Impact identification using nonlinear dimensionality reduction and supervised learning

Article *in* Smart Materials and Structures · September 2019

DOI: 10.1088/1361-665X/ab419e

4 authors, including:

V. Meruane
University of Chile
75 PUBLICATIONS **500** CITATIONS
SEE PROFILE

Enrique López Droguett
University of Maryland, College Park
160 PUBLICATIONS **884** CITATIONS
SEE PROFILE

Alejandro Ortiz-Bernardin
University of Chile
37 PUBLICATIONS **239** CITATIONS
SEE PROFILE

Impact identification using nonlinear dimensionality reduction and supervised learning

V. Meruane [1,2*], C. Espinoza[1], E. Lopez-Droguett[1], and A. Ortiz-Bernardin[1]

[1] Department of Mechanical Engineering, Universidad de Chile, Beauchef 851, Santiago, Chile
[2] Millennium Nucleus on Smart Soft Mechanical Metamaterials, Beauchef 851, Santiago, Chile
* Corresponding author. Email: vmeruane@uchile.cl

Received xxxxxx
Accepted for publication xxxxxx
Published xxxxxx

Abstract

Real-time monitoring systems that can automatically locate and identify impacts as they occur have become increasingly attractive for ensuring safety and preventing catastrophic accidents in airspace structures. In most cases, a set of piezoelectric transducers distributed over the structure captures strain–time data, which are preprocessed to extract relevant features that are fed to a supervised learning algorithm to detect, locate, and quantify impacts. The best results achieved to date in feature extraction for impact identification have been obtained with the use of principal component analysis (PCA). However, this technique cannot handle complex nonlinear data. The primary contribution of this study is the implementation of a novel impact identification algorithm that uses a supervised learning algorithm called linear approximation with maximum entropy (LME) in conjunction with different linear and nonlinear dimensionality reduction techniques, including PCA, kernel PCA, Isomap, local linear embedding (LLE), and multilayer autoencoders. The performance of LME with the different reduction techniques is tested with two experimental applications. The results show that the techniques that do not employ graphs, such as PCA, kernel PCA, and autoencoders, perform better, and the method that provides the best results is LME in conjunction with autoencoders. It is further demonstrated that LME with autoencoders works better than the algorithms available in the literature for similar problems.

Keywords: impact identification, nonlinear dimensionality reduction techniques, linear approximation with maximum entropy, autoencoders

1. Introduction

Impacts caused by foreign objects during manufacturing, operation, or maintenance are a threat to airspace structures because their properties can be significantly degraded by impact-induced damage. Therefore, real-time monitoring systems that can automatically locate and identify impacts as they occur have become increasingly attractive for ensuring safety and preventing catastrophic accidents. These systems usually work with a network of sensors built into the structure, whose information is processed by an artificial intelligence algorithm.

Recently, various impact identification systems have been developed with artificial neural networks (ANNs) being the most popular [1–5]. However, ANNs are susceptible to overfitting and they are often stuck in local minima. Alternatively, least-squares support-vector machines (LS-SVMs) solve convex linear optimization problems, which have a single minimum [6]. Another option is an extreme learning machine (ELM), which is much faster and provides

good generalization. Xu [7] showed that, for impact identification, kernel-ELM provides similar results to those from LS-SVMs while requiring less computational time. The time-reversal approach that measures the correlation between two signals by their convolution has also been applied in impact localization [8–11]. In [12], Sanchez and Meruane demonstrated that, for impact identification, the linear approximation with maximum entropy (LME) performs better than ANN and LS-SVM.

In impact identification, a set of piezoelectric transducers distributed over the structure captures strain–time data, which is preprocessed to extract relevant features that are fed to a supervised learning algorithm to detect, locate, and quantify impacts. The features can be manually extracted by selecting the maximum amplitude or the time of arrival, among other parameters [12]. Another strategy is to use techniques capable of finding a set of representative parameters by themselves, such as principal component analysis (PCA). Fu et al. [13, 14] and Meruane et al. [15] demonstrated that PCA significantly improves the impact localization results. PCA is a linear dimensionality reduction technique that facilitates the compression of high-dimensional data. However, this technique cannot handle complex nonlinear data.

In recent years, many nonlinear dimensionality reduction techniques have been proposed [16] that perform better in the cases of real data with non-linear manifolds. Kernel PCA is a nonlinear extension of PCA that projects the data in a higher-dimensional feature with the use of a kernel function [17]. This allows the formation of nonlinear mappings. Classical PCA scaling retains pairwise Euclidean distances, which does not take into account the distribution of adjacent data points. Isomap [18] resolves this problem by considering the pairwise geodesic distance, which is the distance between a pair of points measured over the manifold. Local linear embedding (LLE) [19], on the other hand, maps non-convex manifolds by considering local data properties. Some other nonlinear feature-reduction techniques include Hessian LLE [20], Laplacian eigenmaps [21], and local tangent space alignment [22], among others.

More recently, deep learning has been used successfully in nonlinear dimensionality reduction thanks to the good performance of multilayer autoencoders [23]. Multilayer autoencoders are neural networks that are trained to reconstruct high-dimensional input vectors. The central layer of a multilayer autoencoder has a small number of neurons, thus forcing the network to learn a compressed representation of the input data. Unlike nonparametric methods (such as kernel PCA, Isomap, and LLE), autoencoders can embed new high-dimensional data into the existing low-dimensional representation without errors, and they can be applied to very large data sets. Maaten et al. [24] investigated the performance of twelve nonlinear dimensionality reduction techniques in artificial and natural datasets and showed that, in the case of natural datasets, PCA and autoencoders outperformed the other techniques.

The primary contribution of this study is the implementation of a novel impact identification algorithm that uses LME in conjunction with different dimensionality reduction techniques, including PCA, kernel PCA, Isomap, LLE, and multilayer autoencoders. The results show that the techniques that do not employ graphs, such as PCA, kernel PCA, and autoencoders, perform better, and the method that provides the best results is LME in conjunction with autoencoders. It is further demonstrated that LME with autoencoders works better than the algorithms available in the literature for similar problems.

The remainder of this paper is structured as follows. Section 2 describes the different dimensionality reduction techniques implemented. Section 3 briefly describes the LME algorithm. The methodology followed in this study is presented in Section 4. Section 5 describes the experimental applications and results. Section 6 compares the results of LME with autoencoders with those of other impact identification algorithms applied in similar problems. Finally, conclusions and recommendations for future research are presented in Section 7.

2. Dimensionality reduction

Let us assume we count with a dataset represented by a $P \times q$ matrix $\boldsymbol{Z}$ that consists of q vectors z_i with dimension $P \times 1$. In this case, each vector corresponds to the envelope of the strain–time response given by a piezoelectric sensor bonded to the structure's surface. This response captures the stress waves generated by an impact force over the structure. Figure 1 presents an example strain–time signal envelope obtained from an impact test. The envelope was computed using a Hilbert transform.

