertainties inevitably exist in the process of structural damage identification, e.g., in the material properties of the finite element modelling, which will affect the performance of damage identification algorithms. In this scenario, 1% uncertainty is included in the elemental stiffness parameters to simulate the finite element modelling errors. It should be noted that the error in the model class selection is not considered in this study. Datasets are generated with this random modelling errors included in the finite element analysis, and used for the training, validation and testing of the neural networks. The performance evaluation results are shown in Table 3. The proposed approach outperforms ANN with an improvement as indicated by both the MSE and R-Values. The results from both ANN models are affected significantly by the uncertainty effect, as reflected by the corresponding R-values, indicating the accuracy of output prediction is lower.

Damage identification results of a single and a multiple damage

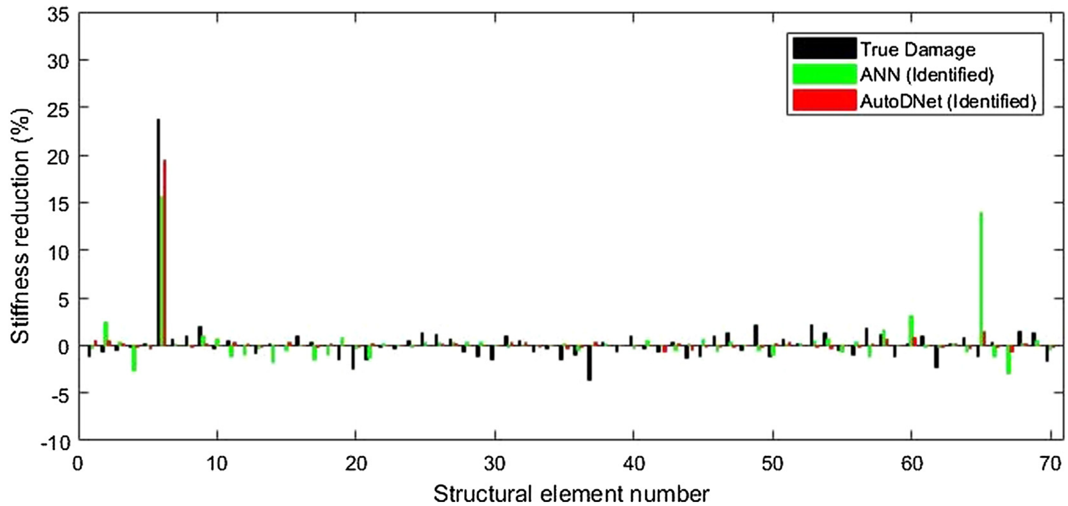

Fig. 11. Damage identification results of a single damage case from ANN and the proposed approach for Scenario 3.

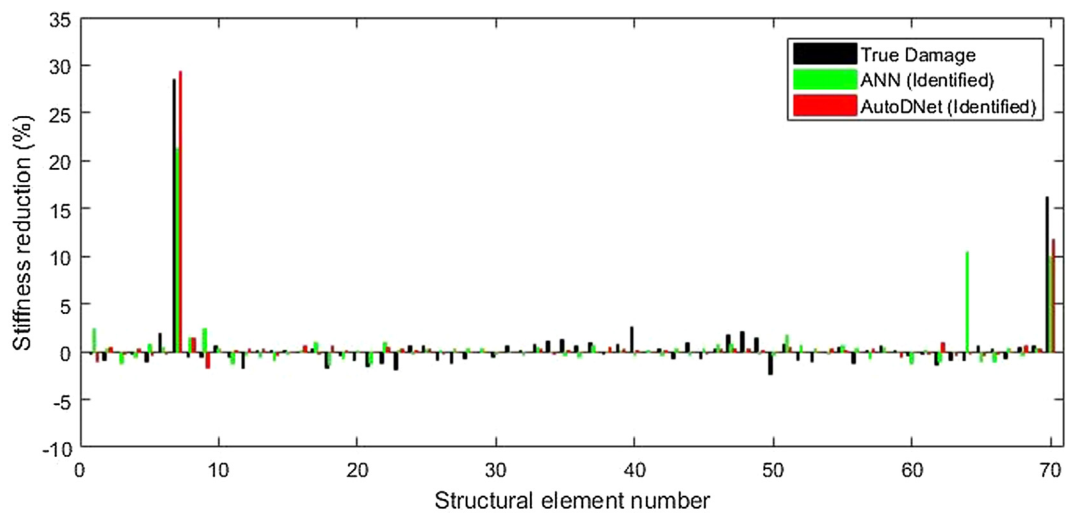

Fig. 12. Damage identification results of a multiple damage case from ANN and the proposed approach for Scenario 3.

