Journal of Sound and Vibration 363 (2016) 584–599

Contents lists available at ScienceDirect

Journal of Sound and Vibration

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

Machine learning algorithms for damage detection: Kernel-based approaches

Adam Santos [a,*], Eloi Figueiredo [b], M.F.M. Silva [a], C.S. Sales [a], J.C.W.A. Costa [a]

[a] Applied Electromagnetism Laboratory, Universidade Federal do Pará, R. Augusto Corrêa, Guamá 01, Belém, 66075-110 Pará, Brazil
[b] Faculty of Engineering, Universidade Lusófona de Humanidades e Tecnologias, Campo Grande 376, 1749-024 Lisbon, Portugal

ARTICLE INFO

Article history:
Received 9 April 2015
Received in revised form
1 October 2015
Accepted 3 November 2015
Handling Editor: K. Shin
Available online 21 November 2015

Keywords:
Structural health monitoring
Damage detection
Kernel
Operational conditions
Environmental conditions

ABSTRACT

This paper presents four kernel-based algorithms for damage detection under varying operational and environmental conditions, namely based on one-class support vector machine, support vector data description, kernel principal component analysis and greedy kernel principal component analysis. Acceleration time-series from an array of accelerometers were obtained from a laboratory structure and used for performance comparison. The main contribution of this study is the applicability of the proposed algorithms for damage detection as well as the comparison of the classification performance between these algorithms and other four ones already considered as reliable approaches in the literature. All proposed algorithms revealed to have better classification performance than the previous ones.

1. Introduction

Civil structures such as buildings, roads, railways, bridges, tunnels and dams are present in every society, regardless of culture, geographical location or economical development. The safest and most durable structures are those that are well managed and maintained. Health monitoring plays an important role in management activities [1]. The massive data obtained from monitoring must be transformed in meaningful information to support the planning and designing maintenance activities, increase the safety, verify hypotheses, reduce uncertainty and to widen the knowledge and insight concerning the monitored structure.

Structural health monitoring (SHM) is certainly one of the most powerful tools for civil infrastructure management. The SHM process consists of permanent, continuous, periodic or periodically continuous acquisition of parameters by a sensor network, feature extraction and statistical modeling for feature classification to detect possible structural damages and to support the decision making [2,3]. Damage is traditionally defined as changes in the material and/or geometric properties of the structures, including variations in the boundary conditions and system connectivity, which adversely affect the system's current or future performance. In contrast, normal condition refers to data acquired under different operational and environmental variability when the structure is known to be undamaged [4,5].

In the feature extraction phase is imperative to derive damage-sensitive features correlated with the severity of damage present in monitored structures, minimizing false judgements in the classification phase. Nevertheless, in real-world SHM applications, operational and environmental effects can camouflage damage-related changes in the features as well as alter

* Corresponding author.
E-mail address: adamdreyton@ufpa.br (A. Santos).

http://dx.doi.org/10.1016/j.jsv.2015.11.008
0022-460X/

the correlation between the magnitude of the features and the damage level. Commonly, the more sensitive a feature is to damage, the more sensitive it is to changing in the operational and environmental conditions (e.g., temperature and wind speed). To overcome this impact, robust feature extraction procedures are usually required [6–8].

Statistical modeling for feature classification phase is concerned with the implementation of machine learning algorithms that analyze the distributions of the extracted features and generate a data model in an effort to determine the structural health of the system. The algorithms used in statistical modeling usually fall into the outlier detection category, i.e., unsupervised learning is applied when training data are only available from the normal condition of the structure [9,10]. The data normalization procedure is normally present in the data acquisition, feature extraction and statistical modeling phases of the SHM process. Herein, data normalization includes a wide range of steps for removing the effect of operational and environmental variations on the extracted features [11].

Kernel-based machine learning algorithms have been widely applied to detect damage in SHM applications [12–16]. These algorithms, mostly based on support vector machines (SVMs), have revealed high sensitivity and accuracy in the damage classification. Mita and Hagiwara proposed a method using the supervised SVM to detect local damages in a building structure with limited number of sensors [17]. This method has been extended in several studies. For instance, a hybrid technique (wavelet SVM) may be considered, where damage-sensitive features are extracted through the wavelet energy spectrum and classified using the SVM [18]. In turn, a combined methodology between symbolic data analysis and classification techniques (e.g., SVM) is developed for damage assessment [19]. And, finally, an approach for detecting damage on shear structures using the SVM and the first three natural frequencies of the translational modes is assumed [20]. However, these approaches have not been implemented to remove the operational and environmental effects aggregated in extracted features; rather, they have been used to classify directly the extracted features in a supervised way, i.e., when data from the undamaged and damaged conditions are available.

However, for most civil engineering infrastructure where SHM systems are applied, the unsupervised learning algorithms are often required because only data from the undamaged condition are available [21,22]. Therefore, in this paper, an autoregressive (AR) model is used to extract damage-sensitive features upon time-series measured from an array of accelerometers, when the structure operates in different structural state conditions. Then, four unsupervised kernel-based machine learning algorithms are adapted for data normalization and damage detection. Firstly, they model the effects of the operational and environmental variability on the extracted features. Secondly, each algorithm produces a scalar output as a damage indicator (DI), which should be nearly invariant when features are extracted from the normal condition. Finally, DIs from the feature vectors of the test data are classified through a threshold defined based on the 95 percent cut-off value over the training data. The implemented algorithms are based on one-class support vector machine (one-class SVM), support vector data description (SVDD), kernel principal component analysis (KPCA) and greedy KPCA (GKPCA).

The main contribution of this study is the applicability of the proposed kernel-based algorithms for damage detection as well as the comparison of the classification performance, between these kernel-based algorithms and other reliable algorithms described in the literature [9,23–26], such as the auto-associative neural network (AANN) [11,27], factor analysis (FA) [28], Mahalanobis squared distance (MSD) [29], and singular value decomposition (SVD) [30], on standard data sets from a laboratory three-story frame aluminum structure. The performance is evaluated through receiver operating characteristic (ROC) curves, which are a means of determining performance on the basis of Type I/Type II error trade-offs. In SHM, in the context of damage detection, a Type I error is a false-positive indication of damage and a Type II error is a false-negative indication of damage. Besides, other contributions attested by the authors are the following: the first-time adaptation of the KPCA and GKPCA algorithms for damage detection in the SHM field; and the combination between one-class SVM/SVDD and MSD for data normalization purposes in the SHM field, particularly in the statistical modeling for feature classification phase.

This paper is organized as follows. Section 2 gives an explanation about the AR model for feature extraction, a brief background about the supervised SVM with the kernel trick for efficient optimization and the methodology of the four kernel-based machine learning algorithms for feature classification. A description of the test bed structure, the simulated operational and environmental variability, and a summary of the data sets is provided in Section 3. In Section 4, a comparative study between kernel-based algorithms and alternative approaches is carried out using features extracted from time-series data sets measured with accelerometers deployed on the test bed structure. Finally, Section 5 highlights a summary and discussion of the implementation and analysis carried out in this paper.