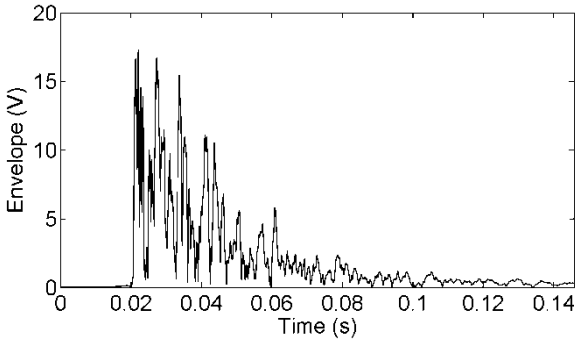


Figure 1 Envelope of a strain–time signal obtained from an impact test.

The dataset represented by $\boldsymbol{Z}$ has an intrinsic dimension p, with $p < P$. Dimensionality reduction techniques transform the dataset $\boldsymbol{Z}$ into a new dataset $\boldsymbol{S}$ with dimension $p \times q$. In most cases, the intrinsic dimension p is unknown, and therefore, its value must be assumed. The following reduction techniques for impact identification are described in this section: (1) PCA, (2) kernel PCA, (3) Isomap, (4) LLE, and (5) multilayer autoencoders.

2.1 PCA

Principal component analysis (PCA) reduces the dimensions of the dataset while maximizing preservation of its variation. PCA projects the data into a new orthogonal coordinate system, yielding a set of uncorrelated features identified as principal components (PCs). Then the sets of PCs with greater contributions to the variance are selected.

Therefore, the objective is to find a matrix $\boldsymbol{P} \in \mathbb{R}^{p\times P}$ that provides a linear mapping from the initial dimension P to a reduced dimension p. A new matrix $\boldsymbol{S} \in \mathbb{R}^{p\times q}$, called the score matrix, is obtained from

$$\boldsymbol{S} = \boldsymbol{PZ}. \tag{1}$$

The matrix $\boldsymbol{P}$ contains the main p eigenvectors of the covariance matrix of $\boldsymbol{Z}$, which is defined by

$$\boldsymbol{C} = \frac{1}{p-1}\boldsymbol{ZZ^T}. \tag{2}$$

2.2 Kernel PCA

Kernel PCA is a nonlinear extension of PCA that projects the data in a higher-dimensional feature with the use of a kernel function [17]. This allows the formation of nonlinear mappings.

In Kernel PCA, a nonlinear transformation $\phi(\boldsymbol{z}_i)$ from the initial dimension P to a reduced dimension p is defined. We define $\boldsymbol{z}_i$ as the i-th column of the matrix $\boldsymbol{Z}$, and the covariance matrix of the projected features is then calculated as follows:

$$\overline{\boldsymbol{C}} = \frac{1}{q}\sum_{i=1}^{q}\phi(\boldsymbol{z}_i)\phi(\boldsymbol{z}_i)^T. \tag{3}$$

The eigenvalues $\lambda > 0$ and eigenvector $\boldsymbol{V}$ of the covariance matrix satisfy $\lambda\boldsymbol{V} = \overline{\boldsymbol{C}}\boldsymbol{V}$, which be written equivalently as

$$\lambda\phi(\boldsymbol{z}_i)\boldsymbol{V} = \phi(\boldsymbol{z}_i)\overline{\boldsymbol{C}}\boldsymbol{V} \tag{4}$$

The solutions for $\boldsymbol{V}$ lie in the span $\phi(\boldsymbol{z}_1), \dots, \phi(\boldsymbol{z}_q)$. Therefore, there exist coefficients $\alpha_1, \dots, \alpha_q$, such that

$$\boldsymbol{V} = \sum_{i=1}^{q}\alpha_i\,\phi(\boldsymbol{z}_i). \tag{5}$$

Substituting Equations (3) and (5) into Equation (4), we obtain

$$\boldsymbol{K^2\alpha} = q\lambda\boldsymbol{K\alpha}, \tag{6}$$

where $\boldsymbol{\alpha}$ denotes the column vector with entries $\alpha_1, \dots, \alpha_q$, and the matrix kernel $\boldsymbol{K}$ is defined as

$$K_{ij} = \kappa(\boldsymbol{z}_i, \boldsymbol{z}_j) = \phi(\boldsymbol{z}_i)^T\phi(\boldsymbol{z}_j). \tag{7}$$

The vector $\boldsymbol{\alpha}$ can be solved by

$$\boldsymbol{K\alpha} = q\lambda\boldsymbol{\alpha}. \tag{8}$$

To extract the principal components, the dataset is projected onto the matrix $\widetilde{\boldsymbol{V}}$ that contains the main p eigenvectors according to

$$\boldsymbol{S} = \boldsymbol{K}(\boldsymbol{Z})\widetilde{\boldsymbol{V}}. \tag{9}$$

A commonly used kernel function is the Gaussian kernel with the parameter σ[25]:

$$\kappa(\boldsymbol{z}_i, \boldsymbol{z}_j) = exp\left(-\left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\|^2 \Big/ 2\sigma^2\right). \tag{10}$$

2.3 Isomap

Classical PCA scaling retains pairwise Euclidean distances, which does not take into account the distribution of adjacent data points. If high-dimensional data lies on a curved manifold, classical scaling may interpret two data points as being close together, whereas their distance over the manifold is longer. Isomap [18] resolves this problem by considering the pairwise geodesic distance, which is the distance between a pair of points measured over the manifold

First, a neighborhood graph G, where every data point is connected with its h-nearest neighbors is constructed. Then, all of the shortest paths in G are computed using an estimate of the geodesic distance, yielding a distance matrix $\widetilde{\boldsymbol{D}}$. The entries of the Gram matrix, $\boldsymbol{K}$, are obtained by double-centering the distance matrix as,

$$K_{ij} = -\frac{1}{2}\left(\widetilde{D}_{ij}^2 - \frac{1}{q}\sum_l \widetilde{D}_{il} - \frac{1}{q}\sum_l \widetilde{D}_{il} - \frac{1}{q^2}\sum_{lm}\widetilde{D}_{lm}^2\right). \tag{11}$$

Let λ_i and v_i be the i-th eigenvalue and eigenvector (in decreasing order) of the matrix $\boldsymbol{K}$, respectively. The i-th reduced dimension vector is equal to $\boldsymbol{s}_i = \sqrt{\lambda_i}\,v_i$.

