Special Issue Article

SHM Structural Health Monitoring

Structural Health Monitoring
2019, Vol. 18(1) 103–122
© The Author(s) 2018
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1475921718800363
journals.sagepub.com/home/shm
SAGE

Development and application of a deep learning–based sparse autoencoder framework for structural damage identification

Chathurdara Sri Nadith Pathirage[1], Jun Li[2,3], Ling Li[1], Hong Hao[2,3], Wanquan Liu[1] and Ruhua Wang[1]

Abstract

This article proposes a deep sparse autoencoder framework for structural damage identification. This framework can be employed to obtain the optimal solutions for some pattern recognition problems with highly nonlinear nature, such as learning a mapping between the vibration characteristics and structural damage. Three main components are defined in the proposed framework, namely, the pre-processing component with a data whitening process, the sparse dimensionality reduction component where the dimensionality of the original input vector is reduced while preserving the required necessary information, and the relationship learning component where the mapping between the compressed dimensional feature and the stiffness reduction parameters of the structure is built. The proposed framework utilizes the sparse autoencoders based deep neural network structure to enhance the capability and performance of the dimensionality reduction and relationship learning components with a pre-training scheme. In the final stage of training, both components are jointly optimized to fine-tune the network towards achieving a better accuracy in structural damage identification. Since structural damages usually occur only at a small number of elements that exhibit stiffness reduction out of the large total number of elements in the entire structure, sparse regularization is adopted in this framework. Numerical studies on a steel frame structure are conducted to investigate the accuracy and robustness of the proposed framework in structural damage identification, taking into consideration the effects of noise in the measurement data and uncertainties in the finite element modelling. Experimental studies on a prestressed concrete bridge in the laboratory are conducted to further validate the performance of using the proposed framework for structural damage identification.

Keywords

Deep learning, neural networks, sparse autoencoders, structural damage identification, pre-training

Introduction

A standard artificial neural network (ANN)[1] consists of many simple and connected processors called neurons, each producing a sequence of real-valued activations. Input neurons get activated through sensors perceiving the environment, and hidden neurons get activated through weighted connections from previously active neurons. Learning is about finding weights that make the neural network exhibit the desired behaviour, such as a prediction of certain set of elements.[2] Depending on the problem and how the neurons are connected, such behaviour may require long causal chains of computational stages, where each stage performs nonlinear transformation of the aggregate activation of the network. Backpropagation (BP) based on gradient descent method is one of the most popular shallow learning

[1]School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley, WA, Australia
[2]Centre for Infrastructural Monitoring and Protection, School of Civil and Mechanical Engineering, Curtin University, Bentley, WA, Australia
[3]School of Civil Engineering, Guangzhou University, Guangzhou, China

Corresponding author:
Jun Li, Centre for Infrastructural Monitoring and Protection, School of Civil and Mechanical Engineering, Curtin University, Kent Street, Bentley, WA 6102, Australia.
Email: junli@curtin.edu.au; LI.Jun@connect.polyu.hk

algorithms used for training the neural network. The fundamental problem in utilizing BP-based training in deep neural networks[2,3] is the problem of vanishing or exploding gradient that often occurs when backpropagating the gradient to the first layers of the network. With standard activation functions, cumulative backpropagated error signals either shrink rapidly or grow out of bounds. In fact, they decay exponentially in the number of layers or they explode. This is also known as the long-time lag problem. Later work by Bengio et al.[4] also studied basins of attraction and their stability under noise from a dynamical systems point of view: either the dynamics are not robust to noise or the gradients vanish.

Over the years, several ways to partially overcome this fundamental problem of training a deep neural network are explored. Due to the increasing availability of big data sets and powerful computers, more research works are using deep learning methodologies for training deep architectures. Hinton and Salakhutdinov[5] have reported that the neural networks structured as autoencoders can model nonlinear interactions and scale well to large data sets. They can also be trained with unlabeled data, which in the case of high-content screening is plentiful. It was shown that autoencoders could be composed to create an efficient and flexible dimensionality reduction algorithm.[6] The idea of composing simpler models in layers to form more complex ones has been successful with a variety of basis models, for example, stacked de-noising autoencoders,[5] Ranzato et al.'s model[7] and Vincent et al.'s model.[8] These models were frequently employed for unsupervised pre-training. A layer-wise scheme is used for initializing the parameters of a multi-layer perceptron, which is subsequently trained by minimizing an appropriate loss function over real versus model-predicted labels of the data. In the works by Srivastava[9] and Goodfellow et al.,[10] the pre-training scheme was overlooked in favour of randomly initialized models coupled with new powerful regularization techniques. The original applications mainly focused on face detection, object recognition, speech recognition and detection and natural language processing.[2,11] Recently, it has been developed for fault detection and diagnosis in mechanical engineering[12] and data anomaly detection in civil engineering.[13] Lin et al.[14] proposed a structural damage detection approach to automatically perform feature extraction through deep learning. Convolutional neural network architecture was used in this study. Cha et al.[15] proposed a vision-based method using a deep architecture of convolutional neural networks for detecting concrete surface cracks without calculating the defect features.

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 (DAE) is utilized for effective feature learning through hierarchical nonlinear mappings via multiple hidden layers of the model.[8] These properties can be utilized for structural health monitoring problem. A DAE model that can provide optimal solutions for problems of highly nonlinear nature, such as learning a mapping between the input modal information and output structural stiffness parameters, has been proposed.[16] It is the first attempt in related literature that utilizes a DAE model on vibration-based structural damage detection/identification and the improved performance is demonstrated against the traditional ANN. In that work, 'tanh' activation functions are utilized associated with the DAE model to perform structural damage identification. This may lead to learning dense latent representations and will not be robust for data sets including measurement noise effect and uncertainty effect in the modelling. It is reported by Glorot et al.[17] that deep sparse autoencoder models achieve better robust features under noise effect compared with the non-sparse variant. With the reconstruction loss regularized by a sparsity-inducing term, a better de-noising performance is achieved. Although the existing methods show strength under some noises such as Gaussian noise, they would be fragile when the data sets contain large amounts of outliers. The reason is that most models are based on mean square error (MSE) criterion which would be sensitive to outliers.[18,19] Sparse representations have a number of theoretical and practical advantages, as demonstrated in a number of recent studies.[20–22] Studies of sparse coding have demonstrated that the sparseness seems to play a key role in learning useful features. In sparse methods, the code is forced to have only a few non-zero units, while most code units are zero most of the time. In particular, sparse coding methods have a good robustness to noise effect.

This article proposes a deep sparse autoencoder framework for structural damage identification, which can be utilized to learn optimal solutions for pattern recognition problems, such as learning a mapping between the vibration characteristics and structural damage. The proposed framework includes three components, namely, the pre-processing components, the dimensionality reduction component and the relationship learning component. The proposed framework utilizes deep sparse autoencoders to enhance the capacity and performance of dimensionality reduction and relationship learning components with a pre-training scheme. The fine-tuning of the whole network is conducted towards achieving a better accuracy in structural damage identification. Since structural damage usually occurs only at several locations that exhibit stiffness reduction at sparse elements out of the large total number of elements in the entire structure, a sparsity-inducing term is adopted in this framework. Numerical studies on a steel frame structure are conducted to investigate the accuracy and robustness of using the proposed framework for structural damage identification, particularly considering the effects of noise in the measurement data and uncertainties in the finite element modelling. Experimental studies on a prestressed concrete bridge in the laboratory are conducted to further validate the performance of the proposed framework.

The rest of this article is organized as follows. Section 'Deep sparse autoencoder framework' will describe the theoretical background and development of deep sparse autoencoder framework. The necessary background information on autoencoder and the development and application of deep sparse autoencoder framework for structural damage identification will be presented. Section 'Numerical studies' will numerically validate the accuracy and robustness of using the proposed framework for damage identification, considering the measurement noise and uncertainty effects. Experimental verifications will be given in section 'Experimental verification' to further demonstrate the performance of the proposed approach. Finally, concluding remarks will be provided in section 'Conclusion'.

Deep sparse autoencoder framework

In this section, the development of deep sparse autoencoder framework along with the training method will be described. A brief review of the traditional autoencoder will be presented in section 'Autoencoder', and the proposed framework will be described in detail in section 'Deep sparse autoencoder framework for structural damage identification'. The objective of the proposed framework is to perform structural damage identification, which is a pattern recognition problem based on the fact that changes in the physical material properties of structures, that is, stiffness, will alter the structural vibration characteristics, that is, natural frequencies and mode shapes. Therefore, the natural frequencies and mode shapes can be used as the input, while the elemental stiffness reduction parameters that represent the structural heath conditions can be used as the output. The training of the proposed framework built upon the deep neural networks with sparse autoencoders is conducted based on handling a massive amount of data samples. A pre-training scheme will be employed and then the fine-tuning will be conducted to optimize the whole network. The accuracy and efficiency of applying the proposed framework for structural damage identification are investigated and compared with some previous methods.

Autoencoder

A typical autoencoder-based neural network[23] is unsupervised, consisting of two parts, namely, encoder and decoder, with a single hidden layer.

Encoder. A deterministic mapping function $f(\bar{x})$, 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 nonlinearity, 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 affine mapping weight matrix, $\bar{b} \in \Re^r$ is the bias vector and Φ is the activation function, which is usually a squashing nonlinear function such as sigmoid function $\Phi(\bar{x}) = sigmoid(\bar{x}) = 1/1 + e^{-\bar{x}}$ and tangent function $\Phi(\bar{x}) = \tanh(\bar{x}) = (e^{\bar{x}} - e^{-\bar{x}})/(e^{\bar{x}} + e^{-\bar{x}})$. 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. A mapping function $g(\bar{h})$, which transforms the latent representation $\bar{h} \in \Re^r$ obtained from equation (1) 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, which can be expressed as follows

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

where $\hat{W} \in \Re^{dxr}$ and $\hat{\bar{b}} \in \Re^d$ are the affine mapping weight matrix and the bias vector, respectively, and $\Phi(x)$ is the activation function which has been described above.

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

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

where m is the number of samples, $x^{(i)}$ is the ith input, $f(\cdot)$ and $g(\cdot)$ are the encoder and decoder mapping functions, respectively. Considering the nonlinearity of the activation function as shown in equation (3), the gradient descent algorithm is commonly employed to perform the optimization. A typical utilization of the above autoencoder model is to reconstruct the original input as the output; however, if a distinct feature rather than the same as the input is obtained as the output of the decoder $g(\cdot)$, it can be considered as a kind of nonlinear regression technique.