Table 4
Performance evaluation results for Scenario 4 in the numerical study.

Methods	MSE	R-Value	Optimization method	Training time (Hours)
ANN	7.3e−04	0.536	SGD	5
ANN	5.2e−04	0.693	SCG	1.5
The proposed approach	3.6e−04	0.732	SCG	2

cases randomly selected from the testing datasets are shown in Figs. 11 and 12, respectively. For the single damage case, the damage location is well identified by using the proposed approach while a significant false positive identification is observed in the result from ANN (SCG), as shown in Fig. 11. The performance of ANN (SCG) is clearly affected by the uncertainty effect significantly, while the proposed approach is robust for such effect. For the multiple damage case shown in Fig. 12, the proposed approach detects damage locations accurately and the identified stiffness reductions are also close to the actual values with very minor false identifications due to the uncertainties. However, ANN (SCG) is not able to produce good detection results in both the locations and severities of damages. A significant false identification is observed in the results from ANN (SCG). By comparing these identification results, the accuracy and robustness of using the proposed approach for structural damage identification with uncertainty effect are demonstrated.

3.3.4. Scenario 4: both measurement noise and uncertainty effect

Both the measurement noise and uncertainty effect defined in Scenarios 2 and 3 are considered in this Scenario. It is very challenging to achieve an effective and reliable structural damage identification when significant measurement noise and uncertainty effect are involved. These uncertainties generally affect the damage detection results greatly. The performance evaluation results for this scenario are shown in Table 4. To demonstrate the effectiveness of the proposed approach, damage identification results from a single and a multiple damage cases in the testing datasets are shown in Figs. 13 and 14, respectively.

As observed in Table 4, the proposed approach once again outperforms the ANN models when both the measurement noise and uncertainty effect are present, evidenced by a higher R-Value and a lower MSE value. The L2-weight decay constraint applied on the cost function formulation ensures that it has less space to over-fit the training data. It should be noticed that a case with a minor damage, i.e. 5%, is selected and shown in Fig. 13. It can be observed that ANN completely fails to identify the single structural damage, while the proposed approach successfully identifies the damage in both the location and severity. With the multiple damage case as shown in Fig. 14, the proposed

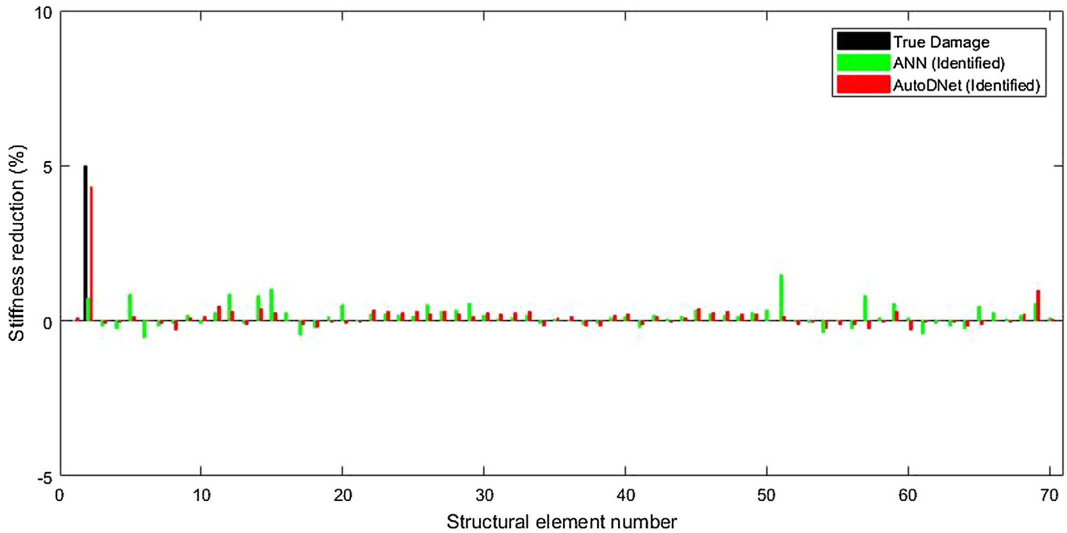

Fig. 13. Damage identification results of a minor damage case from ANN and the proposed approach for Scenario 4.