2.4 LLE

Similar to Isomap, local linear embedding (LLE) [19] also builds a graph of the data points. Nevertheless, in this case, the global linear structure is approximated by local linear regressions, thus avoiding the computation of pairwise distances between widely separated data points. The local regression is constructed by defining each data point as a linear combination of its *h*-nearest neighbors:

$$\boldsymbol{z}_i = \sum w_{ij}\, \boldsymbol{z}_j \,. \tag{12}$$

Considering that the manifold is locally linear, we can assume that the weights w_{ij} used to reconstruct the data point $\boldsymbol{z}_i$ from its neighbors in the high-dimensional space likewise reconstruct the data point $\boldsymbol{s}_i$ from its neighbors in the low-dimensional space. Therefore, the embedding of each high-dimensional vector $\boldsymbol{z}_i$ into the low-dimensional vector $\boldsymbol{s}_i$ is achieved by minimizing the following function:

$$\Phi(\boldsymbol{s}) = \sum_i \left|\boldsymbol{s}_i - \sum w_{ij}\, \boldsymbol{s}_j\right|^2. \tag{13}$$

2.5 Multilayer autoencoders

Multilayer autoencoders are feed-forward neural networks with an odd number of hidden layers, which are trained to reconstruct high-dimensional input vectors. The central layer of a multilayer autoencoder has a small number of neurons, thus forcing the network to learn a compressed representation of the input data [23].

An example of a multilayer autoencoder is presented in Figure 2. The network is trained to minimize the mean square error between the network's input and output, thus attempting to equalize the network output and input. Training the network on the data points $\boldsymbol{z}_i$ yields a network in which the middle layer gives a low-dimensional representation of the datapoints that preserves, as much as possible, the structure of $\boldsymbol{z}_i$. Therefore, the low-dimensional representation $\boldsymbol{s}_i$ is obtained by extracting the output of the middle hidden layer when $\boldsymbol{z}_i$ is given as an input. To allow the encoder to make nonlinear mappings, nonlinear activation functions, such as sigmoid, are generally used.

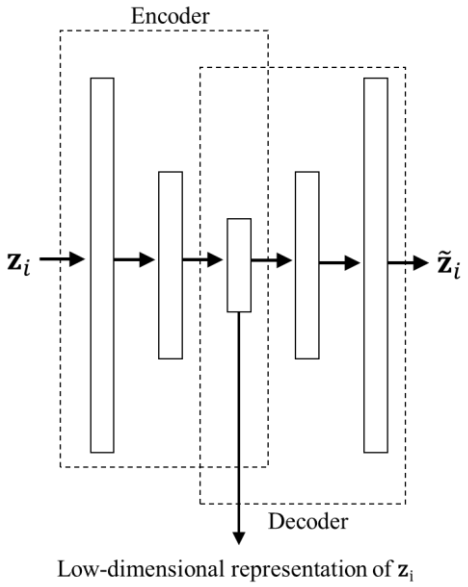


Figure 2 Schematic structure of a multilayer autoencoder.

3. Linear Approximation with Maximum Entropy (LME)

The database is expressed as pairs of observation and feature vectors as: $(\boldsymbol{X}^1, \boldsymbol{Y}^1)$, …, $(\boldsymbol{X}^N, \boldsymbol{Y}^N)$. The feature vector $\boldsymbol{X}^j$ is obtained from the *j*-th column of **S** and represents a set of features associated with the *j*-th observation vector $\boldsymbol{Y}^j$. On the other hand, the observation vector, $\boldsymbol{Y}^j = \{Y_1^j, Y_2^j, Y_3^j\} \in \mathbb{R}^3$, contains information related to the impact location and magnitude, where Y_1^j, Y_2^j are the *x* and *y* coordinates of the force location and Y_3^j is the force magnitude. We assume that we measure the response to an impact and extract its features as the vector $\boldsymbol{X}$. Then, the problem is to provide an estimation of the force magnitude and location as given by the corresponding observation $\boldsymbol{Y}$. The nearest neighbor estimation of $\boldsymbol{Y}$ is given by

$$\widehat{\boldsymbol{Y}} = \sum_{j=1}^{k} w_j\,(\boldsymbol{X})\boldsymbol{Y}^j, \tag{14}$$

where $\boldsymbol{Y}^1$, $\boldsymbol{Y}^2$, …, $\boldsymbol{Y}^k$ are the observation vectors associated with the *k*-closest neighbors of the feature vector $\boldsymbol{X}$, and $w_1(\boldsymbol{X})$, $w_2(\boldsymbol{X})$, …, $w_k(\boldsymbol{X})$ are weighting functions. One alternative is to weight each neighbor equally, as the *k*-nearest neighbor (*k*-NN) algorithm does. Another alternative is to make the weights proportional to the distance from the test vector $\boldsymbol{X}$ to each vector in the database [26]. In linear approximation, $\boldsymbol{X}$ is represented by a linear combination of its nearest neighbors:

$$\boldsymbol{X} = \sum_{j=1}^{N} w_j\,(\boldsymbol{X})\boldsymbol{X}^j, \quad \sum_{j=1}^{N} w_j\,(\boldsymbol{X}) = 1, \tag{15}$$

where $\boldsymbol{X}^1$, $\boldsymbol{X}^2$, …, $\boldsymbol{X}^N$ are the *N* nearest neighbors of $\boldsymbol{X}$ in the training database. Once $w_1(\boldsymbol{X})$, $w_2(\boldsymbol{X})$, …, $w_k(\boldsymbol{X})$ are computed, $\boldsymbol{Y}$ is estimated from Equation (14) with $k = N$.

Equation (15) can be solved through least-squares, but the solution yields some negative weights, which have no physical meaning. Here, the weights are obtained via the maximum-entropy (max-ent) variational principle, which ensures positive solutions [15]; the optimization problem is defined as follows:

$$\max_{\boldsymbol{w}} \left[H(\boldsymbol{w}) = -\sum_{j=1}^{N} w_j(\boldsymbol{X}) \, ln\left(\frac{w_j(\boldsymbol{X})}{m_j(\boldsymbol{X})}\right)\right] \quad (16)$$

subject to the constraints

$$\sum_{j=1}^{N} w_j(\boldsymbol{X})\widetilde{\boldsymbol{X}}^j = 0, \quad \sum_{j=1}^{N} w_j(\boldsymbol{X}) = 1, \quad (17)$$

where $\widetilde{\boldsymbol{X}}^j = \boldsymbol{X}^j - \boldsymbol{X}$ and $m_j(\boldsymbol{X})$ is a prior distribution. For impact identification, the best performance is obtained by using the smooth Gaussian prior function [12],

$$m_j(\boldsymbol{X}) = exp(-\beta_j \left\| \widetilde{\boldsymbol{X}}^j \right\|^2), \quad (18)$$

where $\boldsymbol{\beta_j} = \boldsymbol{\gamma}/\boldsymbol{h_j^2}$; $\boldsymbol{\gamma}$ is a parameter that controls the support of the Gaussian prior at $\boldsymbol{X^j}$ and therefore its associated to the number of neighbors that contribute to the solution; and $\boldsymbol{h_j}$ is a characteristic *n*-dimensional Euclidean distance between neighbors that can be distinct for each $\boldsymbol{X^j}$.

4. Methodology

The general methodology consists of three main parts: building of the datasets, selection of parameters, and evaluation of the impact identification methods.