The latent representation $\bar{h}$ learnt via a typical autoencoder model that retrains the information about the original input $\bar{x}$ is actually not useful by itself for efficient feature learning.[23] The mutual information can be easily maximized by defining the hidden layer dimension r to be the same or larger than the dimension of the input d. This leads to a perfect reconstruction simply by learning an identity mapping between $\bar{x}$ and $\bar{z}$. Due to this effect, the autoencoder model will fail in learning a latent representation which is more useful than the input $\bar{x}$ itself. Ideally, an autoencoder model should perform the input reconstruction task (from $\bar{x}$ to $\bar{z}$) while providing a good latent representation $\bar{h}$. One way to obtain a good latent representation from an autoencoder model is to promote an under-complete representation where $r < d$, thus $\bar{h}$ can be seen as a lossy compression of $\bar{x}$. If the input is drawn completely randomly where each input comes from an independent and identically distributed Gaussian variable that is independent of the other features, then this compression task would be very difficult. However, if there is a structure in the data where the input features are correlated, then this approach will be able to discover some of those correlations while making it difficult for noise to fit in. In fact, a typical autoencoder with linear activation units often ends up learning a low-dimensional representation, which is similar to the result obtained from principal component analysis (PCA).

This argument is relied on $r < d$. When $r \geqslant d$, it is shown that the hidden information in data can still be learnt via imposing other constraints on the network such as a sparsity constraint, particularly on the hidden units. This can avoid the problem of choosing the right amount of hidden units to adjust the model complexity to learn the structure of given data sets while dealing with overfitting well.

Deep sparse autoencoder framework for structural damage identification

This section presents the theoretical development of the proposed deep sparse autoencoder framework for structural damage identification. In the proposed approach, the fact that the natural frequencies and their corresponding mode shapes of a structure relate with the physical properties, such as stiffness, is exploited while accounting for the uncertainties in the structural finite element model and measurement noise in the data. The modal information, such as frequencies and mode shapes, is used as the input to the proposed sparse autoencoder model, and the output will be the structural elemental stiffness parameters. It is noted that the other vibration characteristics, that is, frequency response functions and modal flexibility, could also be used as the input to the framework. In this study, the performance and capability of using several frequencies and mode shapes for damage identification with the proposed framework are investigated. When a number of frequencies and mode shapes are used as input features in a high dimensional space, it may provide more than enough room; hence, unnecessary information may be introduced in the process of learning a mapping from input to output. This could be due to the redundancy in the measurement data and the measurement noise effect that exist in the input space. Learning a relationship from such a high dimensional feature to structural elemental stiffness parameters will likely be less accurate than that of using compressed dimensional features. Therefore, it is necessary to divide the problem into three consequent steps: (1) data pre-processing, (2) sparse nonlinear dimensionality reduction and (3) relationship learning. The proposed sparse framework is formed by these three components connected sequentially, with each component optimized using a specific objective function with relevance to the final objective. Figure 1 shows the structure of the proposed framework, and the specific components will be described in detail in the following sections.

Data pre-processing. The measured structural vibration characteristics, such as frequencies and mode shapes, are pre-processed using a data whitening process, which is a linear transformation that transforms a vector of random variables with a known covariance matrix into a set of new variables with the identity covariance matrix. The purpose of data whitening is to make the input data less redundant and uncorrelated with each other, and all the features have the same identity variance. The transformation is called ‘whitening’ because the input vector is transferred as a white noise vector.

The input features are uncorrelated via the observation of an orthogonal projection matrix U by performing PCA on the original input data

$$x_{rot}^{(i)} = U^T x^{(i)} \tag{4}$$

where $x^{(i)}$ is the ith sample.

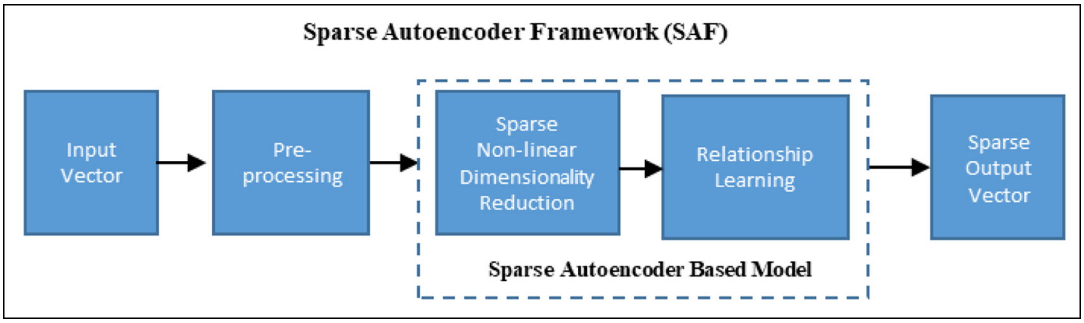

Sparse Autoencoder Framework (SAF)
Input Vector
Pre-processing
Sparse Non-linear Dimensionality Reduction
Relationship Learning
Sparse Output Vector
Sparse Autoencoder Based Model

Figure 1. Structure of the proposed deep sparse autoencoder framework.

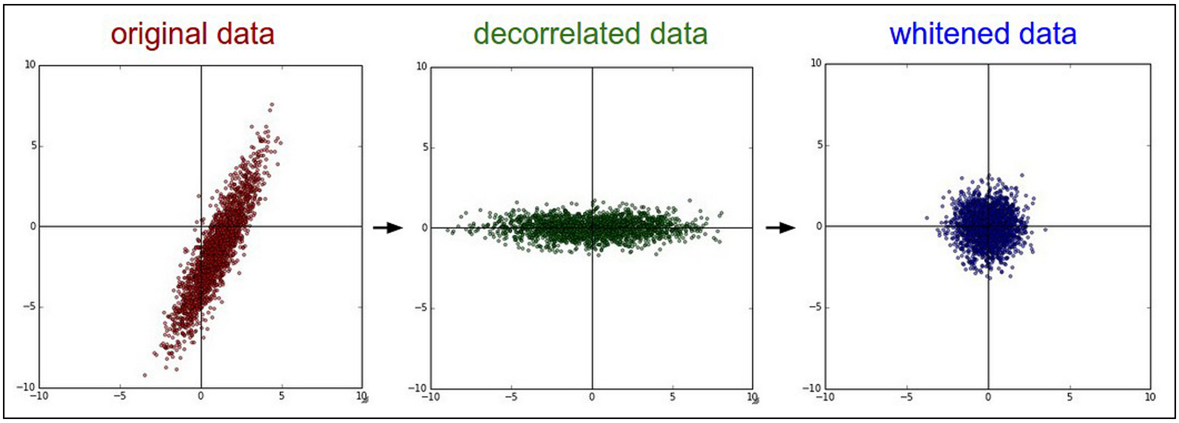


Figure 2. An example reported to show the data whitening process.[24]

To make each of input features have a unit variance, a simple rescale is applied as follows

$$x^{(i)}_{whiten,j} = \frac{x^{(i)}_{rot,j}}{\sqrt{\lambda_j}} \tag{5}$$

where λ_j is the eigenvalue corresponding to the jth eigenvector obtained from PCA, $x^{(i)}_{whiten,j}$ is the jth component of the whitened data sample. Data whitening process uncorrelates and spheres the data based on PCA and provides pre-processed data sets with a less redundancy to perform the training, validation and testing of the network. It is noted that the whitened components that correspond to eigenvalues less than $1e^{-10}$ are discarded and the rest is kept preserving as much as information existed in the original data. Figure 2 shows the schematic process of an example on the data whitening.[24] The first step is to decorrelate the data and the second step is to apply the whitening transformation.

Sparse dimensionality reduction. The main objective of this component is to compress the dimensionality of the features, preserving useful and necessary information that is not only sensitive to structural local damage but also robust to the effect of system uncertainties and measurement noise. A deep neural network based on sparse autoencoder model is trained for the dimensionality reduction, where the first hidden layer is defined to perform the feature fusion of both the frequencies and mode shapes from the structure. The subsequent hidden layers from the second to the kth layer further perform feature compression, as shown in Figure 3. This model can be visualized as the encoding

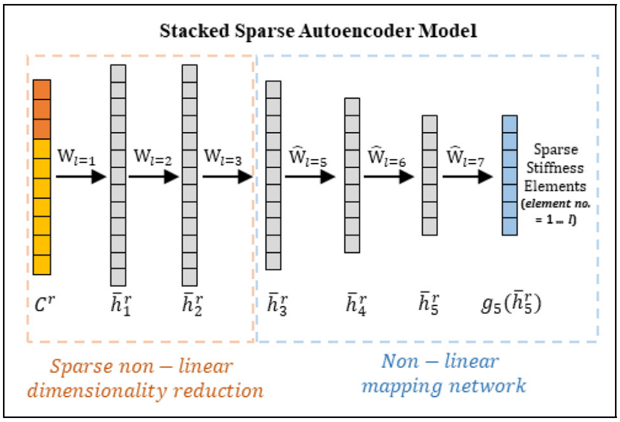


Figure 3. Generic architecture of the sparse autoencoder model.

architecture of a typical DAE[23] but not strictly the generic DAE model with the decoding structure. The brief introductions on the sparse autoencoder, sparse activation function and the formulations of objective functions for the dimensionality reduction component are given in the following sections.

Sparse autoencoder. In this section, the sparse penalty term is explained in detail. Let the activation of a hidden unit j when the network is given with the ith input $x^{(i)}$ be denoted as $h_j(x^{(i)})$, we define

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[h_j(x^{(i)})\right] \tag{6}$$

where m is the number of samples, $\hat{\rho}_j$ is the average activation of the hidden unit j (average over the training set). The enforce constraint is applied as follows

$$\hat{\rho}_j \approx \rho \tag{7}$$

where ρ is a sparsity parameter, which is typically a small value close to zero, for example, $\rho = 0.05$ for the sigmoid activation function. This is chosen via utilizing a validation data set.[25] This enforces the average activation of each hidden neuron j to be close to 0.05. Thus, the hidden unit's activations will mostly be near 0.

Furthermore, an extra penalty term is added to the optimization objective function that penalizes $\hat{\rho}_j$ deviating significantly from ρ as follows

$$KL(\rho||\hat{\rho}_j) = \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{(1-\rho)}{(1-\hat{\rho}_j)} \tag{8}$$

$$J_{sparse}(W,\bar{b}) = \sum_{j=1}^{r} KL(\rho||\hat{\rho}_j) \tag{9}$$

where $KL(.)$ is the Kullback–Leibler divergence[26] between a Bernoulli random variable with the mean equal to ρ and a Bernoulli random variable with a mean of $\hat{\rho}_j$, r is the number of neurons in the hidden layer, and the index j is summing over the hidden units in the network, $J_{sparse}(W,\bar{b})$ denotes the sparse regularization term.