2. Feature extraction and kernel-based machine learning algorithms

The complete methodology applied in this study is depicted in Fig. 1. Basically, AR models are fitted to time series from an array of accelerometers when the structure is in different structural state conditions and their parameters are used as damage-sensitive features. Then, a training matrix, **X**, is composed of undamaged state conditions and a test matrix is composed of undamaged and damaged state conditions. Next, an unsupervised machine learning algorithm is trained and its parameters are adjusted using feature vectors from the training matrix only. In the test phase, the machine learning algorithm will transform each input feature vector from the test matrix, **Z**, into a global DI; the DIs should be nearly invariant for feature vectors extracted from the normal condition, assuming that the test data have been obtained from operational and environmental conditions represented in the training data. Finally, the classification is performed using one-sided

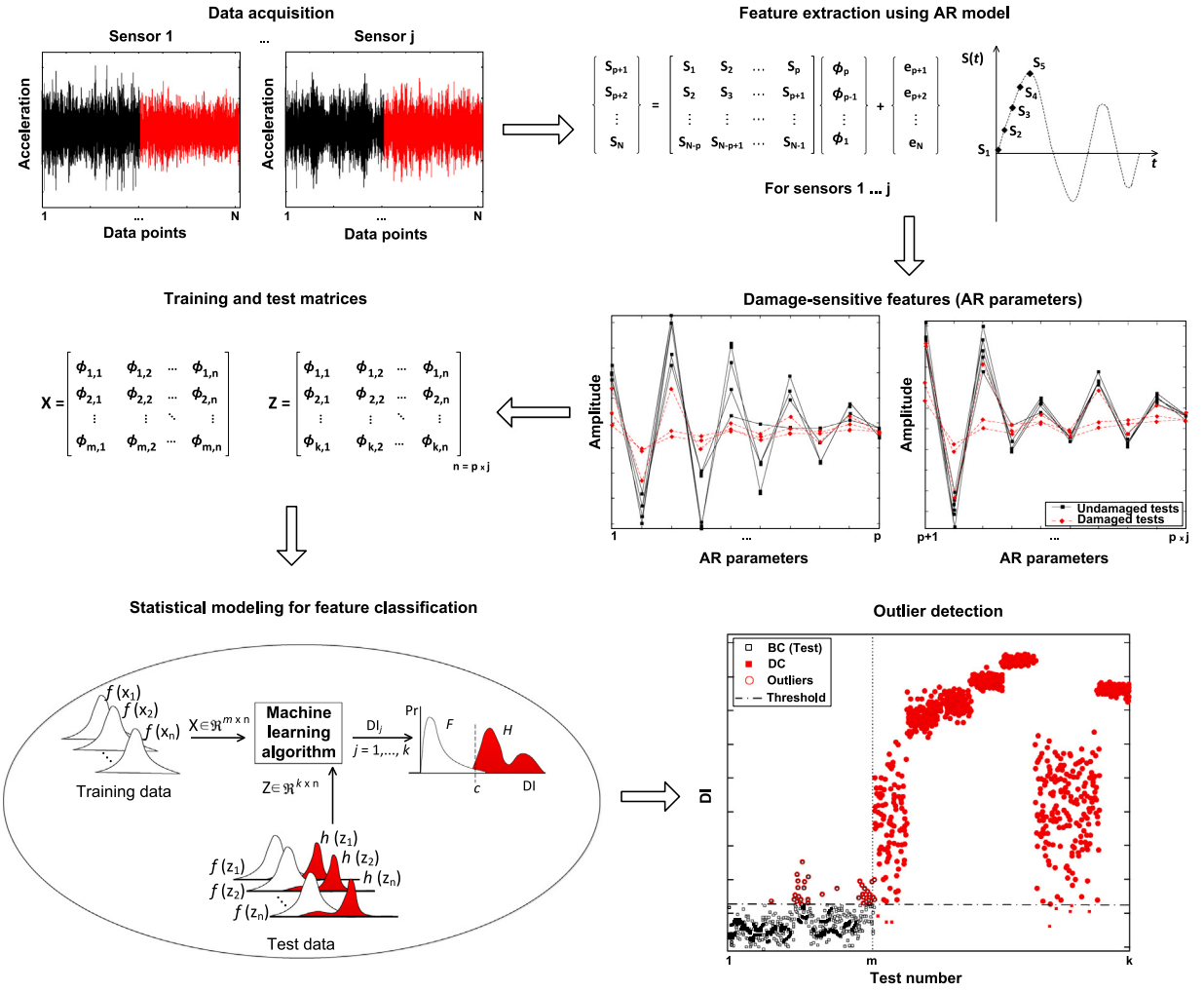

Fig. 1. Feature extraction and feature classification to determine the structural health of a system.

threshold for certain level of significance. If robust data normalization has been achieved, the DIs should be classified as outliers when feature vectors come from the damaged condition even when they include operational and environmental variability. All the methods for feature extraction and feature classification are described in detail in the next sections.

2.1. Autoregressive model for feature extraction

Usually, the modal parameters have been used in the SHM field for various applications as features that characterize the global condition of the structure. However, in this study, the AR model is used to extract damage-sensitive features, because the underlying linear stationary assumption makes it possible to detect the presence of nonlinearities in the time-series. It is considered that in a system where different dynamics are present at different times, the estimated parameters should change between intervals [31].

Alternatively, the AR models have been also used in SHM to extract damage-sensitive features from time-series data, either using the model parameters or residual errors [32,33]. For a measured time-series $s_1, s_2, \ldots, s_N$ the AR(p) model of order p is given by:

$$s_i = \sum_{j=1}^{p} \phi_j s_{i-j} + e_i, \tag{1}$$

where s_i is the measured signal and e_i is an unobservable random error at discrete time index i. The unknown AR parameters, ϕ_j, can be estimated using the least squares or the Yule–Walker equations [34]. The order of the model is always an unknown integer that needs to be estimated from the data. The Akaike information criterion (AIC) has been reported as one

of the most efficient techniques for order optimization [35]. The AIC is a measure of the goodness-of-fit of an estimated statistical model that is based on the trade-off between fitting accuracy and number of estimated parameters. In the context of AR models:

$$\mathrm{AIC} = N_t \ln(\varepsilon) + 2N_p, \tag{2}$$

where N_p is the number of estimated parameters, N_t is the number of predicted data points, and $\varepsilon = \mathrm{SSR}/N_t$ is the average sum-of-square residual (SSR) errors. The AR model with the lowest AIC value gives the optimal order p.

2.2. Support vector machine and kernel trick background

SVM is a powerful machine learning technique with strong regularization property for classification, regression, outlier detection and other learning tasks [36,37]. Given training vectors $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, m$, in two classes, and a label vector $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{y}_i \in \{1, -1\}$, supervised SVM solves the following primal optimization problem:

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi},\mathbf{b}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \mathbf{C}\sum_{i=1}^{m}\boldsymbol{\xi}_i, \\ \text{s.t.} \quad & \mathbf{y}_i(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) + \mathbf{b}) \geq 1 - \boldsymbol{\xi}_i, \\ & \boldsymbol{\xi}_i \geq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{3}$$

where $\boldsymbol{\phi}(\mathbf{x}_i)$ maps $\mathbf{x}_i$ into a high-dimensional space, $\boldsymbol{\xi}_i$ is the intermediate parameter, $\mathbf{C}$ is the regularization or penalty parameter and $\mathbf{b}$ is the adjustable parameter of the decision function; $\boldsymbol{\xi}_i$ is a slack variable for controlling how much training error is allowed and $\mathbf{C}$ is a parameter for balancing between $\boldsymbol{\xi}_i$ (the training error) and $\mathbf{w}$ (the margin). SVM finds a linear separating hyperplane with the maximal margin in the high-dimensional space. This minimization problem can be solved by Lagrangian multiplier or quadratic programming. Due to the possible high dimensionality of the vector $\mathbf{w}$, usually the problem must be transformed to the dual equivalent problem for solving:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}, \\ \text{s.t.} \quad & \mathbf{y}^T\boldsymbol{\alpha} = 0, \\ & 0 \leq \boldsymbol{\alpha}_i \leq \mathbf{C}, \quad i = 1, \ldots, m, \end{aligned} \tag{4}$$

where $\boldsymbol{\alpha}$ is the Lagrange multipliers, $\mathbf{e} = [1, \ldots, 1]^T$ is a vector of all ones, $\mathbf{Q}$ is a positive semi-definite matrix, $\mathbf{Q}_{i,j} \equiv \mathbf{y}_i\mathbf{y}_j\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \equiv \boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}_j)$ is a kernel function.