4.1 Building of the datasets

Three sets of impact data are acquired: one for training, one to set up the parameters of the identification algorithm, and one to evaluate the algorithm. These databases are referred to as training, testing, and evaluation sets, respectively. In the three cases, each location in the structure is impacted once using an instrumented impact hammer. The structure is linear, and consequently, the response is proportional to the magnitude of the force. Therefore, the responses to impacts of different magnitudes can be determined by simply multiplying the measured response by scaling factors. By using this methodology, the initial training set is expanded to a new set with impacts of magnitudes between 5 and 250 N.

The information related to the impact's location and magnitude is stored in observation vectors and the envelope of the sensor's time response is normalized and stored in the matrices $\boldsymbol{Z}^i_{train} \in \mathbb{R}^{Pxq_1}$, $\boldsymbol{Z}^i_{test} \in \mathbb{R}^{Pxq_2}$ and $\boldsymbol{Z}^i_{eval} \in \mathbb{R}^{Pxq_3}$, where i is the sensor number, P is the number of datapoints in the time response, and q_1, q_2 and q_3 are the number of elements in the training, testing, and evaluation sets, respectively. Using the dimensionality reduction techniques described in section 2, the matrices are transformed to $\boldsymbol{S}^i_{train} \in \mathbb{R}^{pxq_2}$, $\boldsymbol{S}^i_{test} \in \mathbb{R}^{pxq_2}$, and $\boldsymbol{S}^i_{eval} \in \mathbb{R}^{pxq_2}$, with $p < P$. Lastly, the matrices of all sensors are assembled together:

$$\boldsymbol{S}_{train} = \begin{Bmatrix} \boldsymbol{S}^1_{train} \\ \boldsymbol{S}^2_{train} \\ \vdots \\ \boldsymbol{S}^r_{train} \end{Bmatrix}, \boldsymbol{S}_{test} = \begin{Bmatrix} \boldsymbol{S}^1_{test} \\ \boldsymbol{S}^2_{test} \\ \vdots \\ \boldsymbol{S}^r_{test} \end{Bmatrix}, \boldsymbol{S}_{eval} = \begin{Bmatrix} \boldsymbol{S}^1_{eval} \\ \boldsymbol{S}^2_{eval} \\ \vdots \\ \boldsymbol{S}^r_{eval} \end{Bmatrix}, \quad (19)$$

where r is the number of sensors.

4.2 Selection of parameters

The first parameter to be defined is the intrinsic dimension p, which is defined using PCA by selecting the number of eigenvalues that retain a cumulative percentage variance of 99.99%. Depending on the dimensionality reduction technique, other parameters and functions must be selected, as listed in Table 1.

Table 1 Parameters and functions associated with the dimensionality reduction techniques

Method	**Parameters**
PCA	None
Kernel PCA	σ (Gaussian kernel)
Isomap	*h*
LLE	*h*
Multilayer autoencoders	Number of hidden layers, activation functions, regularization, and sparsity.

Additionally, other parameters that must be selected are the number of time steps in the time response signal and the number of neighbors, *k*, in the LME algorithm. All the parameters are defined to optimize the impact identification algorithm performance, as quantified by the following error functions:

$$E_x = \frac{1}{n_t}\sum_{j=1}^{n_t}\left|\hat{Y}_1^j - Y_1^j\right|, \quad (20)$$

$$E_y = \frac{1}{n_t}\sum_{j=1}^{n_t}\left|\hat{Y}_2^j - Y_2^j\right|, \quad (21)$$

$$E_F = \frac{1}{n_t}\sum_{j=1}^{n_t}\frac{\left|\hat{Y}_3^j - Y_3^j\right|}{Y_3^j} \times 100, \quad (22)$$

$$E_A = \frac{E_x \times E_y}{A} \times 100 = \frac{\sum_{j=1}^{n_t}\left|\hat{Y}_1^j - Y_1^j\right|\sum_{j=1}^{n_t}\left|\hat{Y}_2^j - Y_2^j\right|}{n_t^2 A} \times 100, \quad (23)$$

where n_t is the number of elements in the database; A is the area of the plate; E_x and E_y are the mean errors in the force estimation in the x and y coordinates; E_F is the percentage error in the estimation of the force magnitude, and E_A is the percentage area localization error. The impact identification error is defined as

$$E_I = E_A \times E_F. \quad (24)$$

4.3 Evaluation of the impact identification algorithm

The last step is to evaluate the performance of the algorithm using the testing database, which consists of the following steps:

1. Extract a feature vector from the testing database $\boldsymbol{S}_{test}$.
2. Select β_j in Equation (18) so that k neighbors contribute to the solution.
3. Compute the vector weight functions $\boldsymbol{w}$ as the solution of the optimization problem given in Equations (16) and (17).
4. Read the observation vectors in the database and estimate the experimental impact using Equation (14).
5. Compute the force, area, and impact identification errors using Equations (22), (23), and (24).
6. Repeat steps 1 to 5 for all the feature vectors in the testing database.

5. Experimental application

5.1 Aluminum sandwich panel

The structure was tested by Meruane et al. [15] and consists of an aluminum sandwich panel of 700 mm × 400 mm × 24 mm. The core consists of triangular stiffeners that cross the panel in the two principal directions. The thickness of the skins is 2 mm and the thickness of the stiffeners is 1 mm. Figure 3 shows the internal structure and the assembled panel.

The experimental setup is shown in Figure 4. The panel has clamped–free–clamped–free (CFCF) boundary conditions and is excited by an instrumented impact hammer. The response is captured by six piezoelectric discs bonded to the surface of the structure.

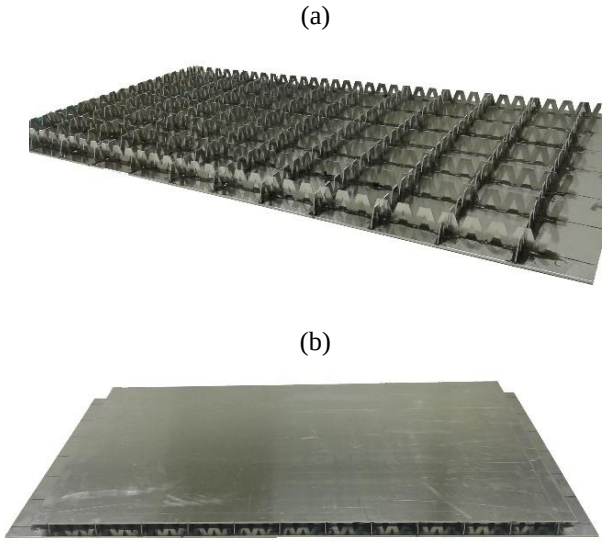
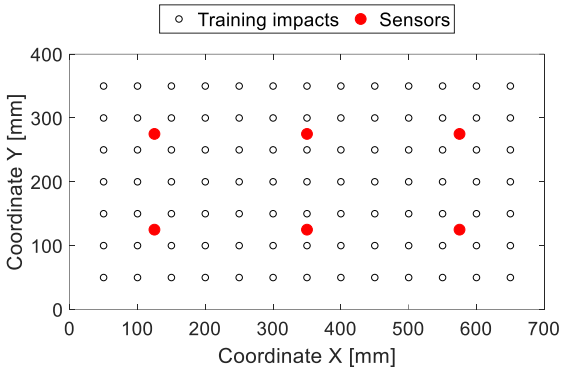