Sparsity-inducing activation and cost function. A widely used way to constraint the information content in the latent representation is to make it sparse or low dimensional.[17] A Rectified Linear Unit (*ReLU*) has special properties over the alternative non-sparse activation functions, that is, 'tanh' and 'sigmoid', in the context of deep neural networks such as information disentangling, efficient variable-size representation, linear separability, distributed sparsity and sparsity-induced regularization. Nevertheless, applying a too strong sparsity constraint may hurt the network prediction performance for an equal number of neurons, since it reduces the effective capacity of the model. In the proposed framework, *ReLU* is utilized carefully to regularize the reconstruction loss function by introducing both a sparsity-inducing term and a weight decay function. The objective cost function is defined as follows

$$J_{\cos t}(W,\bar{b}) = J_{MSE}(W,\bar{b}) + \lambda J_{weight}(W,\bar{b}) + \beta J_{sparse}(W,\bar{b}) \tag{10}$$

where $J_{MSE}(W,\bar{b})$ is the reconstruction loss function; $J_{weight}(W,\bar{b})$ is the weight decay function ($L2$ regularization of all weights); $J_{sparse}(W,\bar{b})$ is the sparsity penalty term as defined in equation (9) which is employed for a better de-noising ability; λ and β are the regularization parameters to balance the reconstruction accuracy and the applied constraints on the solution. The optimal parameter values are obtained via utilizing a validation data set.[25] The weight decay term $J_{weight}(W,\bar{b})$ is added to avoid overfitting and defined as follows

$$J_{weight}(W,\bar{b}) = \frac{1}{2}\sum_{l=1}^{2}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(w_{ji}^{(l)}\right)^2 \tag{11}$$

where $w_{ji}^{(l)}$ represents an element in $W^{(l)}$ and s_l denotes the number of units in the lth layer.

These sparse autoencoders are stacked together to form a deep architecture and learn a robust representation for the input. The first hidden layer performs the feature fusion (both the frequencies and mode shapes from the structure in this study) based on nonlinear dimensionality reduction. The second hidden layer performs compression on the learnt low-dimensional feature observed in the first hidden layer. The following hidden layer performs further compression on the learnt low dimensional feature observed in the previous hidden layer. The robust feature space observed from the final hidden layer will preserve the useful information to perform the mapping to elemental stiffness parameters. It should be noted that the dimensionality of the hidden layers in the sparse nonlinear dimensionality component may not necessarily be smaller than that of the input layer. Choice of the number of nodes per each layer to fit well for a given task is not a straightforward process. Therefore, sparsity constraint is introduced in the objective function and 'ReLu' (sparse activation function) in dimensionality reduction component to allow the network choose the best number of active hidden nodes while training.

Structural natural frequencies and their corresponding mode shapes are fed into the autoencoder as follows

$$\bar{c}^r = [q_1^r \ldots q_i^r, m_1^{q_i r} \ldots m_j^{q_i r}]^T \tag{12}$$

where q_i^r is the ith$(i = 1, \ldots, n)$ structural natural frequency included in the rth sample and $m_j^{q_i r}$ is the $jth(j = 1, \ldots, t)$ mode shape value corresponding to the ith frequency. $\bar{c}^r$ is the concatenated high dimensional feature that combines n frequencies and $n*t$ mode shapes with each frequency having t mode shape values. This feature $\bar{c}^r$ is taken as the input to the proposed sparse autoencoder framework.

Specifically, the reconstruction loss function of the pth layer of the sparse autoencoder model is defined as follows

$$J^p_{MSE}(W, \bar{b}) = \sum_{\tau=1}^{N} \left\| \bar{h}^r_{p-1} - g_p(f_p(\bar{h}^r_{p-1})) \right\|_2^2 \tag{13}$$

where $p = \{1, \ldots, k\}$ with k defined as the last layer in the dimensionality reduction component, N is the number of samples involved in the training, $g(.)$ and $f(.)$ are the decoder and encoder functions, respectively. $\bar{h}^r_{p-1}$ is the low dimensional representation that is established in the $(p - 1)$th layer for the rth sample where $\bar{h}^r_0 = \bar{c}^r$. Encoder function f_p is set to be 'ReLU' since it supports sparse representation of the input, while decoder function g_p is set to be 'purelin' to reconstruct the real values of the input. The objective function defined in equation (10) will be used for training the sparse autoencoders. The latent representation $\bar{h}^r_k$ learnt in the final hidden layer of the dimensionality reduction component, that is the kth layer, is then fed to the relationship learning component, which will be described in the following section.

Relationship learning. The main objective of this component is to learn the relationship between the reduced dimensional feature $\bar{h}^r_k$ and the stiffness parameters of the civil structure. Neural networks with a pre-training scheme are utilized to train the nonlinear relationship mapping. The 'tanh' function is chosen as the activation function to facilitate the nonlinear nature of this problem. The cost function of each layer in the neural networks is defined as follows

$$J^q_{\cos t}(W, \bar{b}) = J^q_{MSE}(W, \bar{b}) + \lambda J^q_{weight}(W, \bar{b}) \tag{14}$$

The same weight decay function as described in equation (11) is used. m hidden layers are defined, and the reconstruction loss function of each layer can be given as follows

$$J^q_{MSE}(W, \bar{b}) = \sum_{\tau=1}^{N} \left\| \bar{o}^r - g_q(f_q(\bar{h}^r_{q-1})) \right\|_2^2 \tag{15}$$

where $q = \{k + 1, \ldots, k + m\}$ for the mth layer in the relationship learning component; $g(.)$ and $f(.)$ are, respectively, the decoder and the encoder functions; $\bar{h}^r_{q-1}$ is the lower dimensional representations that are obtained at the $(q - 1)$th layer for the rth sample; $\bar{o}^r$ is the labelled output vector for the rth sample.

Different layers are defined to have an effect on the global nonlinearity involved in performing the effective relationship learning process. In this way, the error is further reduced in the latter layers. A significant improvement to the global nonlinearity of the problem is achieved via stacking the layers for the final joint optimization. Pre-training for all layers is conducted with the full batch-scaled conjugate gradient BP algorithm.[27] Once the optimal parameters are observed, the whole network is fine-tuned again to optimize all the layers, which will be explained in the next section.

Training and fine-tuning of the full network. The sparse dimensionality reduction and the relationship learning components are combined as a deep neural network. One example of such a deep neural network is shown in Figure 4, where the first two hidden layers employed for the encoding are pre-trained to perform the nonlinear dimensionality reduction and the last three hidden layers are trained to learn the relationship between the compressed dimensional features and the structural stiffness parameters for structural damage identification. In this manner, the deep neural networks manage to retain the required information to establish the relationship between the learnt robust representation and the structural stiffness reductions.

The layer-wise pre-training scheme is performed to mitigate the vanishing gradient problem as mentioned in the work by Hochreiter et al.[3] It is applied to train the proposed sparse autoencoder model, and the training procedure is shown in Figure 4. It can be observed that the hidden layers can be trained one by one to achieve a more efficient and accurate training process.

After the pre-training, the whole deep neural network is then fine-tuned together to optimize all the layers at the same time with the objective function as follows

$$J^F_{\cos t}(W, \bar{b}) = J^F_{MSE}(W, \bar{b}) + \lambda J^F_{weight}(W, \bar{b}) \tag{16}$$

$$J^F_{MSE}(W, \bar{b}) = \sum_{\tau=1}^{N} \left\| \bar{o}^r - p(\bar{c}^r) \right\|_2^2 \tag{17}$$

where $p(\bar{c}^r) = g_L(f_L(f_{L-1}(f_{L-2}(\ldots(\bar{c}^r)))))$ is the estimated elemental stiffness parameter output vector for the rth

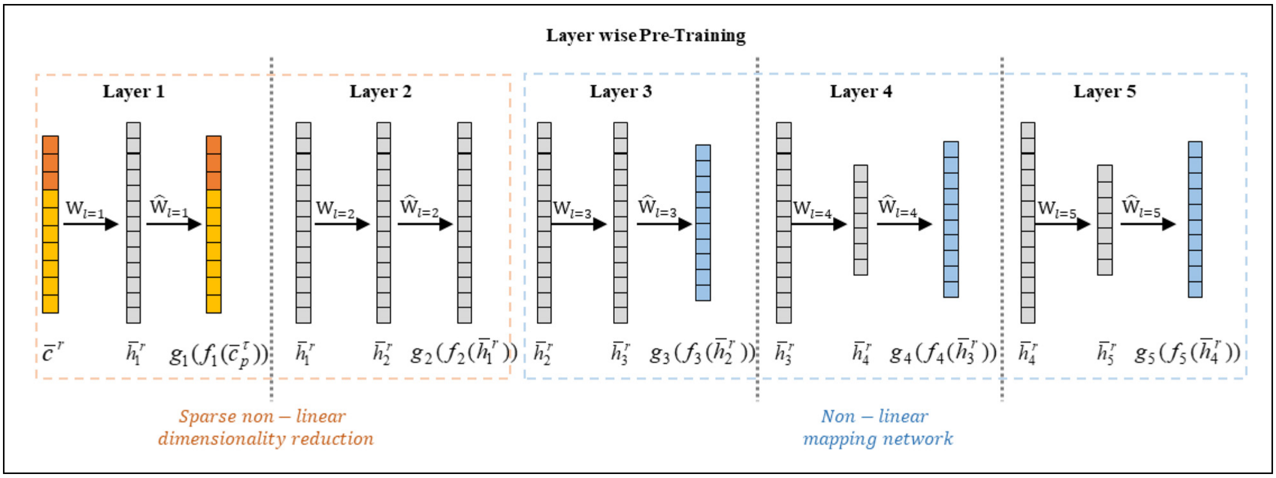


Figure 4. Layer-wise pre-training for the sparse autoencoder model.

sample as shown in Figure 3. The total number of layers is $L = k + m$, and $\bar{o}^r$ is the labelled output vector for the rth sample. Joint optimization of the objective function in equation (16) is conducted to fine-tune the whole networks towards learning the relationship between the feature $\bar{c}^r$ and the stiffness reductions of structures, in order to achieve a better accuracy in damage identification.