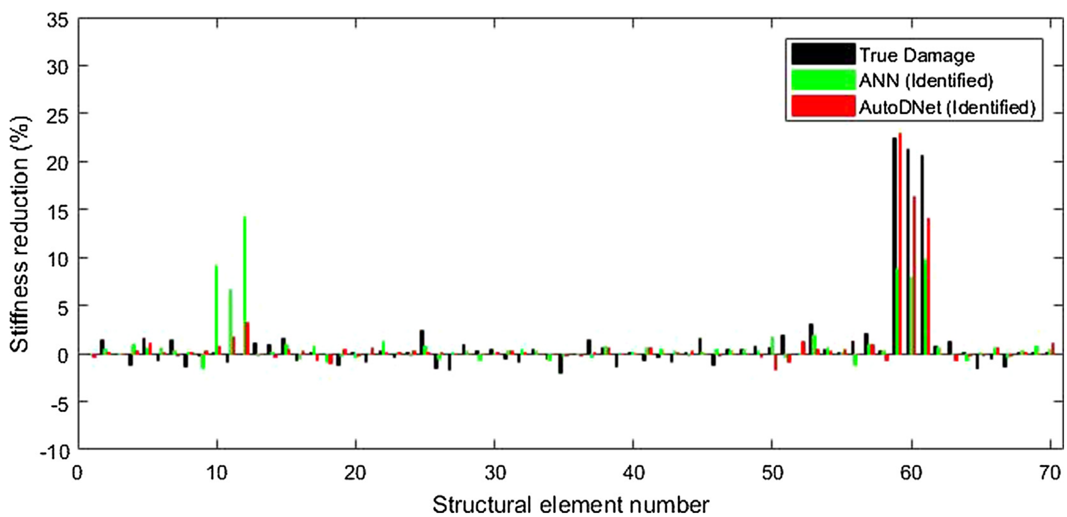

Fig. 14. Damage identification results of a multiple damage case from ANN and the proposed approach for Scenario 4.

approach also gives much more accurate stiffness reduction predictions than ANN (SCG) in terms of both the locations and severities, with ANN (SCG) producing several cases of false positives.

Damage identification results from the four scenarios demonstrate clearly the accuracy and robustness of using the proposed approach in structural damage identification, compared with the traditional ANN (SCG), even when the measurement noise and uncertainty effect are considered.

4. Experimental verifications

Experimental verifications of using the proposed approach for damage identification in a laboratory steel frame model are presented in this section. The experimental setup, network design and training, and damage identification results will be presented in details.

4.1. Experimental model and initial model updating

An eight-story shear-type steel frame model is fabricated in the laboratory for experimental validations of the proposed approach. Fig. 15 shows the testing steel frame model in the laboratory. The height and width of the frame structure are 2000 mm and 600 mm, respectively. Thick steel bars of with dimension of 100 mm × 25 mm are used as the floors of the frame model, and two flat bars of the same cross section with a width of 50 mm and a thickness of 5 mm are used as columns. The beams and columns are welded to form rigid beam-column joints. The bottom of the two columns is welded onto a thick and solid steel plate, which is fixed to a strong floor. The initial elastic modulus of the steel is estimated as 200 GPa, and the mass density 7850 kg/m^3.

Dynamic tests are conducted to identify the vibration characteristics of the testing frame model. A modal hammer with a rubber tip is used to apply the excitation on the model. Accelerometers are installed at all the floors to measure horizontal acceleration responses under the hammer impact. The sampling rate is set as 1024 Hz, and the cut-off frequency range for the band-pass filter is defined from 1 Hz to 100 Hz for all tests. An initial shear-type finite element model with 8 lump masses is built based on the dimensions and material properties of the frame. Vibration testing data from the experimental model under the healthy state are used to perform an initial model updating to minimize the difference between the measured and analytical vibration characteristics, i.e. frequencies and mode shapes. The First-order sensitivity based method is employed for the updating [38,39]. Environmental noise and uncertainties are inevitable in such kind of settings. The detailed experimental test setup and model updating procedure are referred to Ref. [40]. The measured and analytcial natural frequencies of the experimental model before and after model updating are listed in

Fig. 15. A steel frame model in the laboratory.

Table 5
Measured and analytical natural frequencies of the experimental model before and after updating.

Mode	Measured	Before updating		After updating	
		Analytical (Hz)	Error (%)	Analytical (Hz)	Error (%)
1	4.645	4.810	3.55	4.636	0.19
2	13.705	14.267	4.10	13.714	0.06
3	22.554	23.238	3.03	22.558	0.02
4	30.695	31.418	2.36	30.776	0.26
5	38.241	38.528	0.75	38.225	0.04
6	44.434	44.325	0.25	44.422	0.03
7	48.826	48.614	0.43	48.712	0.23
8	52.306	51.246	2.03	52.161	0.28

Table 5. The maximum error in the frequencies after updating is only 0.28% at the eighth mode, indicating a very good agreement. The measured and analytical mode shapes of the model are shown in Fig. 16. The mode shapes after model updating match very well with the measured mode shapes from the vibration tests. This well updated finite element model is achieved to serve as the baseline model in the following studies for generating the training data and validating the performance of the proposed framework in structural damage identification. The following sections will present the data generation process based on the baseline finite element model for network training and validation, the architecture design of the Autoencoder based framework and ANN, and the investigation of using the vibration characteristics from the damaged laboratory model for damage identification with the proposed approach and ANN. Results from ANN and the proposed approach will be compared to demonstrate the performance for a reliable structural damage identification with experimental testing measurements.