After the dual problem is solved, using the primal-dual relationship, the optimal $\mathbf{w}$ reads

$$\mathbf{w} = \sum_{i=1}^{m}\mathbf{y}_i\boldsymbol{\alpha}_i\boldsymbol{\phi}(\mathbf{x}_i) \tag{5}$$

and the decision function is

$$\mathrm{sgn}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}) + \mathbf{b}) = \mathrm{sgn}\left(\sum_{i=1}^{m}\mathbf{y}_i\boldsymbol{\alpha}_i\mathbf{K}(\mathbf{x}_i, \mathbf{x}) + \mathbf{b}\right). \tag{6}$$

Then, $\mathbf{y}_i\boldsymbol{\alpha}_i \forall i, \mathbf{b}$, label names, support vectors, and other information such as kernel parameters are stored in the trained model for future predictions.

The SVM algorithm mentioned early is a supervised linear classifier. In order to capture nonlinearity, nonlinear kernel function must be used. Some basic kernels are [38]:

- *linear*: $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$;
- *polynomial*: $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = (\boldsymbol{\gamma}\mathbf{x}_i^T\mathbf{x}_j + r)^d$, $\boldsymbol{\gamma} > 0$;
- *radial basis function*: $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\boldsymbol{\gamma}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$, $\boldsymbol{\gamma} > 0$;
- *sigmoid*: $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\boldsymbol{\gamma}\mathbf{x}_i^T\mathbf{x}_j + r)$;

where $\boldsymbol{\gamma}$, r, and d are kernel parameters. The radial basis function (RBF) kernel option is most commonly used in a large range of applications [39]. This kernel maps examples into a high-dimensional space so it, unlike the linear kernel, can handle the case when the relationship between class labels and features is nonlinear.

2.3. Feature classification

For general purposes, consider a training data matrix composed of normal condition data, $\mathbf{X} \in \mathbb{R}^{m \times n}$, with n-dimensional feature vectors from m different operational and environmental conditions when the structure is undamaged and a test data matrix, $\mathbf{Z} \in \mathbb{R}^{k \times n}$, where k is the number of feature vectors from the undamaged and/or damage conditions. Note that a feature vector represents some property of the system at a given time.

2.3.1. One-class support vector machine

The one-class SVM algorithm or distribution estimation was proposed by Schölkopf [40] for estimating the support of a high-dimensional distribution in an unsupervised way. Given training vectors $\mathbf{x}_i \in \mathbb{R}^n$, $i=1,\ldots,m$, without any class information (no labels), the primal problem of one-class SVM is

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi},\rho} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} - \rho + \frac{1}{\boldsymbol{\nu} m}\sum_{i=1}^{m}\boldsymbol{\xi}_i, \\ \text{s.t.} \quad & \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) \geq \rho - \boldsymbol{\xi}_i, \\ & \boldsymbol{\xi}_i \geq 0, \quad i=1,\ldots,m. \end{aligned} \tag{7}$$

The distribution estimation SVM introduces a new parameter $\boldsymbol{\nu} \in (0,1]$. This parameter is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors [41]. In other words, $\boldsymbol{\nu}$ has a similar effect of $\mathbf{C}$ for supervised SVM. Furthermore, the dual problem or unsupervised learning phase is

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}, \\ \text{s.t.} \quad & \mathbf{e}^T\boldsymbol{\alpha} = 1, \\ & 0 \leq \boldsymbol{\alpha}_i \leq 1/(\boldsymbol{\nu} m), \quad i=1,\ldots,m, \end{aligned} \tag{8}$$

where $\mathbf{Q}_{i,j} = \mathbf{K}(\mathbf{x}_i,\mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}_j)$. The decision function is defined as

$$\operatorname{sgn}\left(\sum_{i=1}^{m}\boldsymbol{\alpha}_i\mathbf{K}(\mathbf{x}_i,\mathbf{x}) - \rho\right). \tag{9}$$

When the model is obtained, a damage indicator can be generated by applying the MSD algorithm for each test vector $\mathbf{z}$ and the support vectors from the model,

$$\mathbf{DI} = (\mathbf{z}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}), \tag{10}$$

where $\boldsymbol{\mu}$ is the mean of the support vectors from the model and $\boldsymbol{\Sigma}$ is the covariance matrix of these support vectors. Therefore, this approach combines the robustness of the one-class SVM algorithm to derive a nonlinear model from undamaged data and the effectiveness of an outlier detection metric by means of the MSD.

2.3.2. Support vector data description

The SVDD algorithm, proposed by Tax and Duin [42], is an unsupervised method to find the boundary around a data set (a hypersphere). Given a set of training data $\mathbf{x}_i \in \mathbb{R}^n$, $i=1,\ldots,m$, the SVDD solves the following primal optimization problem:

$$\begin{aligned} \min_{\mathbf{R},\mathbf{a},\boldsymbol{\xi}} \quad & \mathbf{R}^2 + \mathbf{C}\sum_{i=1}^{m}\boldsymbol{\xi}_i, \\ \text{s.t.} \quad & \|\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{a}\|^2 \leq \mathbf{R}^2 + \boldsymbol{\xi}_i, i=1,\ldots,m, \\ & \boldsymbol{\xi}_i \geq 0, \quad i=1,\ldots,m, \end{aligned} \tag{11}$$

where the sphere is characterized by center $\mathbf{a}$ and radius $\mathbf{R}>0$, and $\boldsymbol{\phi}$ is a function mapping data to a high-dimensional space. After Eq. (11) is solved, a test instance $\mathbf{z}$ is detected as an outlier if $\|\boldsymbol{\phi}(\mathbf{z}) - \mathbf{a}\|^2 > \mathbf{R}^2$, i.e., if $\mathbf{z}$ is outside the boundary set by the SVDD model.

Because of the large number of variables in $\mathbf{a}$ after data mapping, the following Lagrange dual problem is solved in the training phase:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m}\boldsymbol{\alpha}_i\mathbf{Q}_{i,i} - \boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}, \\ \text{s.t.} \quad & \mathbf{e}^T\boldsymbol{\alpha} = 1, \\ & 0 \leq \boldsymbol{\alpha}_i \leq \mathbf{C}, \quad i=1,\ldots,m, \end{aligned} \tag{12}$$

where $\mathbf{e} = [1,\ldots,1]^T$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_m]^T$, and $\mathbf{Q}$ is the kernel matrix such that $\mathbf{Q}_{i,j} = \boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}_j), \forall 1 \leq i,j \leq m$. The DIs are generated by the same procedure described for the one-class SVM algorithm. However, in this case, the support vectors represent a hypersphere, instead of a hyperplane.

A relationship between one-class SVM and SVDD algorithms can be established by the following expression [43]:

$$\mathbf{C} = \frac{1}{\boldsymbol{\nu} m}. \tag{13}$$

2.3.3. Kernel principal component analysis

The KPCA algorithm is the nonlinear extension of the linear principal component analysis (PCA) [44]. The input training matrix $\mathbf{X}$ is mapped by $\boldsymbol{\phi}$: $\mathbf{X} \rightarrow \mathcal{F}$ to a high-dimensional feature space $\mathcal{F}$. The linear PCA is applied on the mapped data $\mathcal{T}_{\Phi} = \{\boldsymbol{\phi}(\mathbf{x}_1),\ldots,\boldsymbol{\phi}(\mathbf{x}_m)\}$. The computation of the principal components and the projection on these components can be

expressed in terms of dot products thus the kernel functions can be employed. The KPCA trains the kernel data projection:

$$\mathbf{y} = \mathbf{A}^T \mathbf{k}(\mathbf{x}) + \mathbf{o}, \tag{14}$$

where $\mathbf{A} \in \mathbb{R}^{m \times d}$ is the projection matrix, $\mathbf{k}(\mathbf{x}) = [\mathbf{k}(\mathbf{x}, \mathbf{x}_1), \ldots, \mathbf{k}(\mathbf{x}, \mathbf{x}_m)]^T$ is a vector of kernel functions centered in the training vectors and $\mathbf{o} \in \mathbb{R}^d$ is the bias vector. Additionally, $\mathbf{A}$ is the matrix containing the corresponding eigenvectors. The eigenvectors associated with the higher eigenvalues are the principal components of the data matrix and they correspond to the dimensions that have the largest variability in the data. Basically, this method permits one to perform a transformation by retaining only the principal components d, also known as the number of factors, in a high-dimensional feature space $\mathbb{R}^{m \times m}$.