Sparsity constraint is enforced only in the dimensionality reduction component due to its effectiveness in reducing the dimensionality via equation (10). In the relationship learning component, it is essential to learn a nonlinear mapping from the reduced dimensional feature to the output which lies in another feature spaces. For this component, the sparsity constraint or the sparse activation function will perform poorly and will not be able to learn an efficient mapping during the training. Therefore, the sparsity-inducing terms are excluded for the relationship learning component, as shown in equations (14) and (15). Furthermore, once the pre-training is performed, the fine-tuning of the whole network is performed with equations (16) and (17), excluding the sparsity term since the lower layers are already pre-trained for sparse representations of the input (most of the hidden nodes' output will be equal to 0 after pre-training). It is important to note that fine-tuning will mostly affect the higher layers (relationship learning component) comparatively to the lower layers (dimensionality reduction component).

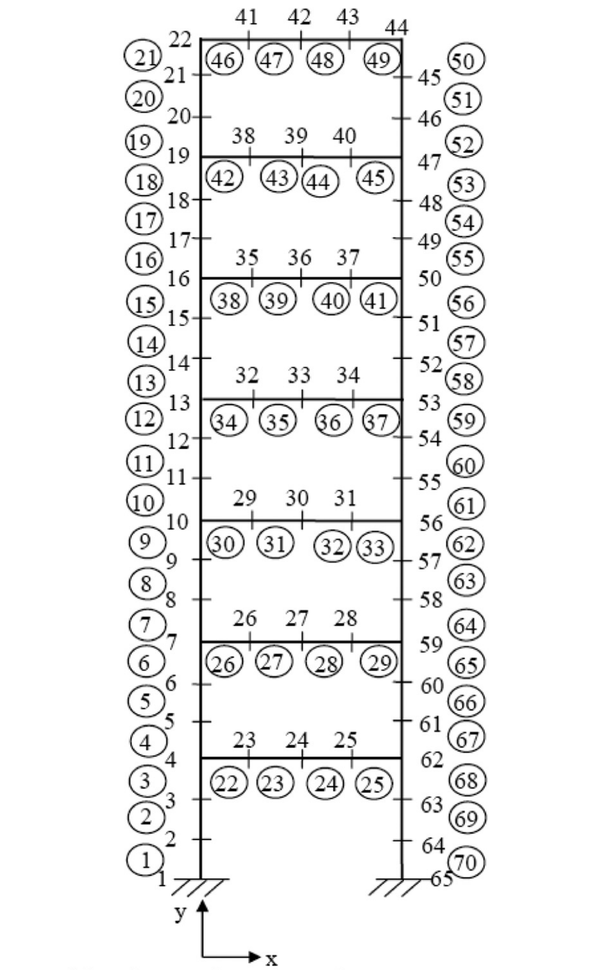
Numerical studies

In this section, numerical studies on a steel frame structure are conducted to investigate the accuracy and robustness of the proposed framework for structural damage identification, particularly considering the effects of noise in the measurement data and uncertainties in the finite element modelling. The numerical model, data generation, data pre-processing and performance evaluation of the proposed framework will be presented. The accuracy and efficiency of applying the proposed framework for structural damage identification are validated with the simulation data. Furthermore, the effects of the uncertainties in the finite element modelling and measurement noise effects in the data on the performance of the proposed approach for structural damage identification will be studied.

Numerical model

A scaled seven-storey steel frame structure model as shown in Figure 5 is used as a numerical example to generate the training data and verify the accuracy and efficiency of the proposed framework. The column of the steel frame structure 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 defined as 49.98 mm × 4.85 mm and 49.89 mm × 8.92 mm, respectively. The mass densities of the column and beam elements are 7850 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 element in each floor to simulate the mass of a building structure. The bottoms of the two columns of the frame are fixed.

The finite element model of the frame structure, as shown in Figure 5, consists of 65 nodes and 70 planar



Note: (1): 1 denotes the node number
(2): (1) denotes the element number in the structure

Figure 5. Finite element model of the steel frame structure model.

frame elements. The weights of steel blocks are added at the corresponding nodes of the finite element model as concentrated masses. Each node has 3 degrees of freedom (DOFs), with two translational displacements along x and y axles and a rotational displacement θ. In total, the model has 195 DOFs. 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 frame structure is manufactured in the laboratory. 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 descriptions of the model and the updating process can be found in the work by Li and Hao.[28] This updated finite element model is taken as the baseline model for generating the training, validation and testing data in the numerical study.

Data generation

Modal analysis is performed using the baseline model with different simulated damage scenarios to generate the input and output data for training the proposed framework. The first seven frequencies and the corresponding mode shapes of 14 beam-column joints are obtained as the input data to the network. 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 equals to 0.9, 10% stiffness reduction is introduced in this specific element. A total of 12,400 data sets are generated based on the baseline model to include both the single element and multiple 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 other elements undamaged. Therefore, 30 data sets are generated for the scenario when a local damage is introduced in a specific element. With 70 elements in the finite element model, 2100 single element damage cases are obtained. In multiple element damage cases, the stiffness parameters for random two or more elements vary from 1, 0.99, 0.98, …, to 0.7 while keeping the other elements undamaged. In total, 10,300 multiple element damage cases are defined. It is noted that a maximum number of six damaged elements are considered in the multiple damage cases. The first seven frequencies and the corresponding mode shapes at those 14 beam-column joints are taken as the input, and the pre-defined elemental stiffness reduction parameters are considered as the labelled output. These generated input and output data are used for the training, validation and testing of the proposed framework.

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

1. Scenario 1: No measurement noise and modelling uncertainty, that is, no noise effect in the vibration characteristics and uncertainties in the finite element modelling is considered;
2. Scenario 2: Measurement noise effect considered. White noises are added on the input vectors, that is, 1% noise in the frequencies and 5% in the mode

shapes, considering structural frequencies are usually identified more accurately than mode shapes;[29]

3. Scenario 3: Uncertainty effect included. About 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 in Scenarios 2 and 3 are considered together.

Data pre-processing

Concatenated feature $\bar{c}^r$ includes both the frequencies and mode shapes that are measured in different scales; therefore, it is necessary to perform a normalization process before it is fed into the network. Data whitening described in section 'Data pre-processing' followed by a feature scaling phase (scaling to a value between 0 and 1) is performed. This will introduce a common ground for all features to be treated equally in the learning process. The data range 0 to 1 is chosen since the operating range of the activation function 'ReLU' falls into 0 and 1. Furthermore, the initial weights of the neural networks before pre-training are chosen randomly in a way that the input to the 'ReLU' hidden unit will lie in the positive range. Considering structural damages are usually observed at a few number of elements, sparse output vector is defined by defining 0 as the intact state and 1 as 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 using the proposed framework for structural damage identification based on the pre-processed data sets will be described in the following section.

Performance evaluation of the proposed approach

Performance evaluation of the proposed approach against the state-of-the-art method 'Autoencoder Based Network (AutoDNet)'[16] is conducted with four different scenarios as described in section 'Data generation'. It is noted that the improved performance of using 'AutoDNet' for structural damage identification has been demonstrated against ANN. Good accuracy in identifying structural damage using ANN methods can be achieved with the optimally selected network design and pattern features.[30,31] The proposed approach is able to deal with a large amount of data sets and identify the damage effectively with significant uncertainties and measurement noise effect. The layer-wise pre-training and fine-tuning are employed to improve the training efficiency and accuracy. It will also be demonstrated that the proposed approach is applicable for more complex problems with a complicated network structure with a high dimensional input data, multiple hidden layers, a large number of output parameters and significant uncertainties.

Table 1. Architecture information of the used three networks.

Architecture		Dimensionality reduction component	Relationship learning component
AutoDNet	Layers	2	1
	Activation function	Tanh	Tanh
	Nodes	100 and 100	80
SAF-0	Layers	2	1
	Activation function	ReLu	Tanh
	Nodes	100 and 100	80
SAF	Layers	2	3
	Activation function	ReLu	Tanh
	Nodes	100 and 100	90, 80 and 70

Therefore, in this study, the performance of the proposed framework will be compared with 'AutoDNet'. The structure of the defined sparse autoencoder model used in this numerical study includes $k = 2$ hidden layers with each comprising 100 hidden nodes in the dimensionality reduction component and $m = 3$ hidden layers with each comprising 90, 80 and 70 neurons defined, respectively, in the relationship learning component. In this study, a deeper neural network is defined with more hidden layers than the previous study.[16] In order to perform a fair comparison with 'AutoDNet' method, a separate sparse autoencoder model named as SAF-0 with the same structure as that used in 'AutoDNet' method is also formed by choosing $k = 2$ number of hidden layers with each comprising 100 hidden nodes in the dimensionality reduction component, and $m = 1$ hidden layer with 80 neurons used in the relationship learning component. Table 1 lists the architecture information of the above three networks.

Sparsity constraint and sparse activation functions are utilized to select the optimal number of hidden nodes in each layer. The choice of the number of hidden nodes in each layer is made by the network during the training process. A Bayesian model class selection method has been developed to select an optimal ANN model.[30,31] In this study, sparsity constraint is introduced in the objective function and ReLu (sparse activation function) in dimensionality reduction component to let the network choose the best number of active hidden nodes while training.

The number of entries in the original input vector includes seven frequencies and 14×7 mode shape values, that is, 105 in total. In total, 70 elemental stiffness parameters are included in the final output vector. The

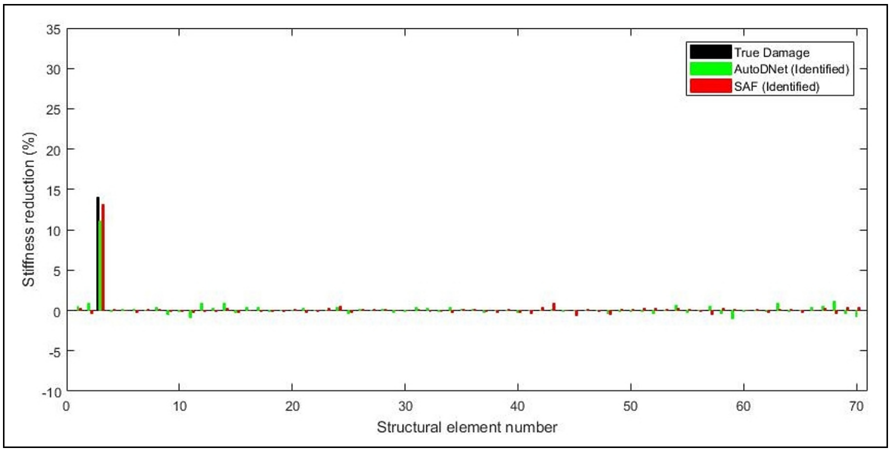


Figure 6. Damage identification results of a single damage case from 'AutoDNet' and the proposed framework for Scenario 1.

complexity of the target problem is relatively high considering that a large number of parameters are included in the input and output. A more complex problem may require a deeper neural network architecture. The nature of the problem will be considered to determine the number of hidden layers and hidden units of the sparse autoencoder model. *ReLU* and linear function are employed in the sparse dimensionality reduction component, while tangent and linear functions are employed in the relationship learning component in the pre-training scheme. After the pre-training, the same configuration is preserved for the hidden layers in order to fine-tune the whole network. Training, validation and testing data sets are defined to have 70%, 15% and 15% of the pre-processed data sets, respectively. In order to evaluate the performance of using the proposed framework for structural damage identification, MSE and regression value (*R* value: the coefficient of multiple determination for multiple regression) are employed to show the quality in the network training. All the numerical computations are conducted using a desktop with an Intel i7 processor, 16 GB RAM and the graphics card NVidia 1080 Ti GTX using GPU for parallel computing.