4.2. Training data generation

Modal analysis is performed using the baseline model to generate the input and output data to train the networks. The first eight frequencies and the corresponding mode shapes from these eight floors are obtained based on the pre-defined structural stiffness parameters. Similar to Section 3.2, the elemental stiffness parameters are normalized to the range between 0 and 1, where 1 denotes the intact state and 0 denotes the completely damaged state. 25,440 datasets are generated from the baseline model that include all possibilities for single element and two element damage cases. In single element damage cases, the stiffness parameter for each element varies from 1, 0.99, 0.98, …, to 0.7 while keeping all other elements undamaged. 30 data sets are generated for the scenario when a local damage is introduced in a specific element. With 8 elements in the finite element model, 240 single element damage cases are defined. In multiple element damage cases, the stiffness parameters for two random elements vary from 1, 0.99, 0.98, …, to 0.7 while keeping the other elements undamaged. 25,200 multiple element damage cases are defined. An additional measurement noise is added into the data with 1% random noise in the frequencies and 5% in modes shapes in order to make the model robust to noisy measurements. Adding noise to the training data can improve the robustness and accuracy of using the proposed approach to deal with the real testing data. These datasets are processed with the same pre-processing procedure as described in Section 3.2, and then used for training and validation.

4.3. Network structure

A relatively simpler Autoencoder model is defined here considering the complexity of the target problem and the number of unknown parameters to be identified. One hidden layer ($k = 1$) with 36 neurons is designed in the dimensionality reduction component, and a hidden layer with 16 neurons is used in the relationship learning component. The input vector contains 8 frequencies and 8×8 mode shape values, that is, 72 values in total. 8 stiffness reduction parameters are involved in the final output vector. For the pre-training, hyperbolic tangent function is used as the encoder function and linear function is used as the decoder function in the autoencoder. Hyperbolic tangent function is used as the activation functions for all the layers. To have a fair comparison, the same number of hidden layers and neurons are used to form an ANN model and the same training datasets are used.

4.4. Damage identification results

Damages are introduced by reducing the column cross sections of the specific floors of the steel frame model. The flexural stiffness of each floor is proportional to the moment of inertia $bh^3/12$ of the column, where b and h are defined as the width and thickness of the column respectively. The equivalent stiffness reduction can be obtained based on the decrease of the moment of inertia. However, it should be noted that only the stiffness reduction is considered and the mass change is ignored since the structural damage is mainly related with the stiffness reduction. Two damage cases, namely, Case 1 and Case 2, are introduced in the structure. Only a single damage is defined in Case 1 with 20% reduction of the equivalent stiffness of the 2nd floor. Case 2 has multiple damages. Besides the damage in Case 1, another damage is introduced with 10% stiffness reduction in the 7th floor. The introduced damages in the 2nd and 7th floors are shown in Fig. 17. Experimental vibration tests are conducted with the damaged model to identify the structural vibration characteristics, i.e. frequencies and mode shapes, of the above two damage cases.

After training and validating the designed networks, frequencies and mode shapes from the above two damage cases with the additional added noises are used as the testing input to predict the structural damages and investigate the performance and robustness of using the