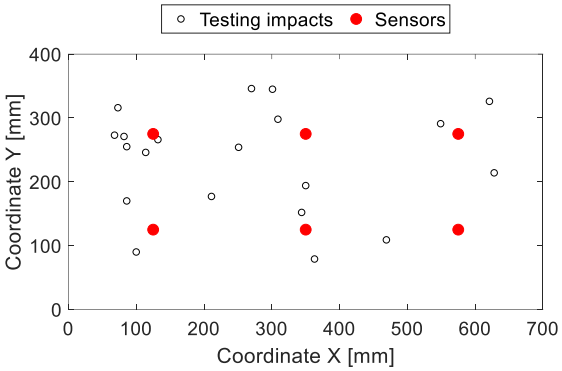
Figure 3 Aluminum sandwich panel. (a) Internal structure; (b) panel.

Figure 4 Experimental setup for the aluminum sandwich panel.

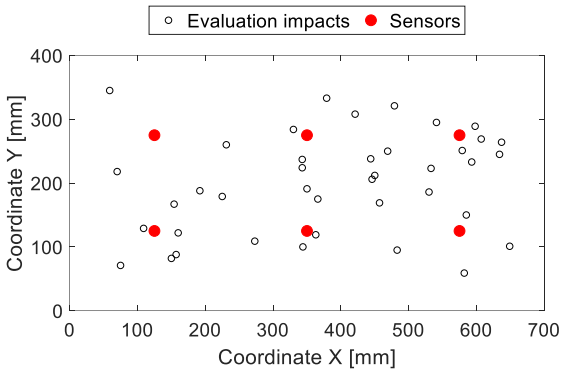
Data from the six piezoelectric sensors and impact hammer are recorded with a sampling rate of 24 kHz. The hammer is used as a trigger, and 20 data points before the impact and 3500 data points after the impact are recorded. The training set consists of a uniform grid of 91 points, as shown in Figure 5(a). The testing and evaluation sets consist of 20 and 40 random impacts distributed over the panel, as shown in Figure 5(b), and (c), respectively.



(a)



(b)



(c)

Figure 5 Location of the experimental impacts applied to the aluminum sandwich panel. (a) Training; (b) Testing; (c) Evaluation.

The first parameters to be defined are the number of time steps, P, and the intrinsic dimension, p. First, PCA is implemented as a dimensionality reduction technique with a different number of time steps in the response. The intrinsic dimension is defined by selecting the number of eigenvalues that retain a cumulative percentage variance of 99.99% and the results are presented in Table 2. Next, the performance of the impact identification algorithm is evaluated using PCA for the different number of time steps. The results of the impact identification algorithm depend on the number of neighbors, k, that contribute to the solution. Therefore, the performance is evaluated as a function of k, as shown in Figure 6. The performance does not change significantly with more than 800 time steps and the best results are obtained with 1600 time steps. Consequently, the selected parameters are $P = 1600$ and $p = 84$.

The testing set is used for all parameter definitions, whereas the evaluation set is used to evaluate the final algorithm performance.

Table 2 Intrinsic dimensions obtained for different numbers of time steps.

Number of time steps, *P*	**Intrinsic dimension, *p***
200	66
400	81
800	84
1600	84
3200	84

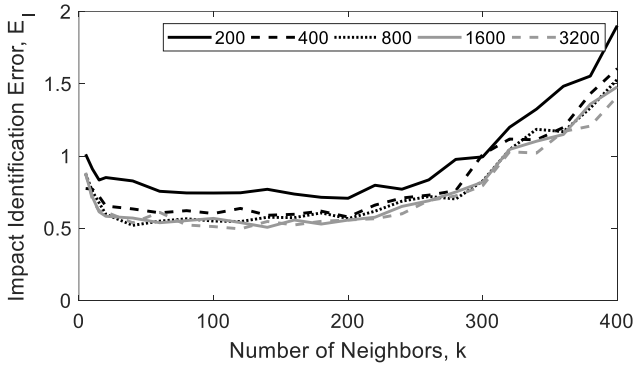


Figure 6 Performance of the impact identification algorithm with PCA as a function of k for different number of time steps.

For the multilayer autoencoder, the definition of the activation functions, regularization, and sparsity were performed to optimize the signal reconstruction using an autoencoder with one hidden layer. The final parameters are listed in Table 3, whereas Figure 7 shows an example of an input signal from the evaluation dataset and its autoencoder reconstruction. The signal is well reconstructed using only 84 parameters.

Table 1 Parameters of the optimized autoencoder.

Parameter or function	**Value**
Encoder transfer function	logsig
Decoder transfer function	purelin
L2 weight regularization	0.001
Sparsity regularization	1

Sparsity proportion	0.6

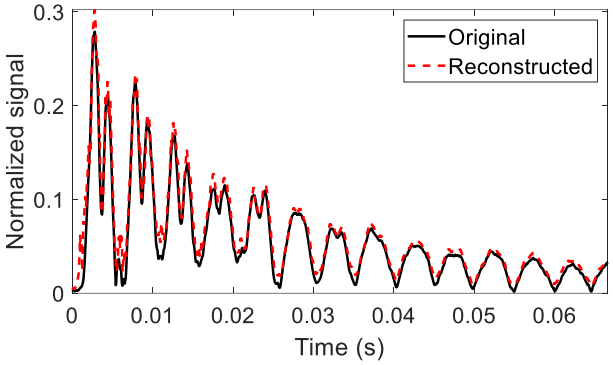


Figure 7 Original and reconstructed signal using an autoencoder for the aluminum sandwich panel.

The number of hidden layers in the autoencoder and the parameters associated with the other dimensionality reduction techniques, as shown in Table 4, are defined by optimizing the performance of the impact identification algorithm. Figure 8 illustrates the performance of the impact identification algorithm using all dimensionality reduction techniques as a function of the number of neighbors, *k*, that contribute to the solution. The best performance using each dimensionality reduction technique is summarized in Table 5. The best results overall are obtained with autoencoders, although kernel PCA yields a lower error in the force magnitude estimation. Figure 9 presents the impact identification results for the evaluation set using autoencoders and LME.

Table 2 Dimensionality reduction technique parameters.