Scenario 1: no measurement noise and modelling uncertainties. In this scenario, the data sets without measurement noise and uncertainty effect are used. The performance of using 'AutoDNet', SAF-0 and the proposed framework is compared by examining the MSE values and *R* values on the testing data sets. The performance evaluation results are shown in Table 2.

Table 2. Performance evaluation results for Scenario 1 in the numerical study.

Methods	MSE	*R* value
AutoDNet	2.5e−04	0.921
SAF-0	8.27e−05	0.975
The proposed framework	2.9e−05	0.993

MSE: mean square error.

The complexity of this structural damage identification problem is relatively high with 70 structural elemental stiffness parameters included in the output vector. 'AutoDNet' may not be able to achieve a very good accuracy with several hidden layers without inducing the sparsity constraint. *R* value obtained from 'AutoDNet' is 0.921, while SAF-0 is 0.975, as shown in Table 2. This shows the effectiveness and improvement of the sparse dimensionality reduction process. The improvement in *R* value obtained from SAF-0 and the proposed framework shows the effectiveness of utilizing more relationship learning layers and a deeper neural networks as described in section 'Training and fine-tuning of the full network'. The results from the proposed framework show the improvement in the network prediction compared with 'AutoDNet' and SAF-0 with a smaller MSE value and a better *R* value close to 1. To further demonstrate the performance of using the proposed framework for structural damage identification, Figures 6 to 8 show the identification results of several single damage and multiple damage cases randomly selected from the testing data sets. It shall be noted that the results from the proposed framework are shown in this figures with a legend of 'SAF'.

The damage identification results of a single damage case obtained from 'AutoDNet' and the proposed framework are shown in Figure 6. It can be observed that the proposed framework provides more accurate damage identification results than 'AutoDNet' with less minor false positives and negatives. The damage location is well identified, and the identified stiffness

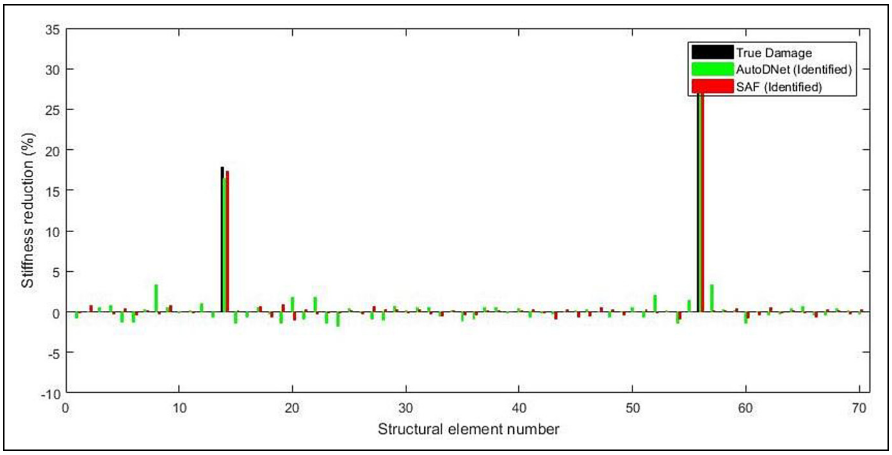


Figure 7. Damage identification results of a multiple damage case from 'AutoDNet' and the proposed approach for Scenario 1.

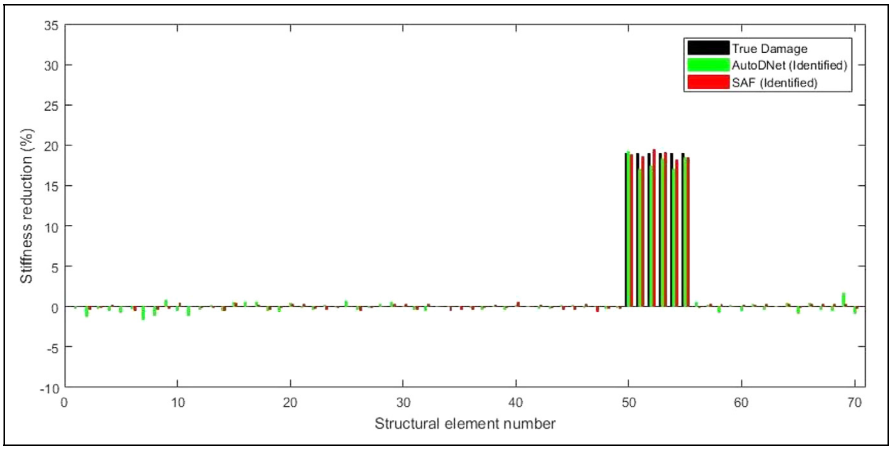


Figure 8. Damage identification results of a continously distrubted multiple damage case from 'AutoDNet' and the proposed approach for Scenario 1.

reduction at the damaged element is very close to the actual value from the proposed framework. The identified stiffness reduction values at the non-damage elements are very close to zero. The proposed approach is also evaluated against 'AutoDNet' with multiple structural damage cases, and the identification results of two multiple damage cases randomly selected from the testing data sets are shown in Figures 7 and 8. It can be seen clearly that the proposed approach works very well for the identification of different types of multiple damage cases, for example, two separate damages as shown in Figure 7 and continously distributed multiple damages as shown in Figure 8. Damage locations are accurately detected, and the identified stiffness reductions are very close to the actual values with very small false identifications using the proposed framework.

Table 3. Performance evaluation results for Scenario 2 in the numerical study.

Methods	MSE	*R* value
AutoDNet	3.7e−04	0.794
SAF-0	2.7e−04	0. 858
The proposed framework	2.3e−04	0.886

MSE: mean square error.

Scenario 2: with measurement noise. The performance evaluation of the proposed approach is conducted when the noise effect in the measurement data is considered. It is noted that 1% noise is included in the frequencies and 5% in the mode shapes. As shown in Table 3, a significant performance improvement in *R* value is observed using the proposed approach compared with

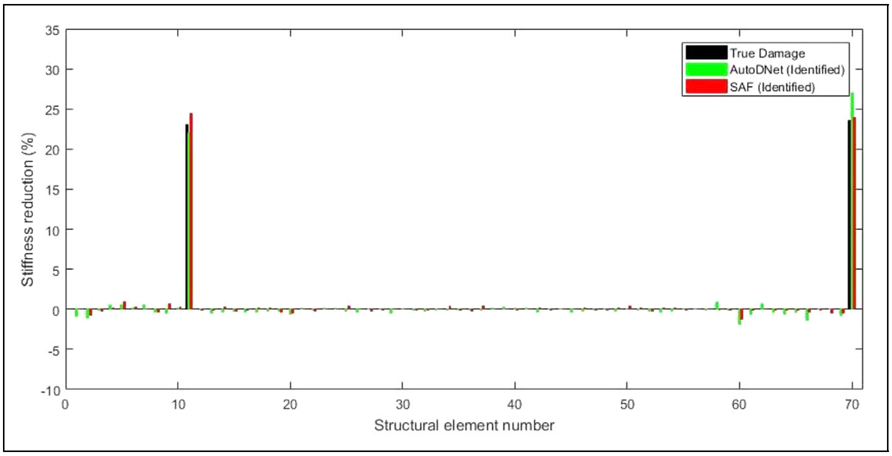


Figure 9. Damage identification results of a multiple damage case from 'AutoDNet' and the proposed approach for Scenario 2.

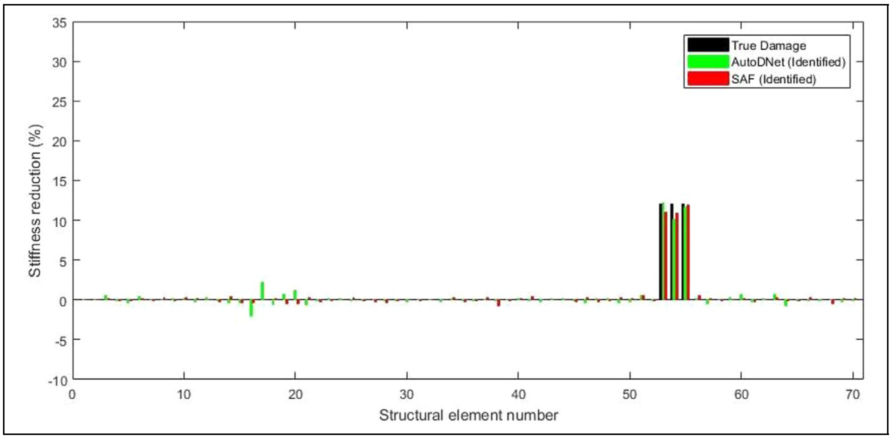


Figure 10. Damage identification results of another multiple damage case from 'AutoDNet' and the proposed approach for Scenario 2.

'AutoDNet'. It shows the robustness and effectiveness of using the sparsely learned features from the dimensionality reduction component for relationship learning when the noise effect is included in the measurements.

To further demonstrate the quality of the prediction in terms of both magnitudes and the locations in the damage identification, two multiple damage identification results are shown in Figures 9 and 10, respectively. Multiple structural damage identification is challenging and needs more precision on the identification of accurate stiffness reductions at the exact stiffness elements compared to the single damage cases, especially under the significant noise effect. It is observed that damage locations are accurately identified and the identified stiffness reductions are also very close to the actual values with very small false identifications. A better accuracy is achieved with the proposed approach than 'AutoDNet' regarding identifying the damage magnitudes, even for the minor damage cases as shown in Figure 10.