The kernel mean squared reconstruction error, which must be minimized, is defined as

$$\varepsilon_{\mathrm{KMS}}(\mathbf{A}, \mathbf{o}) = \frac{1}{m} \sum_{i=1}^{m} \| \boldsymbol{\phi}(\mathbf{x}_i) - \tilde{\boldsymbol{\phi}}(\mathbf{x}_i) \|^2, \tag{15}$$

where the reconstructed vector $\tilde{\boldsymbol{\phi}}(\mathbf{x})$ is given as a linear combination of the mapped data $\mathcal{T}_{\boldsymbol{\phi}}$:

$$\tilde{\boldsymbol{\phi}}(\mathbf{x}) = \sum_{i=1}^{m} \beta_i \boldsymbol{\phi}(\mathbf{x}_i), \quad \beta = \mathbf{A}(\mathbf{y} - \mathbf{o}). \tag{16}$$

In contrast to the linear PCA, the explicit projection from the feature space $\mathcal{F}$ to the input space usually does not exist [45]. The problem is to find the vector $\mathbf{x}$ and its image $\boldsymbol{\phi}(\mathbf{x}) \in \mathcal{F}$ that well approximates the reconstructed vector $\tilde{\boldsymbol{\phi}}(\mathbf{x}) \in \mathcal{F}$. This procedure consists of the following steps:

Step 1: Project the input vector $\mathbf{x}_{\mathrm{in}} \in \mathbb{R}^n$ onto its lower-dimensional representation $\mathbf{y} \in \mathbb{R}^d$ using Eq. (14).
Step 2: Compute the output vector $\mathbf{x}_{\mathrm{out}} \in \mathbb{R}^n$ which is satisfactory pre-image of the reconstructed vector $\tilde{\boldsymbol{\phi}}(\mathbf{x}_{\mathrm{in}})$, such that

$$\mathbf{x}_{\mathrm{out}} = \underset{\mathbf{x}}{\operatorname{argmin}} \| \boldsymbol{\phi}(\mathbf{x}) - \tilde{\boldsymbol{\phi}}(\mathbf{x}_{\mathrm{in}}) \|^2. \tag{17}$$

For the test matrix $\mathbf{Z}$, the residual errors matrix $\mathbf{E}$ is given by:

$$\mathbf{E} = \mathbf{Z} - \hat{\mathbf{Z}}, \tag{18}$$

where $\hat{\mathbf{Z}}$ corresponds to the estimated feature vectors that are the output of the pre-image computation from the model obtained by KPCA in the training phase. In other words, the pre-image problem maps the feature vectors from the high-dimensional feature space back to the input space. An iterative fixed point algorithm is used for this purpose [46]. In order to establish a quantitative measure of damage, for the feature vector f ($f = 1, 2, \ldots, k$), a DI is adopted in the form of the squared root of the sum-of-square errors (Euclidean norm):

$$\mathbf{DI}(f) = \| \mathbf{e}_f \|. \tag{19}$$

If f feature vector is related to the undamaged condition, $\mathbf{z}_f \approx \hat{\mathbf{z}}_f$ and $\mathbf{DI}(f) \approx 0$. On the other hand, if the feature vector comes from the damaged condition, the residual errors increase, and the DI deviates from zero, indicating an abnormal condition in the structure.

2.3.4. Greedy kernel principal component analysis

The GKPCA algorithm is an efficient method to compute the KPCA algorithm [47]. Let $\mathcal{T}_{\mathbf{X}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, m$, be the set of input training vectors. The goal is to train the kernel data projection:

$$\mathbf{y} = \mathbf{A}^T \mathbf{k}_{\mathbf{S}}(\mathbf{x}) + \mathbf{o}, \tag{20}$$

where $\mathbf{k}_{\mathbf{S}}(\mathbf{x}) = [\mathbf{k}(\mathbf{x}, \mathbf{s}_1), \ldots, \mathbf{k}(\mathbf{x}, \mathbf{s}_m)]^T$ are the kernel functions centered in the vectors from $\mathcal{T}_{\mathbf{S}} = \{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$. The vector set $\mathcal{T}_{\mathbf{S}}$ is a subset of training data $\mathcal{T}_{\mathbf{X}}$.

In opposition to the PCA, the basis vectors of the lower-dimensional space used for data representation are properly selected vectors from the training set and not as their linear combinations. The basis vectors can be selected by a simple algorithm which has low computational requirements and allows real-time processing by approximating the training set in the high-dimensional space [47]. Moreover, in contrast to the original KPCA, the subset $\mathcal{T}_{\mathbf{S}}$ does not contain all the training vectors in $\mathcal{T}_{\mathbf{X}}$ thus the complexity of the projection in Eq. (20) is reduced compared to Eq. (14). The objective of the GKPCA is to minimize the reconstruction error while the size of the subset $\mathcal{T}_{\mathbf{S}}$ is kept small. The DIs are generated by the same procedure described for KPCA, considering the residuals from $\mathbf{E}$.

3. Test bed structure and data

The standard data sets used in this study are from a three-story frame aluminum structure reported in [48], and has been intensively used for SHM validation in recent statistical damage identification approaches [13,23,25]. These data were collected by four accelerometers mounted on a test bed building model, as shown in Fig. 2, forming a essentially four-degree-of-freedom system with varied practical conditions, including variations in stiffness (e.g., to simulate temperature