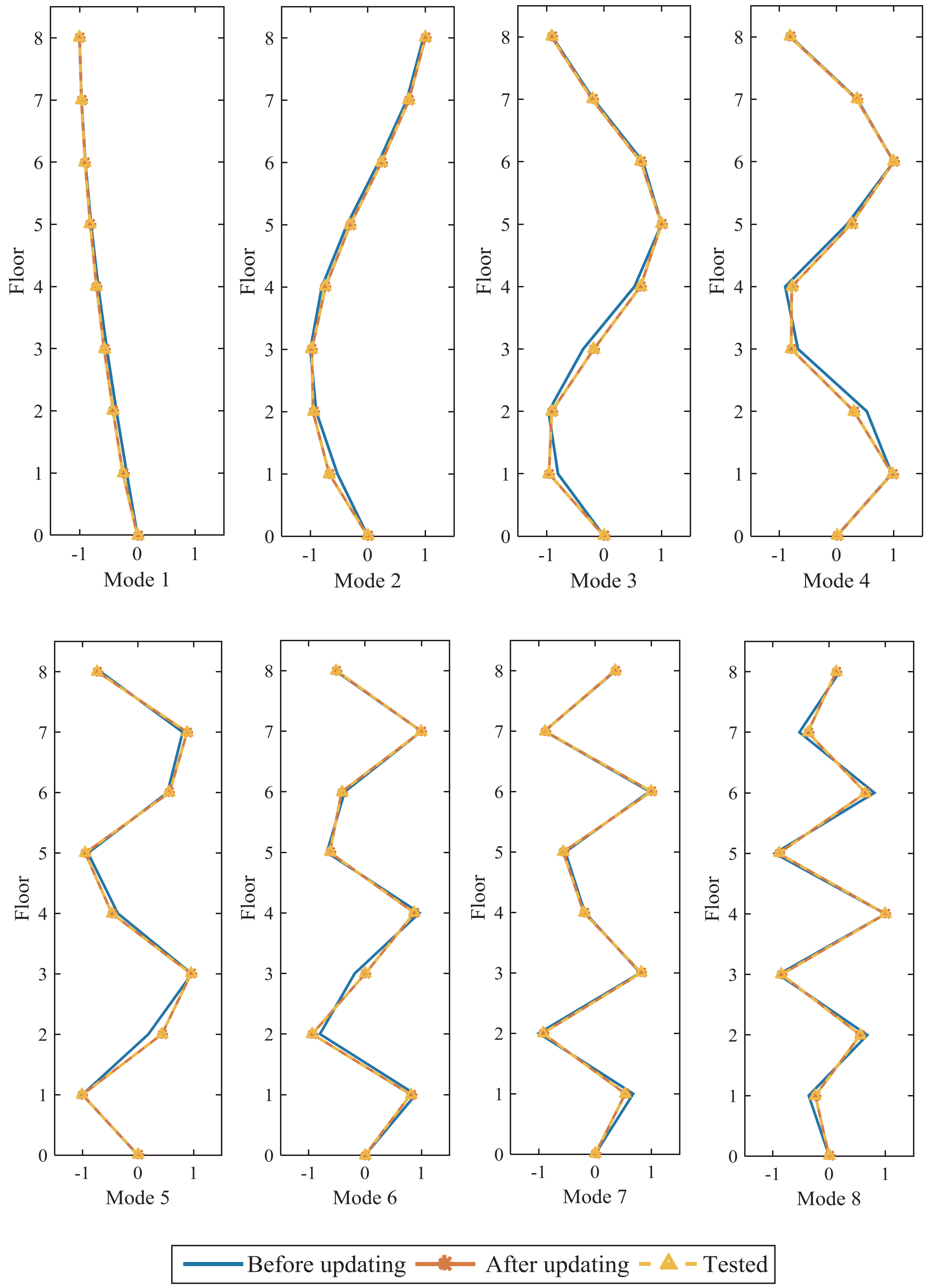

Fig. 16. Mode shapes before and after updating.

proposed approach with real testing measurements for structural damage identification. The performance evaluation results for these two test cases by using ANN and the proposed approach are shown in Table 6. It can be observed that the MSE value from the proposed approach is significantly smaller than those from the ANN methods. Besides, the regression from the proposed approach is also improved, as represented by the R-Value. ANN with SGD training method requires a much higher amount of training time, while same training time is required for both the proposed approach and ANN with SCG. Figs. 18 and 19 shows the identified structural damages of both damage Case 1 and Case 2. Comparing with the true introduced damages and results from ANN methods, it is demonstrated that the identified stiffness reductions using the proposed approach are very close to the exact values with less false identifications and smaller false values. This indicates that the proposed approach can well identify the pre-set structural damages in the laboratory model with experimental testing data including environmental noise and uncertainties.

5. Conclusion

An autoencoder based deep learning framework for structural damage identification is proposed in this paper. It can well perform the pattern recognition between the modal information, such as frequencies and mode shapes, and structural stiffness parameters. Two main components, that is, dimensionality reduction and relationship learning, are included in the proposed framework. The dimensionality reduction