Scenario 3: with modelling uncertainty. In this study, an accurate finite element model is required to generate the training data. When a model-based method is performed for structural damage identification, the inevitable uncertainties existed in the finite element modelling will significantly affect the accuracy and performance of structural damage identification. In this scenario, the uncertainty in the finite element modelling, that is, 1% uncertainty in the elemental stiffness parameters are considered to simulate the finite element modelling errors. The performance evaluation results are shown in Table 4. The proposed approach outperforms 'AutoDNet' with an improvement in both MSE and R value. The results from 'AutoDNet' is affected by the

Table 4. Performance evaluation results for Scenario 3 in the numerical study.

Methods	MSE	*R* value
AutoDNet	2.9e−04	0.83
SAF-0	5.2e−05	0.975
The proposed framework	2.9e−05	0.986

MSE: mean square error.

uncertainty effect, as reflected by the corresponding lower *R* value.

To demonstrate the robustness of the proposed framework to the uncertainty effect, damage identification results of a single and a multiple damage cases randomly selected from the testing data sets are shown in Figures 11 and 12, respectively. It is noted that the random stiffness distributions in those figures are the introduced uncertainties in the structural system. As shown in both cases, the identified stiffness reductions are very close to the actual values with very small false identifications, even when the finite element modelling errors are present in the input data. By comparing these identification results with those form 'AutoDNet', the accuracy and robustness of using the proposed approach for structural damage identification with uncertainty effect are demonstrated. The proposed approach gives more accurate damage identification results.

Scenario 4: with both the measurement noise and modelling uncertainty. Both the measurement noise and the modelling uncertainty in Scenarios 2 and 3 are considered together in this section to further investigate the performance and effectiveness of the proposed approach. It is very challenging to achieve an effective and reliable structural damage identification when both measurement noise and modelling uncertainty are present in the input data. These uncertainties may adversely affect

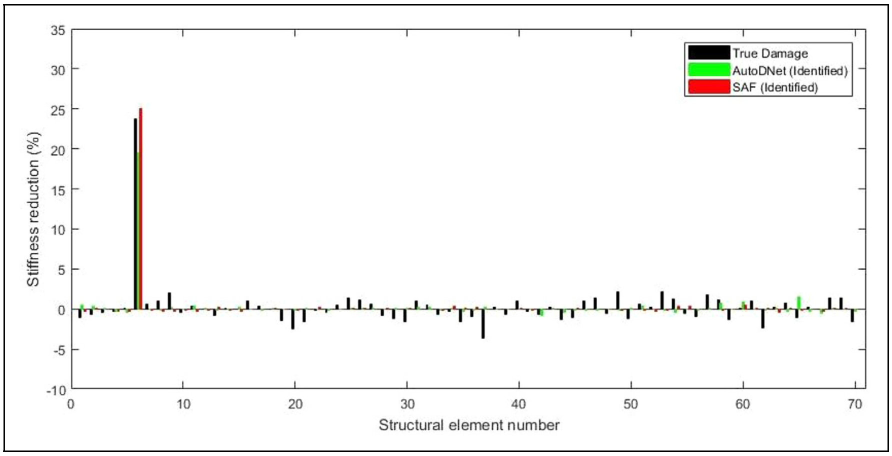


Figure 11. Damage identification results of a single damage case from 'AutoDNet' and the proposed approach for Scenario 3.

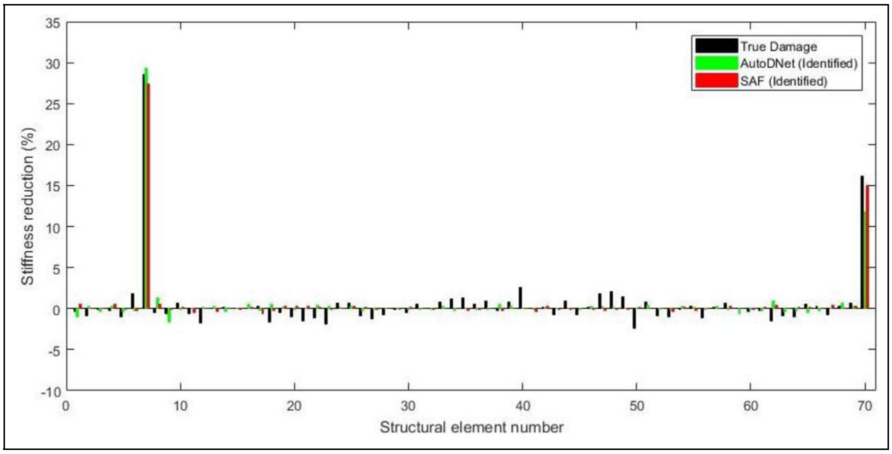


Figure 12. Damage identification results of a multiple damage case from 'AutoDNet' and the proposed approach for Scenario 3.

Table 5. Performance evaluation results for Scenario 4 in the numerical study.

Methods	MSE	*R* value
AutoDNet	3.6e−04	0.732
SAF-0	3.3e−04	0.763
The proposed framework	3.2e−04	0.792

MSE: mean square error.

the damage detection results. The performance evaluation results for this scenario are shown in Table 5.

As observed in Table 5, the proposed approach once again outperforms the 'AutoDNet' when both the measurement noise and uncertainty effect are considered. This is evidenced by a higher *R* value and a lower MSE value. *L*2-weight decay along with the sparsity constraint applied on the cost function formulation in this study ensures that the proposed approach has a smaller space to fit the training data and therefore effectively avoids the overfitting problem and consequently improves the accuracy and robustness for the structural damage identification. Figures 13 and 14 show the identification of a single damage case and a multiple damage case, respectively. It can be observed that the 'AutoDNet' method produces several false identifications with around 3% damage values, while the proposed approach does not produce false positives at those locations. The identified damage severities using the proposed approach are closer to the true damage values. Regarding the multiple damage case, the proposed approach also provides much more accurate stiffness reduction predictions than 'AutoDNet' in terms of both the damage locations and severities. With the considered uncertainties and noise effect, a minor stiffness reduction in 5% can be well identified.

Damage identification results from the above four scenarios demonstrate clearly the accuracy and

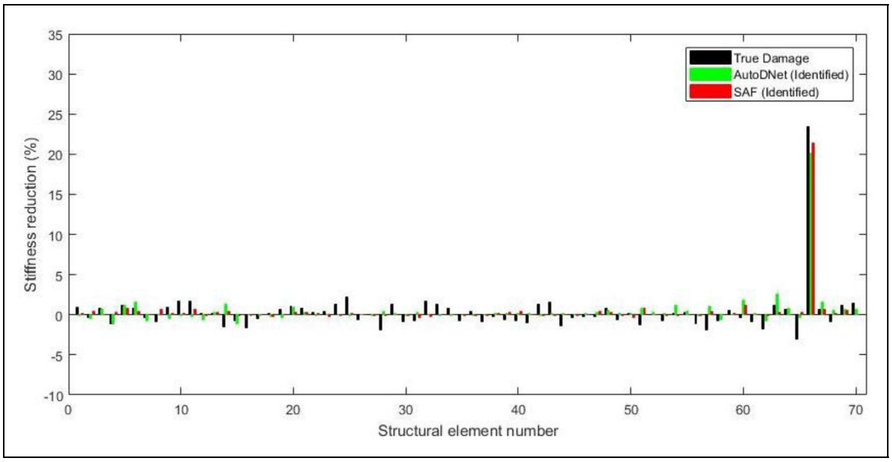


Figure 13. Damage identification results of a single damage case from 'AutoDNet' and the proposed approach for Scenario 4.

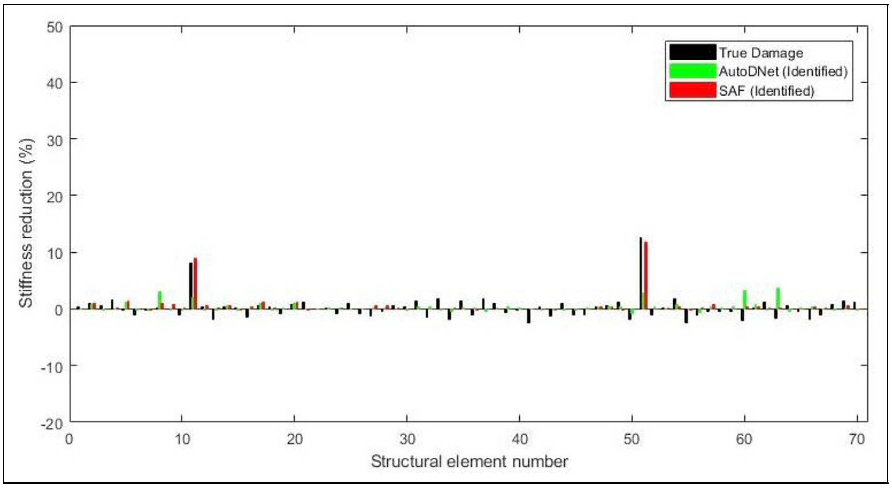


Figure 14. Damage identification results of a multiple damage case from 'AutoDNet' and the proposed approach for Scenario 4.

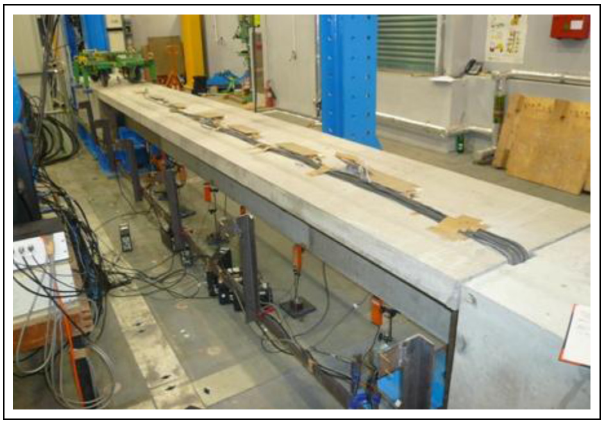


Figure 15. The experimental testing model.

robustness of using the proposed approach in structural damage identification, compared with the latest existing study based on ‘AutoDNet’, even when the measurement noise and uncertainty effect are both present in the input data. The improvement is also demonstrated by the identification results from various scenarios defined in this numerical study.