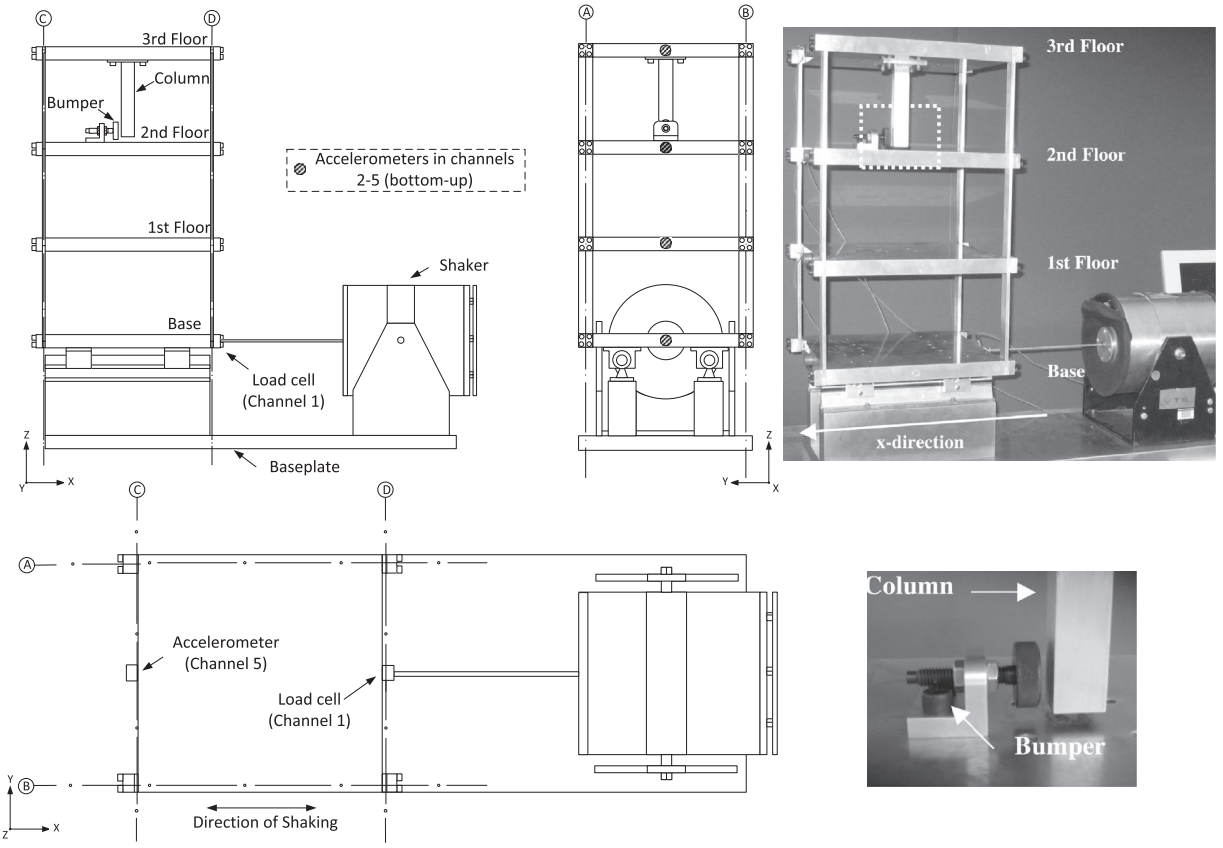

Fig. 2. Three-story frame aluminum structure and shaker.

variations) and in mass-loading (e.g., to simulate traffic). Those changes were designed to introduce variability in the fundamental natural frequency up to approximately 7 percent from the baseline condition, which is within the range normally observed in real-world structures [49,50]. Nonlinear damage was introduced by contacting a suspended column with a bumper mounted on the floor below to simulate fatigue crack that can open and close under loading conditions or loose connections in structures. Different levels of damage were created by adjusting the gap between the column and the bumper (smaller the gap, higher the level of damage). More details about the test structure can be found in [48].

Acceleration time-series (discretized into 4096 data points sampled at 3.125 ms intervals corresponding to a sampling frequency of 320 Hz) from 17 different structural state conditions were collected, as described in Table 1. For each structural state condition, data were acquired from 100 separate tests. According to the test description [51], state1 is the baseline condition (reference state) of the structure and states2–9 include those states with simulated operational and environmental variability. State14 is considered as the most severe damaged one as it corresponds to the smallest gap case, which induces the highest number of impacts. State10 is the least severe damaged scenario and states11–13 represent mid-level damage scenarios. States15–17 are the variant states of either state10 or state13 with mass added effect in order to create more realistic conditions.

4. Experimental results and analysis

In this study, the AR parameters from response time series are used as damage-sensitive features. Thus, for each test of each state condition, the parameters are estimated using the least squares technique applied upon time-series from all four accelerometers (channels 2–5) and stored into a feature vector. For each test, the number of estimated parameters is $p \times 4$ and ε is the sum of the average sum-of-square errors of channels 2–5. In this analysis is assumed an output-only damage detection approach, and so data from the channel 1 (the input force) is not used.

The linear AR models achieve enough small AIC value around $p=10$ [23]. Based on that evaluation, for each test, four individual AR(10) models are used to fit the corresponding time series from the four accelerometers and their parameters are used as damage-sensitive features in concatenated format, yielding 40-dimensional feature vectors. Note that AR parameters should be constant when estimated based on time-series data obtained from time-invariant systems. However, in the presence of operational and environmental conditions as well as damage, the parameters are expected to change

Table 1
Data labels of the structural state conditions.

Label	State condition	Description
State1	Undamaged	Baseline condition
State2	Undamaged	Added mass (1.2 kg) at the base
State3	Undamaged	Added mass (1.2 kg) on the 1st floor
State4	Undamaged	
State5	Undamaged	States4–9: 87.5% stiffness reduction at
State6	Undamaged	various positions to simulate temperature
State7	Undamaged	impact (more details in [48])
State8	Undamaged	
State9	Undamaged	
State10	Damaged	Gap (0.20 mm)
State11	Damaged	Gap (0.15 mm)
State12	Damaged	Gap (0.13 mm)
State13	Damaged	Gap (0.10 mm)
State14	Damaged	Gap (0.05 mm)
State15	Damaged	Gap (0.20 mm) and mass (1.2 kg) at the base
State16	Damaged	Gap (0.20 mm) and mass (1.2 kg) on the 1st floor
State17	Damaged	Gap (0.10 mm) and mass (1.2 kg) on the 1st floor

accordantly, as shown in Fig. 3, for instance, for one test corresponding to states1, 3, 5, 7, 9, 10, 14 and 17. The feature vectors are divided according to their structural condition into two major groups: undamaged and damaged conditions. Particular changes are noticed in the amplitude of the AR parameters at channels 4 and 5 when features are from the damaged condition (those channels are closer to the source of damage). Clearly, the AR parameters reveal high sensitivity to the presence of damage. In general, the higher the level of damage (the smaller the gap), the lower the amplitude of the AR parameters.

For generalization purposes, the feature vectors are split into the training and test matrices. The training matrix, $\mathbf{X}$, permits each algorithm to learn the underlying distribution and dependency of all undamaged states on the simulated operational and environmental variability. Thus, this matrix is composed of AR parameters from 50 out of 100 tests of each undamaged state (states1–9), and so it has a dimension of 450×40. The test matrix $\mathbf{Z}$ (1250×40) is composed of AR parameters from the remaining 50 tests of each undamaged state together with AR parameters from all the 100 tests of each damaged state (states10–17). This procedure permits one to evaluate the generalization performance of the machine learning algorithms in an exclusive manner, because time-series used in the test phase are not included in the training phase. During the test phase, the algorithms are expected to detect deviations from the normal condition when feature vectors come from damaged states, even in the presence of operational and environmental effects.

The next step is to carry out statistical modeling for feature classification. In that regard, the algorithms based on one-class SVM, SVDD, KPCA and GKPCA are implemented in an unsupervised learning mode by first taking into account features from all the undamaged state conditions (training matrix). All kernel-based algorithms use the RBF kernel with parameter $\gamma = 0.025$. The regularization parameter is defined as $\nu = 0.8$ ($\mathbf{C}$ is obtained by the equivalence between one-class SVM and SVDD). The KPCA and GKPCA algorithms are configured to retain 90 percent of the variability in the data after dimension reduction. The subset of 25 percent of the training data is used for GKPCA kernel projection. The algorithms based on AANN, FA, MSD and SVD are implemented and configured as described in [23]. Finally, for each algorithm, the DIs are stored into a 1250-length vector.

The ROC curves provide a comprehensive means of summarizing the performance of classifiers. They focus on the trade-off between true detection and false alarm rates. The point at the left-upper corner of the plot (0, 1) is called a perfect classification. Fig. 4 plots, partially, the ROC curves for the kernel-based algorithms. Qualitatively, looking at the curves, one can verify that none of the algorithms can have a perfect classification with a linear threshold because none of the curves go through the left-upper corner, neither have supremacy in terms of true detection rate for all the false alarm domain. Furthermore, one can figure out that for levels of significance around 5 percent, the KPCA and GKPCA have better true detection rate than one-class SVM and SVDD, i.e., the approaches that maximize the true detection of damaged cases with similar performances in terms of false alarm rate. Nonetheless, for low probabilities of false alarm, all the algorithms show to have acceptable true detection rate (for instance, for a false alarm rate of 0.05, the minimal true detection rate is around 0.98, given by the one-class SVM and SVDD). Note that all proposed algorithms apply data transformation in the high-dimensional feature space to achieve a data model that represents the normal structural condition. Nevertheless, the one-class SVM and SVDD algorithms discover nonlinear relationship in the data via hyperplane and hypersphere separations. On the other hand, the KPCA and GKPCA algorithms reduce the data dimensionality in high-dimensional space, storing the principal components that have the largest variability in the data.