Method	**Parameters**
Kernel PCA	σ = 3
Isomap	*h = 15*
LLE	*h = 5*
Multilayer autoencoders	Number of hidden layers = 1

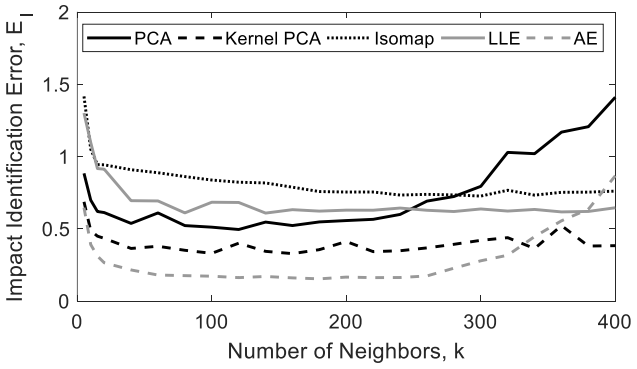
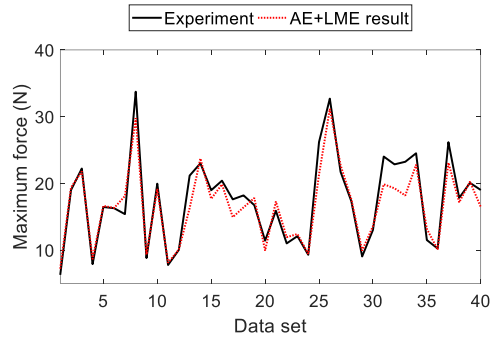


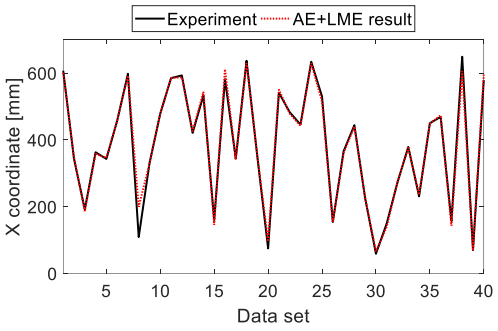
Figure 8 Performance of the impact identification algorithm using all dimensionality reduction techniques as a function of *k*.

Table 5 Best performance of the impact identification algorithm using the different dimensionality reduction techniques

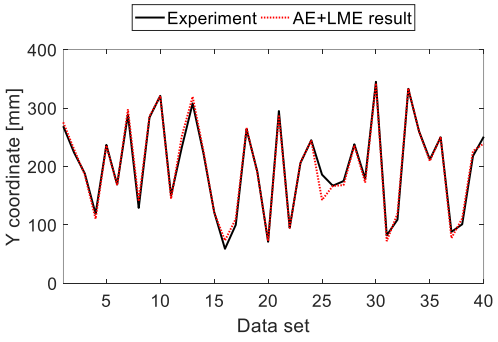
Method	E_A (%)	E_F (%)	E_I (%)
PCA	0.044	11.84	0.521
Kernel PCA	0.050	6.66	0.333
Isomap	0.058	12.50	0.725
LLE	0.054	11.39	0.615
AE	0.019	8.16	0.155



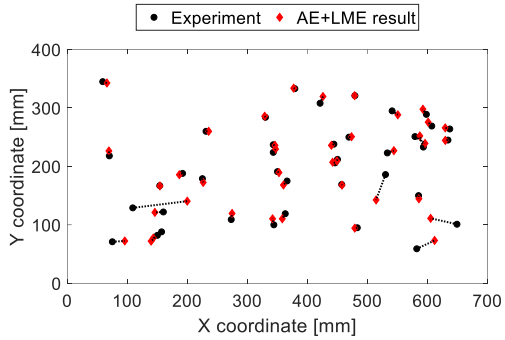
(a)



(b)



(c)



(d)

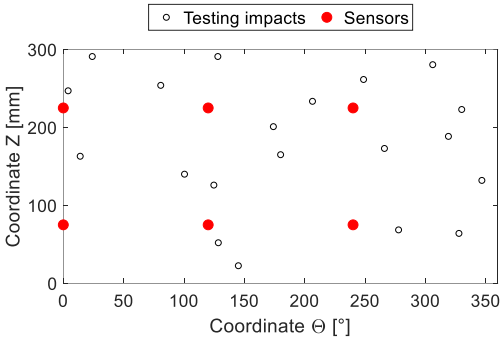
Figure 9 Evaluation results for the aluminum sandwich panel using the LME + AE algorithm. (a) Force amplitude; (b) X coordinate; (c) Y coordinate; (d) localization.

5.2 Steel cylinder

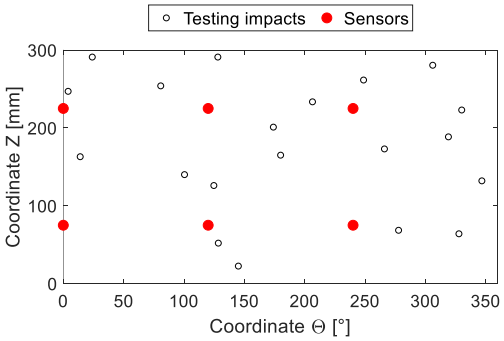
The structure consists of a steel cylinder with the diameter of 300 mm, height of 300 mm, and thickness of 5 mm. As shown in Figure 10, the cylinder is suspended by springs to simulate a free–free boundary condition. An instrumented hammer applies the impact forces and the response is captured by six piezoelectric discs bonded to the surface.

Figure 10 Experimental setup for the steel cylinder.

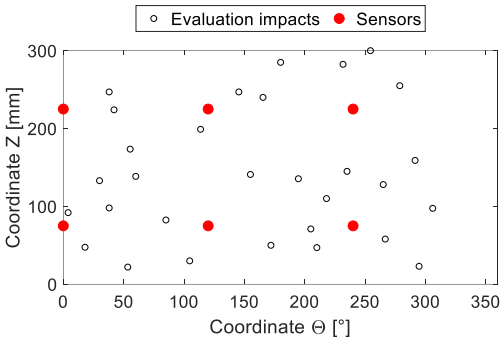
Data from the six piezoelectric sensors and impact hammer are recorded with a sampling rate of 25.6 kHz. The hammer is used as a trigger, and 20 data points before the impact and 3500 data points after the impact are recorded. The training set consists of a uniform grid of 91 points, as shown in Figure 11(a). Because the structure is a cylinder, a cylindrical coordinate system is used. The testing and evaluation sets consist of 20 and 30 random impacts distributed over the cylinder, as shown in Figure 11(b) and (c), respectively.



(a)



(b)



(c)

Figure 11 Location of the experimental impacts applied to the steel cylinder. (a) Training; (b) testing; (c) evaluation.

Similar to the aluminum sandwich panel case, the intrinsic dimension is defined by selecting the number of eigenvalues that retain a cumulative percentage variance of 99.99%. The results for the different number of time steps are presented in Table 6. Next, the performance of the impact identification algorithm for the different number of time steps is evaluated using PCA. The results of the impact identification algorithm depend on the number of neighbors, *k*, that contribute to the solution. Therefore, the performance is evaluated as a function of *k*, as shown in Figure 12. The performance does not change significantly with more than 800 time steps and the best results

are obtained with 3200 time steps. Consequently, the selected parameters are $P = 3200$ and $p = 79$.