Experimental verification

Experimental verification on a laboratory reinforced concrete bridge model using the proposed approach for damage identification will be described in this section. The experimental setup, modal testing, initial model updating, training data generation, structural design of the neural networks and damage identification results will be described in the following sections.

Experimental model and initial model updating

A simply supported T-section prestressed concrete bridge model, as shown in Figure 15, is fabricated in the laboratory and tested to verify the effectiveness of applying the proposed framework for structural damage identification. Figure 16 shows the dimensions of the bridge model and the locations of placed accelerometers for the modal tests. The bridge is 5-m long. The widths of the slab and web are 0.65 and 0.15 m, respectively. The height of the beam is 0.415 m. The initial Young’s modulus and density are 2.6 × 10^4 MPa and 2707.7 kg/m^3, respectively. Three prestressing tendons with each having 99.8-mm^2 area are included in the bridge web with a total prestress force of 140 kN. The tensile strength of the tendons is 1949 N/mm^2. The cable profile is parabolic and locations of the prestress tendons at the ends and mid-span of the bridge model are shown in Figure 16. The cable duct is grouted after prestressing. Seven accelerometers are placed on the top of the bridge model for recording the dynamic vibration responses in the vertical direction.

The initial model updating is conducted to create a baseline model for generating the training data.

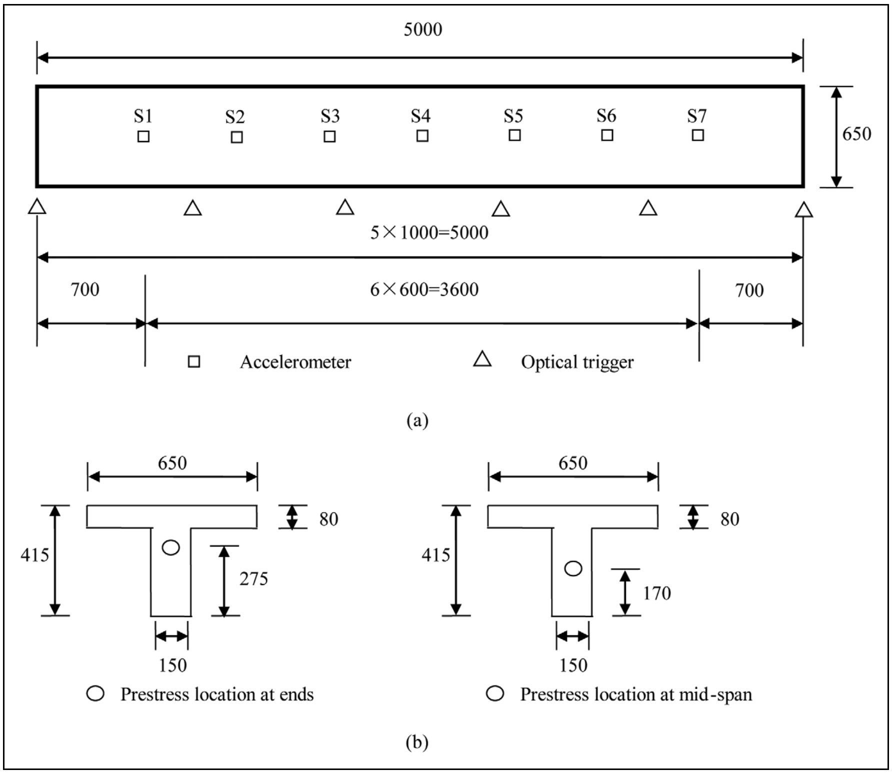


Figure 16. Dimensions of the testing model and the sensor placement.

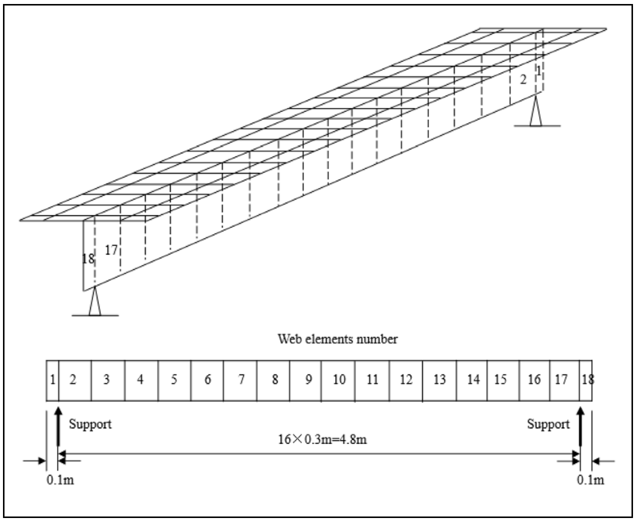


Figure 17. Finite element model of the testing bridge.

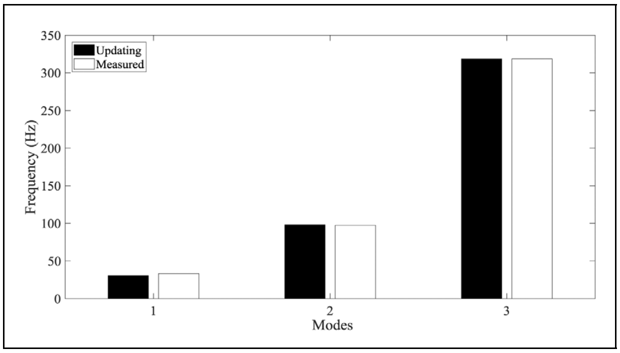


Figure 18. Updated frequencies from the finite element model.

Dynamic vibration tests are conducted to identify the vibration characteristics of the model using a modal hammer to excite the model. The sampling rate is set as 2000 Hz to well cover the frequency range of excited modes of the bridge model. An initial finite element model of the bridge is built with flat shell elements, as shown in Figure 17. The finite element model consists of 90 elements and 114 nodes with 6 DOFs at a node. The model has 684 DOFs in total. The initial model updating is conducted to adjust the built finite element model to serve as the baseline model. The model updating is conducted by minimizing the difference between the first three natural frequencies and mode shapes calculated from the finite element model and measured from the tests. In the initial model updating, Young's modulus of slab and web of beam and the support stiffness are selected as parameters to be updated. The dimensions and mass density are measured and not included as the updating parameters. The identified damping ratios of the beam are included in the initial finite element model. Identified modal information, for example, natural frequencies and mode shapes of the first three modes, is used to perform the initial model updating. The updated natural frequencies are close to the measured ones, as shown in Figure 18. The detailed experimental model, updating procedure and results can be found in the work by Li et al.[32] This baseline model will be used in the following studies to generate the training data.

Introduced damage scenario and training data generation

Two-point static loads are applied in the mid-span of the bridge model to introduce the cracks in the model. The static load is continuously increased to 180 kN and a number of cracks are observed in the web elements at the middle span, as shown in Figure 19. This will be considered as the damaged state. The vibration tests on the damaged bridge are conducted again to identify the first three natural frequencies and the corresponding mode shapes, which will be used later to investigate the performance of the trained sparse autoencoder model defined in this experimental study for damage identification with the real measurement data.

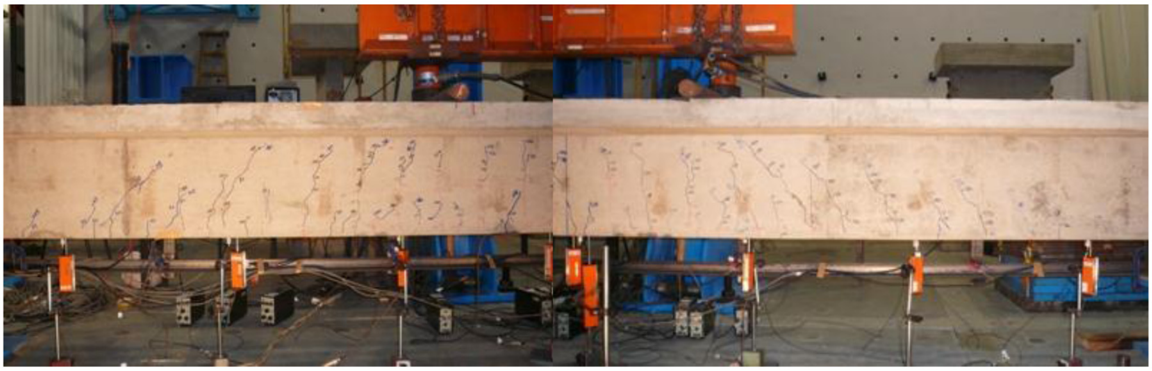


Figure 19. Introduced cracks in the web elements of the tested bridge.

When using the baseline finite element model to generate the training data with simulated different damage scenarios, it shall be noted that only the web elements from No. 2 to No. 17, as shown in Figure 17, are included in the damage identification since the damages are mainly observed in the web elements of the bridge model. Uniformly distributed random damages are simulated in all the 16 web elements from No. 2 to No. 17 with the damage severity distributed between 0% and 15%. The step of the damage severity is defined as 0.01%. A total of 20,000 data samples are generated. Eigenvalue analysis is then conducted based on the finite element model with the introduced damage to obtain the vibration characteristics, that is, the first three frequencies and mode shapes at the placed sensor locations. The obtained modal information will be taken as the input to the neural networks, and the simulated damage will be targeted as the labelled output. These data sets will be taken as the training data in this experimental investigation.

The sparse autoencoder deep neural network structure

Considering the complexity of the target problem and the number of parameters in the input and output vectors, the structure of the sparse autoencoder model is properly defined to have one hidden layer ($k = 1$) with 16 neurons in the dimension reduction component and one layer ($m = 1$) with 16 neurons in the relationship learning component. The input vector includes three frequencies and 3×7 mode shape values, that is, 24 values in total. In total, 16 stiffness reduction parameters are involved in the final output vector. *ReLU* and linear function are employed in the sparse autoencoders in the dimensionality reduction component, while tangent and linear functions are used in the relationship learning component in the pre-training scheme. After the pre-training, the same configuration is preserved for the hidden layers in order to fine-tune the whole network, based on the procedure as described in section 'Deep sparse autoencoder framework'. To have a fair comparison, the same number of hidden layers and neurons are used to form an 'AutoDNet' model and the same training data sets are used for comparing the performance. In order to evaluate the quality of the damage predictions using 'AutoDNet' and the proposed framework, MSE and *R* value are employed.