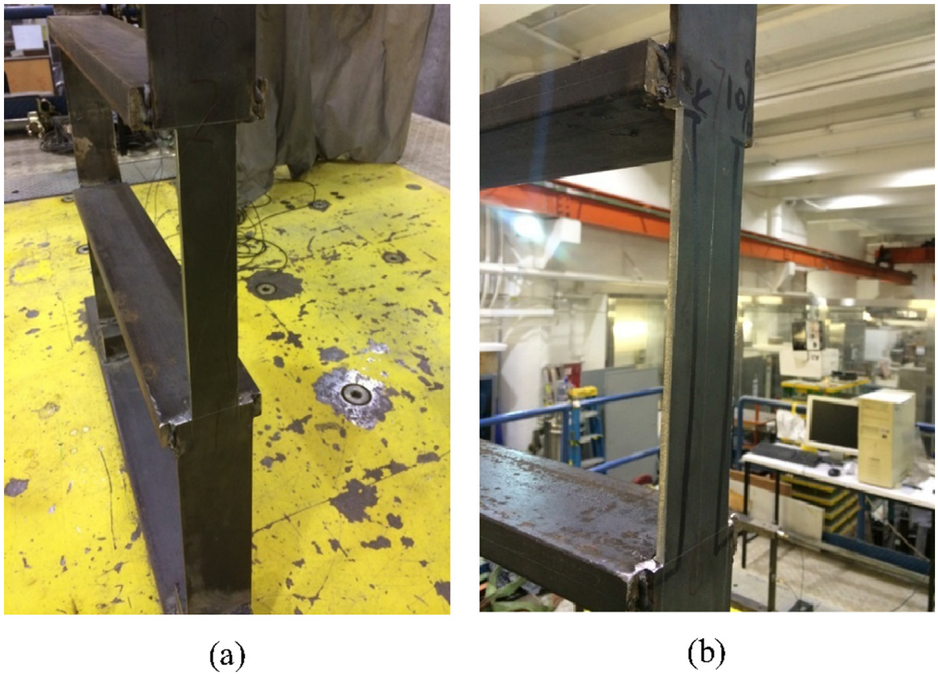

Fig. 17. Introduced damages of the frame model: (a) Introduced damage at the 2nd floor; (b) Introduced damage at the 7th floor.

Table 6
Performance evaluation results in the experimental study.

Methods	MSE	R-Value	Optimization method	Training time (Hours)
ANN	2.1e − 04	0.897	SGD	3
ANN	9.3e − 05	0.989	SCG	1
The proposed approach	9.1e − 06	0.996	SCG	1

component utilizes an autoencoder model to compress the original input vector to obtain a robust low dimensional feature vector that preserves the necessary information through multiple hidden layers. This not only effectively removes the redundancy in the data but also keep the most useful information to serve as the input to the relationship learning component. A regression model is defined in the relationship learning component to map the compressed feature vector to the output stiffness reduction parameters. The dimensionality reduction process of the proposed framework could be applied and extended to a deeper network architecture from a complex problem. L2-weight decay is utilized to enhance the overall training process by limiting the over-fitting tendency while training deep architectures. A layer-wise pre-training is performed to optimize the weights of the individual layers, and the whole network is fine-tuned using a joined optimization towards the final objective function. Numerical and experimental validations on steel frame structures are conducted and the results demonstrate the improved accuracy and efficiency of the proposed framework, comparing with the traditional ANN methods. More accurate structural damage identification results can be obtained in regards to both the locations and severities of the damages, even when both the measurement noise and uncertainty effect are considered. The proposed framework is capable of handling a large amount of training data. The layer-wise pre-training and fine-tuning are employed to improve the training efficiency and accuracy. It can also be used for more complex problems with a complicated network structure, for example, a high dimensional input data, multiple hidden layers and a large number of

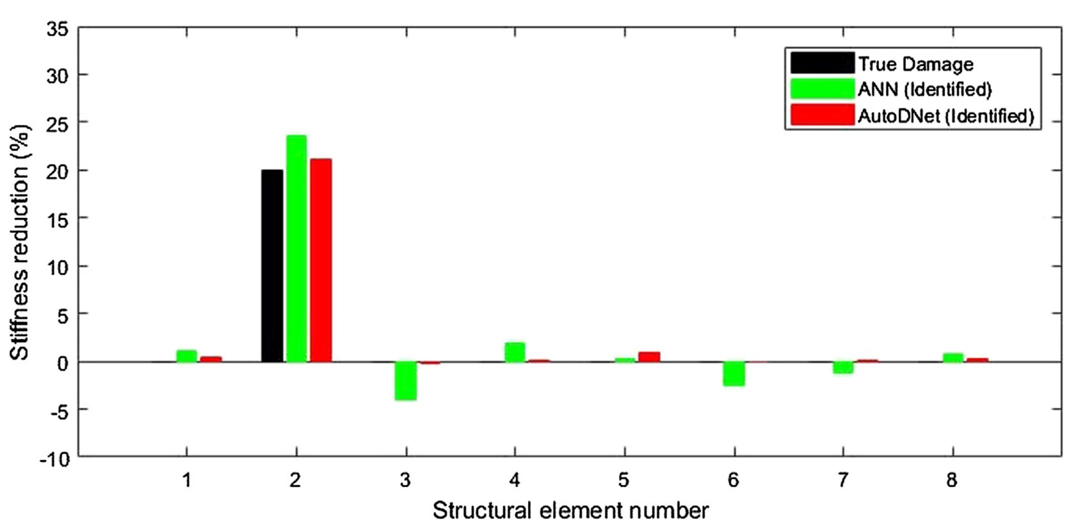

Fig. 18. Damage identification results of Case 1 from ANN and the proposed approach.