Table 6 Intrinsic dimensions obtained for different numbers of time steps

Number of time steps, P	Intrinsic dimension, p
200	46
400	63
800	72
1600	77
3200	79

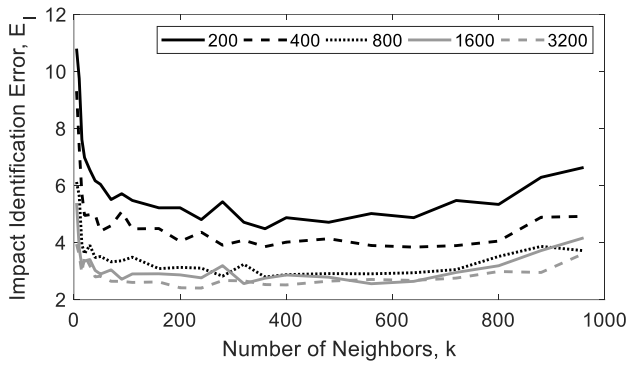


Figure 12 Performance of the impact identification algorithm with PCA as a function of k for different numbers of time steps.

The configuration of the autoencoder parameters and functions are the same as that listed in Table 3. An example of an input signal from the evaluation dataset and its autoencoder reconstruction is shown in Figure 13. The signal appears to be less dampened than in the case of the aluminum sandwich panel. This is because the structure is cylindrical and the waves travel around the cylinder and take longer to dampen themselves. However, the autoencoder still reconstructs the signal well using only 79 parameters.

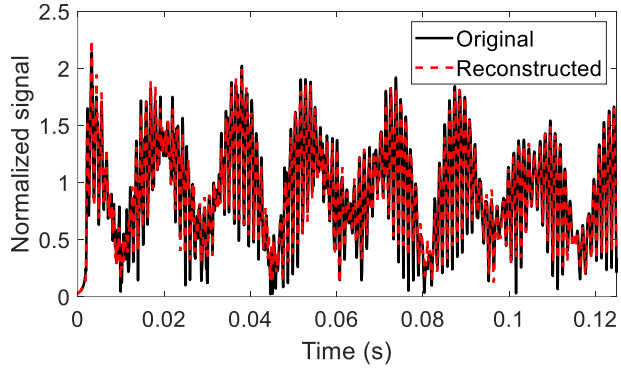


Figure 13 Original and reconstructed signal using an autoencoder for the steel cylinder.

The number of hidden layers in the autoencoder and the parameters associated with the other dimensionality reduction techniques are defined by optimizing the performance of the impact identification algorithm. Table 7 presents the final parameters and Figure 14 illustrates the performance of the impact identification algorithm using all dimensionality reduction techniques as a function of the number of neighbors, k, that contribute to the solution. As shown in Figure 14, the autoencoder produces the best results, but only for a small value of k. The best performance using each dimensionality reduction technique is summarized in Table 8. The best results overall are obtained with autoencoders, although PCA provides a lower error in the force magnitude estimation. Figure 15 presents the impact identification results for the evaluation set using autoencoders and LME.

Table 7 Dimensionality reduction technique parameters

Method	Parameters
Kernel PCA–Gaussian	$\sigma = 240$
Isomap	$h = 30$
LLE	$h = 5$
Multilayer autoencoders	Number of hidden layers = 1

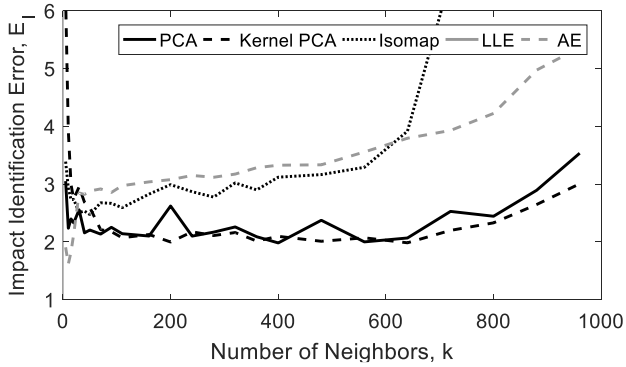
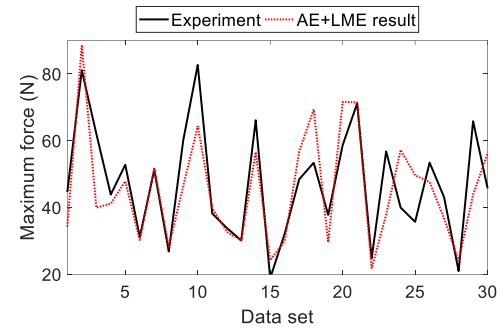


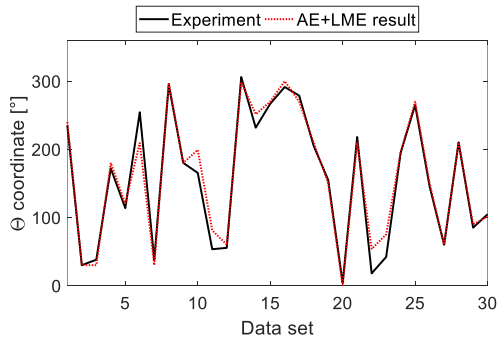
Figure 14 Performance of the impact identification algorithm using all dimensionality reduction techniques as a function of k.

Table 8 Best performance of the impact identification algorithm using the different dimensionality reduction techniques

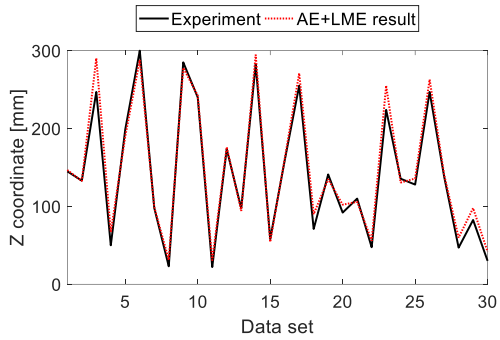
Method	E_A (%)	E_F (%)	E_I (%)
PCA	0.117	16.93	1.98
Kernel PCA	0.117	17.02	1.99
Isomap	0.132	18.77	2.47
LLE	0.847	29.18	24.70
AE	0.095	17.09	1.62



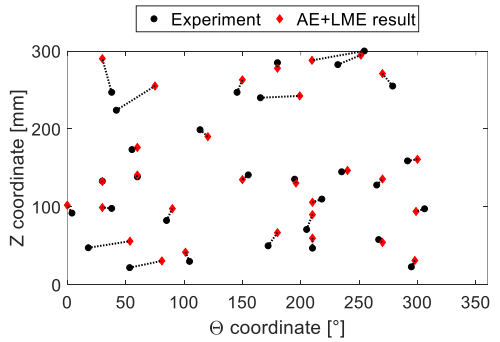
(a)



(b)



(c)



(d)

Figure 15 Evaluation results for the steel cylinder using the LME + AE algorithm. (a) Force amplitude; (b) θ coordinate; (c) Z coordinate; (d) Localization.