Training performance and damage identification results

Considering that the measurement data recorded in the laboratory inevitably include the noise effect, a robust deep neural network is required to accommodate this effect. To this end, the noise effect is included in the modal information of the original training data, for example, 1% noise in the frequencies and 5% in the mode shapes. These data sets will be pre-processed with data whitening and used as the training data for the proposed sparse autoencoder model. Training, validation and testing data sets are randomly chosen from the pre-processed data sets with percentages of 70%, 15% and 15% of data samples, respectively. The performance evaluation results for the testing data sets using 'AutoDNet' 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 the one from 'AutoDNet'. Besides, the regression from the proposed approach is also improved, as observed from the *R* values.

Table 6. Performance evaluation results in the experimental study.

Methods	MSE	*R* value
AutoDNet	1.38e−05	0.983
The proposed approach	4.57e−06	0.998

MSE: mean square error.

To further investigate the performance of the proposed model using real testing measurements for structural damage identification, the modal information obtained from the damaged state is used as the input to the trained sparse autoencoder model. Figure 20 shows the identified structural damage comparing with those obtained from the 'AutoDNet' and the proposed approach. It is demonstrated that the identified pattern of stiffness reductions from the proposed approach is closer to those shown in Figure 19, compared to the result from 'AutoDNet'. The main damage identified by the proposed approach is located at the centre of the bridge model, which matches well with the observed cracks as shown in Figure 19. Similar damage pattern with most stiffness reductions distributed in the centre of the beam can be observed from 'AutoDNet' method. However, it should be noted that significant false identification is observed from the 'AutoDNet' method, that is, in element 12. Since there is no analytical model relating to the crack damage in a prestressed concrete beam with its flexural stiffness, it is impossible to calculate the analytical damage extents based on the observed cracks such as those shown in Figure 19. Therefore, the identified damage pattern will be compared with the observed one. There are in total 24 major cracks observed in the experimental tests, and these major cracks are mainly located in the six web elements from No. 8 to No. 13. The identified damage pattern by the proposed sparse autoencoder model

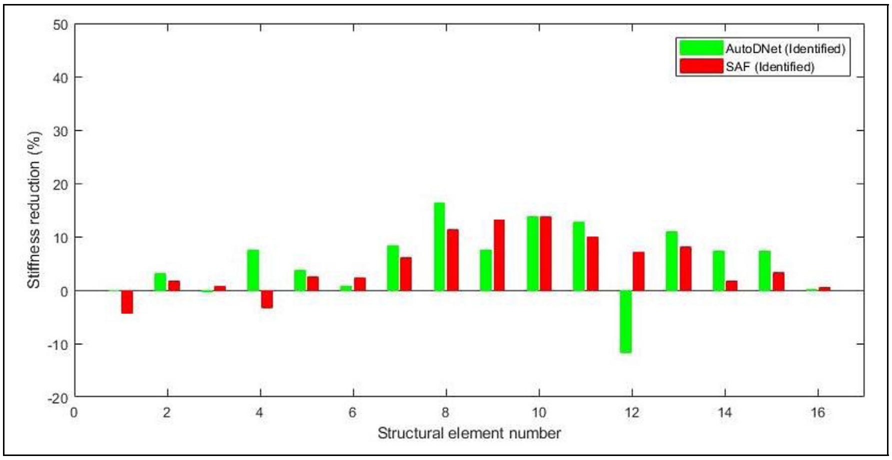


Figure 20. Damage identification results from 'AutoDNet' and the proposed approach in the experimental study.

shown in Figure 20 has a good agreement with the experimental crack pattern observed in Figure 19. The identified damages from the proposed approach are mainly distributed in web elements from No. 8 to No. 13. These results indicate that the proposed approach can well identify the introduced structural damages in the laboratory testing model with real measurement data. It is also observed in Figure 20 that there are some negative values of stiffness reductions. Those are false predictions, which are probably due to the cost function utilized in the last layer of the network. MSE does not consider the sign of the damage but only the magnitude.

Conclusion

The proposed deep sparse autoencoder framework learns the relationship between the modal information, such as frequencies and mode shapes, and structural stiffness parameters. Three main components are defined in this framework, namely, pre-processing, sparse dimensionality reduction and relationship learning. A deep neural network structure associated with the sparse autoencoders is defined to enhance the capability and performance of dimensionality reduction and relationship learning components with a pre-training scheme. Then, these two components are jointly optimized to fine-tune the whole network towards a better accuracy in structural damage identification. A sparse regularization term is adopted in this framework considering that the structural damage usually occurs at only a few elements out of the entire structure. Numerical studies on a steel frame structure are conducted to investigate the accuracy and robustness of using the proposed framework for structural damage identification, particularly considering the effects of noise in the measurement data and uncertainties in the finite element modelling. Experimental studies on a prestressed concrete bridge in the laboratory are conducted to further validate the performance of using the proposed framework for structural damage identification. The results demonstrate that the proposed framework provides a better accuracy and robust performance in structural damage identification than 'AutoDNet' method in the latest study.

Compared with 'AutoDNet' method, the proposed framework has three significantly different features which strongly improve the effectiveness and robustness of structural damage identification against the measurement noise and modelling uncertainty: (1) the data whitening process is applied to uncorrelate the data and transform them into a set of new variables with the identity covariance matrix, as a pre-processing procedure; (2) the sparse regularization term is employed in the training of the proposed framework to improve the performance in damage identification; and (3) a deep neural network structure is defined to include more hidden layers. The comparison results well demonstrate the superiority of the proposed approach. It should be noted that only natural frequencies and mode shape values of the first several modes are taken as the input in this study. The proposed framework can be explored further using other vibration characteristics which may be more sensitive to structural damage for structural damage identification. This will be conducted in future studies.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The work described in this article was supported by an Australian Research Council project.

ORCID iD

Jun Li https://orcid.org/0000-0002-0148-0419

References

1. Padil KH, Bakhary N and Hao H. The use of a non-probabilistic artificial neural network to consider uncertainties in vibration-based-damage detection. *Mech Syst Signal Pr* 2017; 83: 194–209.
2. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Networks* 2015; 61: 85–117.
3. Hochreiter S, Bengio Y, Frasconi P, et al. Gradient flow in recurrent nets: the difficulty of learning longterm dependencies. In: Kolen JF and Kremer KC (eds) *A field guide to dynamical recurrent neural networks*. New York: IEEE Press, 2001, pp. 2–8.
4. Bengio Y, Simard P and Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE T Neural Networ* 1994; 5(2): 157–166.
5. Hinton GE and Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science* 2006; 313(5786): 504–507.
6. Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th international conference on machine learning*, Helsinki, 5–9 July 2008, pp. 1096–1103. New York: ACM.
7. Ranzato M, Boureau YL and Le Cun Y. Sparse feature learning for deep belief networks. *Adv Neur In* 2008; 20: 1185–1192.
8. Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 2010; 11: 3371–3408.
9. Srivastava N. *Improving neural networks with dropout*. Master's Thesis, University of Toronto, Toronto, ON, Canada, 2013.
10. Goodfellow IJ, Warde-Farley D, Mirza M, et al. Maxout networks. JMLR WCP 2013; 28(3): 1319–1327.
11. Arel I, Rose DC and Karnowski TP. Deep machine learning – a new frontier in artificial intelligence research. *IEEE Comput Intell Mag* 2010; 5(4): 13–18.
12. Jia F, Lei Y, Lin J, et al. Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech Syst Signal Pr* 2016; 72: 303–315.
13. Bao Y, Tang Z, Li H, et al. Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Struct Health Monit*. Epub ahead of print 19 February 2018. DOI: 10.1177/1475921718757405.
14. Lin YZ, Nie ZH and Ma HW. Structural damage detection with automatic feature-extraction through deep learning. *Comput-Aided Civ Inf* 2017; 32: 1025–1046.
15. Cha YJ, Choi W and Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. *Comput-Aided Civ Inf* 2017; 32(5): 361–378.
16. Pathirage CSN, Li J, Li L, et al. Structural damage identification based on autoencoder neural networks and deep learning. *Eng Struct* 2018; 172: 13–28.
17. Glorot X, Bordes A and Bengio Y. Deep sparse rectifier neural networks. *PMLR* 2011; 15: 315–323.
18. Martinez AM. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE T Pattern Anal* 2002; 24(6): 748–763.
19. Fidler S, Skocaj D and Leonardis A. Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE T Pattern Anal* 2006; 28(3): 337–350.
20. Doi E, Balcan DC and Lewicki MS. A theoretical analysis of robust coding over noisy overcomplete channels. In: *Proceedings of the 18th international conference on neural information processing systems*, Vancouver, BC, Canada, 5–8 December 2005, pp. 307–314. New York: ACM.
21. Olshausen BA and Field DJ. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Res* 1997; 37(23): 3311–3325.
22. Ranzato M, Poultney C, Chopra S, et al. Efficient learning of sparse representations with an energy-based model. In: *Proceedings of the 19th international conference on neural information processing systems*, Vancouver, BC, Canada, 4–7 December 2006, pp. 1137–1144. New York: ACM.
23. Bengio Y. Learning deep architectures for AI. *Found Trends Mach Learn* 2009; 2(1): 1–127.
24. Kessy, Agnan and Lewin, Alex and Strimmer, Korbinian. Optimal whitening and decorrelation. *Am Stat* 2018; pp. 1–6.
25. Bengio Y. Practical recommendations for gradient-based training of deep architectures. In: Montavon G, Orr G and Müller K (eds) *Neural networks: tricks of the trade (Lecture Notes in Computer Science)*. New York: Springer, 2012, pp. 437–478.
26. Joyce JM. Kullback-Leibler divergence. In: Lovric M (ed.) *International encyclopedia of statistical science*. New York: Springer, 2011, pp. 720–722.
27. Møller MF. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 1993; 6(4): 525–533.
28. Li J and Hao H. Substructure damage identification based on wavelet domain response reconstruction. *Struct Health Monit* 2014; 13(4): 389–405.
29. Xia Y, Hao H, Brownjohn JMW, et al. Damage identification of structures with uncertain frequency and mode shape data. *Earthq Eng Struct D* 2002; 31(5): 1053–1066.
30. Lam HF and Ng CT. The selection of pattern features for structural damage detection using an extended Bayesian ANN algorithm. *Eng Struct* 2008; 30(10): 2762–2770.
31. Ng CT. Application of Bayesian-designed artificial neural networks in phase II structural health monitoring benchmark studies. *Aust J Struct Eng* 2014; 15(1): 27–36.
32. Li J, Law SS and Hao H. Improved damage identification in a bridge structure subject to moving vehicular loads: numerical and experimental studies. *Int J Mech Sci* 2013; 74: 99–111.