Due to the similar classification performances of the one-class SVM and SVDD, and the superiority of the GKPCA relative to the KPCA, the authors chose the one-class SVM and GKPCA algorithms for a comparative analysis with algorithms already analyzed in the literature. In the first case, Fig. 5 shows a comparison between the FA, SVD, one-class SVM and GKPCA

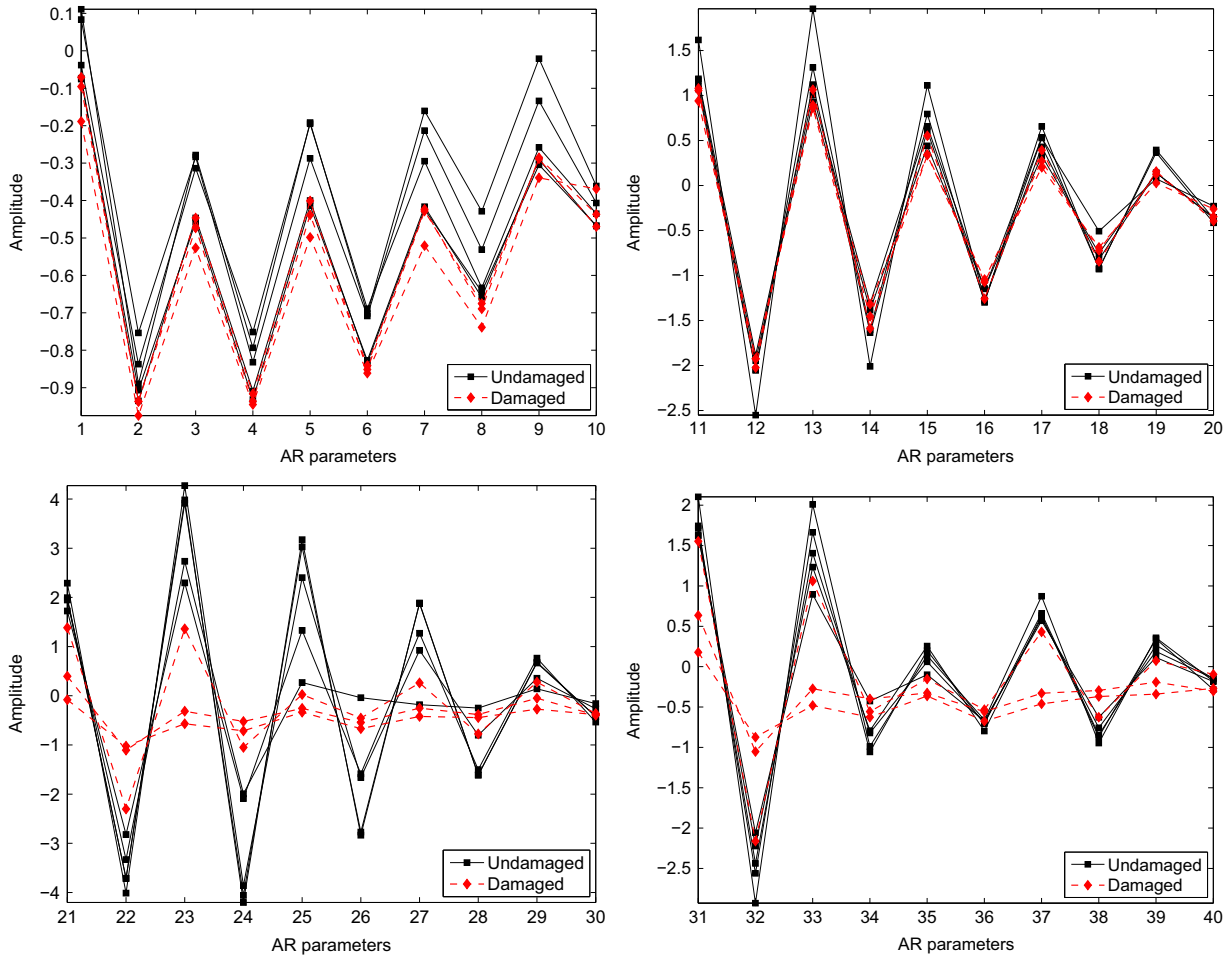

Fig. 3. Feature vectors from one test of states1, 3, 5, 7 and 9 (undamaged) and states10, 14 and 17 (damaged): AR parameters from channels 2 (upper left), 3 (upper right), 4 (lower left) and 5 (lower right).

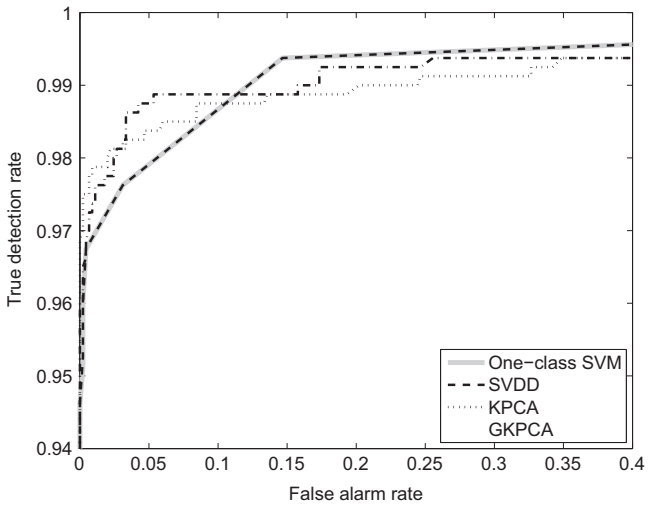

Fig. 4. Partial ROC curves for the kernel-based algorithms.

algorithms. In the whole false alarm range and for a given threshold, the kernel-based algorithms have, clearly, better performance to detect abnormal conditions in the test structure than the FA and SVD algorithms, which is expected as the FA and SVD algorithms can only capture linear patterns in the data. In the second case, Fig. 6 shows a comparison between

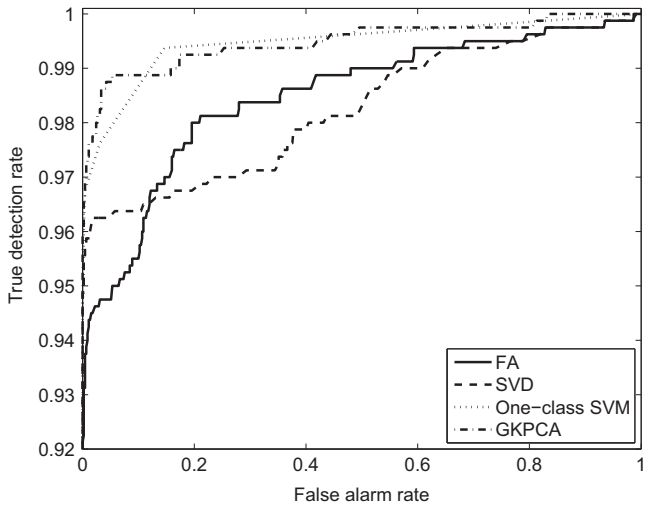

Fig. 5. ROC curves for the FA, SVD, one-class SVM and GKPCA algorithms.