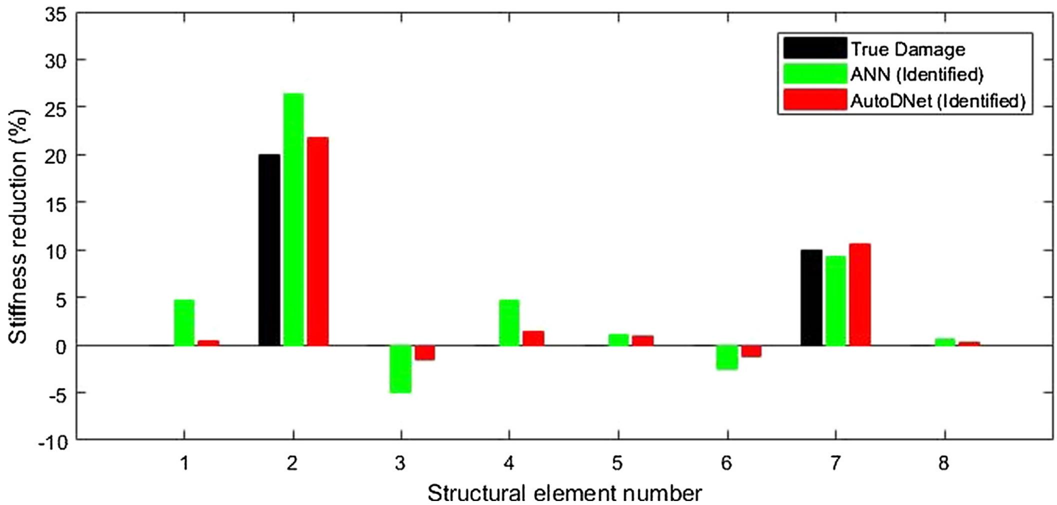

Fig. 19. Damage identification results of Case 2 from ANN and the proposed approach.

output parameters. The proposed framework will be extended to utilize other structural vibration characteristics, e.g., flexibility and frequency response function, etc., as the input, in order to increase the sensitivity of the network and improve the performance of structural health monitoring and damage identification for detecting minor damages under various uncertainties and noise effect.

Acknowledgments

The work described in this paper was supported by an Australian Research Council project.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.engstruct.2018.05.109.