6. Discussion

The performance of the dimensionality reduction techniques varied for each application case, which implies that despite the similarity of the problems (i.e., impact identification), the dataset manifolds depend on the structure type. The techniques that do not employ graphs, such as PCA, kernel PCA, and autoencoders, performed better, and the dimension reduction technique that provided the best results was the autoencoder. These results agree with what was shown by Maaten et al. [24] for natural datasets. This may be because the construction of neighborhood graphs is susceptible to the problems of dimensionality, overfitting, and presence of outliers (see [24] for a detailed explanation). Most likely, the good performance of autoencoders in both applications is due to their ability to learn structures with highly varying manifolds.

Another possible reason for the autoencoder performance is its ability to embed new high-dimensional data points from the testing and evaluation sets into an existing low-dimensional representation. In PCA, this transformation is defined by the linear mapping P that was applied to the original training data. For autoencoders, the trained network defines the transformation from the high-dimensional to the low-dimensional data representation. However, for the other nonlinear techniques (kernel PCA, Isomap, and LLE) only nonparametric methodologies are available, which lead to estimation errors.

Table 9 presents the comparison of the results obtained for the aluminum sandwich panel using autoencoders and LME with those of other algorithms available in the literature. These algorithms were evaluated using similar structures of aluminum plates that were simply supported by four screws in [6], [7], [13], and [14] and an aluminum sandwich panel in [15]. The comparison results demonstrate that the proposed methodology has better precision.

Table 9 Comparison of impact identification algorithms available in the literature with those in the current work. SVM: support vector machine; LS-SVM: least-squares SVM; ELM: extreme learning machine; LME: linear approximation with maximum entropy.

Reference	Algorithm	Plate size (mm^2)	Number of sensors	Number of training impact points	Area error (%)	Force error (%)
Xu [6]	LS-SVM	Simple plate, 490 × 390	4	63	1.06	51.2
Fu and Xu [13]	PCA+SVM	Simple plate, 490 × 390	4	63	0.13	-
Xu [7]	Kernel-ELM	Simple plate, 490 × 390	4	63	0.74	-
Fu et al. [14]	PCA+ Kernel-ELM	Simple plate, 490 × 390	4	63	0.24	-
Sanchez et al. [12]	LME	Simple plate, 490 × 390	4	61	0.12	7.18
Meruane et al. [15]	PCA + LME	Sandwich plate, 700 × 400	6	91	0.031	12.39
Current work	AE + LME	Sandwich plate, 700 × 400	6	91	0.019	8.16

7. Conclusions

This study proposed the implementation of a novel impact identification algorithm that uses different nonlinear dimensionality reduction techniques in conjunction with the LME method. The algorithm performance was tested by considering two experimental cases of an aluminum sandwich panel and a steel cylinder. Time-varying strain data were measured using piezoceramic sensors bonded to the structures.

The results indicate that, for this application, the techniques that do not employ graphs, such as PCA, kernel PCA, and autoencoders, perform better. Autoencoders provide the best results, most likely because they can learn the structures of highly varying manifolds and embed new high-dimensional data points into the existing low-dimensional representation without errors.

The results of LME with autoencoders were compared with those of other impact identification algorithms available in literature. The results demonstrate the potential of LME with autoencoders over existing methods.

The experimental applications represent structures that can be used, for example, in aeronautical applications. Therefore, precise results can be obtained for the location and quantification of impacts in realistic structures. Nonetheless, it remains necessary to validate the performance in more complex structures.

Acknowledgements

The authors acknowledge the financial support provided by the Chilean National Fund for Scientific and Technological Development (FONDECYT) under Grant No. 1170535 and the Millennium Science Initiative of the Ministry of Economy, Development and Tourism Grant "Millennium Nucleus on Smart Soft Mechanical Metamaterials."

References

[1] Worden K, Staszewski WJ 2000 *Strain* **36** 61–68
[2] Staszewski WJ, Worden K, Wardle R, and Tomlinson GR 2000 *Smart Mater. Struct.* **9** 3 298–303
[3] Haywood J, Coverley PT, Staszewski WJ, and Worden K 2005 *Smart Mater. Struct.* **14** 1 265–271
[4] LeClerc JR, Worden K, Staszewski WJ, and Haywood J 2007 *J. Sound Vib.* **299** 3 672–682
[5] Sharif-Khodaei Z, Ghajari M, and Aliabadi MH 2012 *Smart Mater. Struct.* **21** 10 105026
[6] Xu Q 2014 *Struct. Heal. Monit.* **31** 1 5–18
[7] Xu Q 2014 *Math. Probl. Eng.* **2014** 906732
[8] Ing RK, Quieffin N, Catheline S, and Fink M 2005 *Appl. Phys. Lett.* **87** 20 204104
[9] Ribay G, Catheline S, Clorennec D, Ing RK, Quieffin N, and Fink M 2007 *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **54** 2 378–385
[10] Park B, Sohn H, Olson SE, DeSimio MP, Brown KS, and Derriso MM 2012 *Struct. Heal. Monit.* **11** 5 577–588
[11] Ciampa F and Meo M 2012 *Struct. Heal. Monit.* **11** 1 43–49
[12] Sanchez N, Meruane V, and Ortiz-Bernardin A 2016 *Smart Mater. Struct.* **25** 9 095050
[13] Fu H and Xu Q 2013 *Math. Probl. Eng.* **2013** 352149

[14] Fu H, Vong C-M, Wong P-K, and Yang Z 2014 *Neural Comput. Appl.* **27** 1 1–10
[15] Meruane V, Véliz P, Droguett EL, and Ortiz-Bernardin A 2017 *Entropy* **19** 4 137
[16] Gao L, Song J, Liu X, Shao J, Liu J, and Shao J 2017 *Multimed. Syst.* **23** 3 303–313
[17] Schölkopf B, Smola A, and Müller K-R 1997 *International Conference on Artificial Neural Networks* 583–588
[18] Tenenbaum JB, De Silva V, and Langford JC 2000 *Science* **290** 5500 2319–2323
[19] Roweis ST and Saul LK 2000 *Science* **290** 5500, 2323–2326
[20] Donoho DL and Grimes C 2003 *Proc. Natl. Acad. Sci.* **100** 10 5591–5596
[21] Belkin M and Niyogi P 2003 Advances in Neural Information Processing Systems 585–591
[22] Zhang Z and Zha H 2004 *SIAM J. Sci. Comput* **26** 1 313–338
[23] Hinton GE and Salakhutdinov RR 2006 *Science* **313** 5786, 504–507
[24] Van Der Maaten L, Postma E, and den Herik J 2009 *J. Mach. Learn. Res.* **10** 66–71
[25] Wang Q 2012 *arXiv:*1207.3538 [cs.CV]
[26] Devroye L, Gyorfi L, Krzyzak A, and Lugosi G 1994 *Ann. Stat.* **22** 3 1371–1385