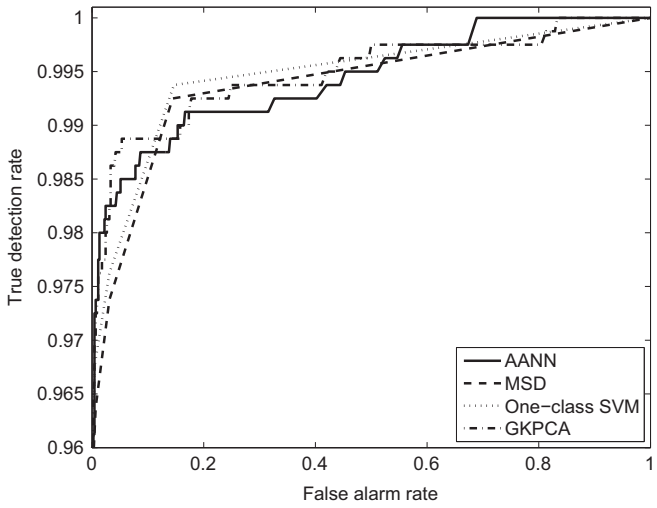

Fig. 6. ROC curves for the AANN, MSD, one-class SVM and GKPCA algorithms.

the AANN, MSD, one-class SVM and GKPCA algorithms. In general, the ROC curve of the one-class SVM follows the same behavior as the ROC curve of the MSD algorithm; the GKPCA and AANN algorithms also output similar ROC curves. In particular, the one-class SVM has superior performance in feature classification than MSD in whole false alarm rate range, by properly selecting the support vectors to represent an optimal subset of the undamaged condition; whereas the GKPCA seems to be more effective than the AANN in most points of the ROC curves by assimilating the normal structural condition embedded into the principal components, especially for false alarm rates around 5 percent (the threshold normally used in real-world scenarios).

In order to quantify the performance of the classifiers for a given threshold, Figs. 7 and 8 plot the DIs for the feature vectors of the entire test data along with a threshold defined based on the 95 percent cut-off value over the training data. All the algorithms show a monotonic relationship between the level of damage and the amplitude of the DI, even when operational and environmental variability is present, i.e., the approaches are able to remove the operational and environmental effects in such a way that DIs from states15, 16 and 10 have similar amplitude, as well as state17 is associated with state13. Recall that states15–17 are the variant states of either state10 or state13 with operational effects.

The Type I (false-positive indication of damage) and Type II (false-negative indication of damage) errors are traditionally used to report the performance of a binary classification. This technique recognizes that a false-positive classification may have different consequences than false-negative one. In Figs. 7 and 8, the Type I errors are DIs that exceed the threshold value in the undamaged condition domain (1–450). On the other hand, the Type II errors are DIs that do not surpass the threshold value in the damaged condition domain (451–1250). Table 2 summarizes the number and percentage of Type I

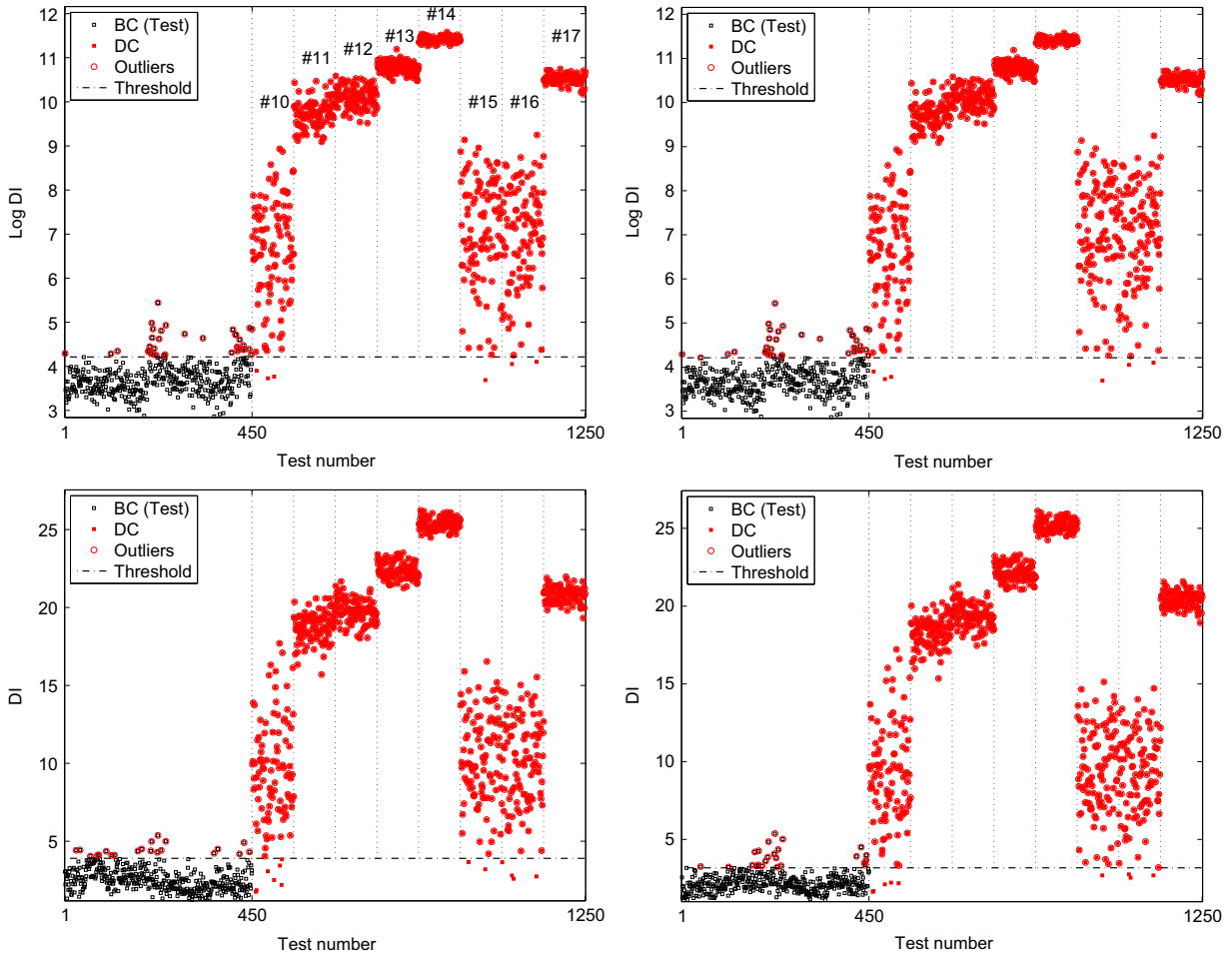

Fig. 7. DIs calculated based on feature vectors from the undamaged/baseline condition (BC) and damaged condition (DC) using the four kernel-based algorithms along with thresholds defined by the 95% cut-off value over the training data: one-class SVM (upper left), SVDD (upper right), KPCA (lower left), and GKPCA (lower right).

and Type II errors for each algorithm (based on the composition of the test matrix **Z**), demonstrating a trade-off between Type I and Type II errors for all eight algorithms.

In terms of an overall analysis, the one-class SVM and SVDD algorithms have similar performance and the best performance overall to detect damage (0.75 percent); however they output a relatively high false alarm rate (≥ 8 percent), as the selected support vectors can easily identify the damaged data, but are not properly representing the normal condition as the number of Type I errors is higher than 5 percent. The FA algorithm has the better performance to avoid false indications of damage (2.22 percent), but the worst performance to detect damage (5.38 percent) due to its sensitivity to the number of factors driving changes in features. The AANN reveals the worst performance in terms of Type I errors (9.78 percent), but a relatively good performance in terms of minimization of Type II errors (1.25 percent), which gives indications that when the sensitivity (the portion of damaged cases correctly identified) of the classifier is increased, and so it detects more damaged cases, it also increases the number to mislabels of undamaged cases. Thus, as the sensitivity goes up, specificity (the portion of undamaged cases which are correctly identified) goes down. The KPCA and GKPCA attempt a balancing between Type I and Type II errors, having the minimum errors in total, which is supported by the retention of principal components in the high-dimensional space, which eliminates variability caused by operational and environmental effects. Finally, the proposed four algorithms have a tendency to reduce the total number of misclassifications (average of 3.04 percent) when compared with the four approaches previously tested (average of 4.3 percent). This superiority might be related with the ability of the proposed algorithms to find nonlinear patterns in the data via the kernel trick, as well as the independence on the choice of the initial parameters (unlike the FA, for instance, where one needs to set the number of hidden factors driving changes in the damage-sensitive features).