References

[1] Brownjohn JMW. Structural health monitoring of civil infrastructure. Philosoph Transact Royal Soc London A: Mathemat, Phys Eng Sci 2007;365(1851):589–622.
[2] Li J, Hao H. A review of recent research advances on structural health monitoring in Western Australia. Struct Monitor Maintenance 2016;3(1):33–49.
[3] Padil KH, Bakhary N, Hao H. The use of a non-probabilistic artificial neural network to consider uncertainties in vibration-based-damage detection. Mech Syst Sig Process 2017;83:194–209.
[4] Hao H, Xia Y. Vibration-based damage detection of structures by genetic algorithm. J Comput Civil Eng ASCE 2002;16(3):222–9.
[5] Ding ZH, Huang M, Lu ZR. Structural damage detection using artificial bee colony algorithm with hybrid search strategy. Swarm Evol Comput 2016;28:1–13.
[6] Ding ZH, Yao RZ, Huang JL, Huang M, Lu ZR. Structural damage detection based on residual force vector and imperialist competitive algorithm. Struct Eng Mech 2017;62(6):709–17.
[7] Yun CB, Yi JH, Bahng EY. Joint damage assessment of framed structures using a neural networks technique. Eng Struct 2001;23(5):425–35.
[8] Lee JJ, Lee JW, Yi JH, Yun CB, Jung HY. Neural networks-based damage detection for bridges considering errors in baseline finite element models. J Sound Vib 2005;280(3):555–78.
[9] Zang C, Imregun M. Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection. J Sound Vib 2001;242(5):813–27.
[10] Ni YQ, Wang BS, Ko JM. Constructing input vectors to neural networks for structural damage identification. Smart Mater Struct 2002;11(6):825.
[11] Yeung WT, Smith JW. Damage detection in bridges using neural networks for pattern recognition of vibration signatures. Eng Struct 2005;27(5):685–98.
[12] Bakhary N, Hao H, Deeks AJ. Damage detection using artificial neural network with consideration of uncertainties. Eng Struct 2007;29(11):2806–15.
[13] Li J, Dackermann U, Xu YL, Samali B. Damage identification in civil engineering structures utilizing PCA-compressed residual frequency response functions and neural network ensembles. Struct Cont Health Monitor 2011;18(2):207–26.
[14] Bandara RP, Chan TH, Thambiratnam DP. Frequency response function based damage identification using principal component analysis and pattern recognition technique. Eng Struct 2014;66:116–28.
[15] Dackermann U, Smith WA, Randall RB. Damage identification based on response-only measurements using cepstrum analysis and artificial neural networks. Struct Health Monitor 2014;13(4):430–44.
[16] Hochreiter S, Bengio Y, Frasconi P, and Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, A Field Guide to Dynamical Recurrent Neural Networks. S.C. Kremer and J.F. Kolen, Eds. Wiley-IEEE Press. 2001:1–15.
[17] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313:504–7.
[18] Arel I, Rose DC, Karnowski TP. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. Computat Intelligence Magazine IEEE 2010;5(4):13–8.
[19] Schmidhuber J. Deep learning in neural networks: an overview. Neural Net 2015;61:85–117.
[20] Jia F, Lei Y, Lin J, Zhou X, Lu N. Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. Mech Syst Sig Process 2016;72:303–15.
[21] Gan M, Wang C. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. Mech Syst Sig Process 2016;72:92–104.
[22] Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. J Sound Vib 2017;388:154–70.
[23] Cha YJ, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civ Infrastruct Eng 2017;32(5):361–78.
[24] Pathirage CSN, Li J, Li L, Hao H, and Liu W. Deep autoencoder models for pattern recognition in civil structural health monitoring. World Congress on Engineering Asset Management (WCEAM 2016). Jiuzhaigou, Sichuan, China 2016.
[25] Bengio Y, and Lecun Y. Scaling learning algorithms towards AI. Bottou L, Chapelle O, DeCoste D, Weston J, Eds, Large-scale kernel machines. MIT Press. 2007.
[26] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.
[27] Kan M, Shan S, Chang H, and Chen X. Stacked progressive autoautoencoders (SPAE) for face recognition across poses. in 2014 IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2014:1883–1890.
[28] Pathirage CSN, Li L, Liu W, and Zhang M. Stacked face de-noising auto encoders for expression-robust face recognition. in: in 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), IEEE, 2015:1–8.
[29] Bengio Y. Learning deep architectures for AI. Foundations Trends® Mach Learn 2009;2(1):1–127.
[30] Bengio Y, Lamblin P, Popovici D, and Larochelle H. Greedy layer-wise training of deep networks. NIPS'06 Proceedings of the 19th International Conference on Neural Information Processing Systems. 2006:153–160.
[31] Bengio Y. Practical recommendations for gradient-based training of deep architectures, in Neural Networks: Tricks of the Trade, Springer, 2012:437–478.
[32] Li J, Hao H. Substructure damage identification based on wavelet-domain response reconstruction. Struct Health Monitor 2014;13(4):389–405.
[33] Xia Y, Hao H, Brownjohn JMW, Xia PQ. Damage identification of structures with uncertain frequency and mode shape data. Earthquake Eng Struct Dyn 2002;31(5):1053–66.
[34] Yuen K-V, Lam H-F. On the complexity of artificial neural networks for smart structures monitoring. Eng Struct 2006;28(7):977–84.
[35] Lam H-F, Yuen K-V, Beck JL. Structural health monitoring via measured Ritz vectors utilizing artificial neural networks. Comput-Aided Civ Infrastruct Eng 2006;21(4):232–41.
[36] Bottou L. Stochastic gradient learning in neural networks. Proceedings Neuro-Nımes 1991;91:1–12.

[37] Møller MF. A scaled conjugate gradient algorithm for fast supervised learning. Neural Net 1993;6(4):525–33.
[38] Friswell M, and Mottershead JE. Finite element model updating structural dynamics, Springer Science & Business Media, 1995.
[39] Lu ZR, Wang L. An enhanced response sensitivity approach for structural damage identification: convergence and performance. Int J Numer Meth Eng 2017;111:1231–51.
[40] Ni P, Xia Y, Li J, Hao H. Improved decentralized structural identification with output-only measurements. Measurement 2018;122:597–610.