In particular, and for the proposed algorithms, the one-class SVM and SVDD algorithms have similar performance in the feature classification (changing only Type I error), demonstrating the relationship between them via the regularization parameters $\boldsymbol{\nu}$ and **C**. In turn, the KPCA and GKPCA algorithms have also similar performance; however, both Type I and Type

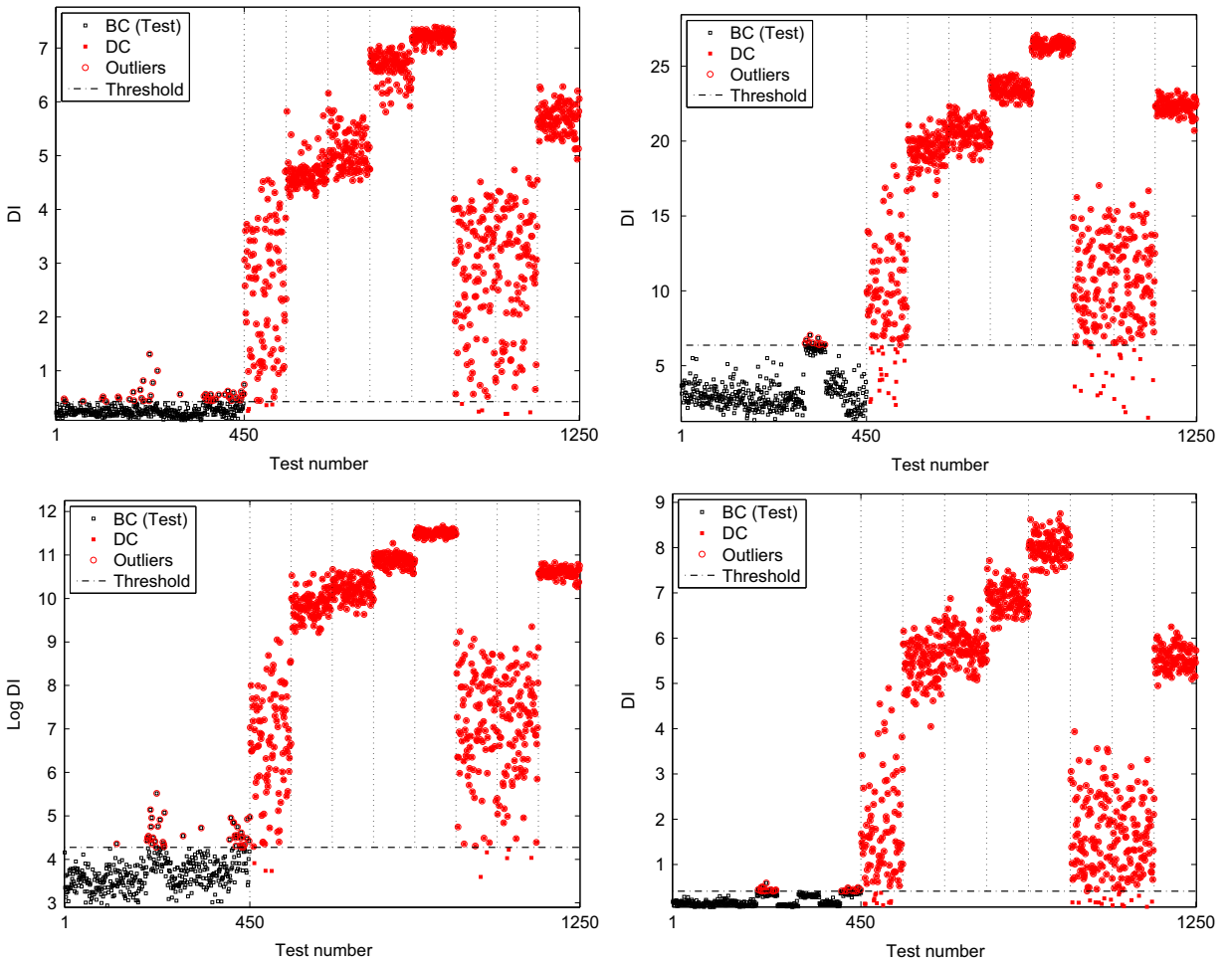

Fig. 8. DIs calculated based on feature vectors from the undamaged/baseline condition (BC) and damaged condition (DC) using the state-of-the-art algorithms along with thresholds defined by the 95% cut-off value over the training data: AANN (upper left), FA (upper right), MSD (lower left), and SVD (lower right).

Table 2
Number and percentage of Type I and Type II errors for each algorithm.

Algorithm	Error		
	Type I	Type II	Total
AANN	44 (9.78%)	10 (1.25%)	54 (4.32%)
FA	10 (2.22%)	43 (5.38%)	53 (4.24%)
MSD	42 (9.33%)	8 (1.00%)	50 (4.00%)
SVD	29 (6.44%)	29 (3.62%)	58 (4.64%)
One-class SVM	36 (8.00%)	6 (0.75%)	42 (3.36%)
SVDD	37 (8.22%)	6 (0.75%)	43 (3.44%)
KPCA	21 (4.67%)	13 (1.62%)	34 (2.72%)
GKPCA	24 (5.33%)	9 (1.12%)	33 (2.64%)

II errors change slightly and show clearly a trade-off. One should note that the indications given by the ROC curves from Fig. 4, about the true detection rate, are not consistent with the results from Table 2. Basically, from the ROC curves, one might infer that the KPCA and GKPCA have better true detection rate for a level of significance around 5 percent than the one-class SVM and SVDD. However, the table shows that the one-class SVM and SVDD minimize the number of Type II errors, i.e., maximize the true detection rate. This phenomenon is related with the threshold definition, as it was defined based only on 50 percent of the undamaged data. Additionally, for the KPCA and GKPCA, as the level of false alarm rate is very close to the 5 percent level of significance assumed to set up the threshold, it gives indications that these algorithms

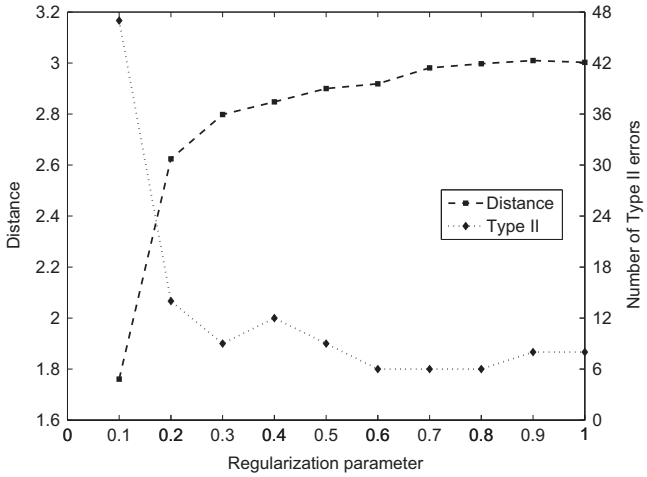

Fig. 9. Distance between the mean of the DIs from the undamaged and damaged conditions along with the number of Type II errors as a function of the regularization parameter ($\nu = 0.1, \ldots, 1$), using the one-class SVM algorithm.

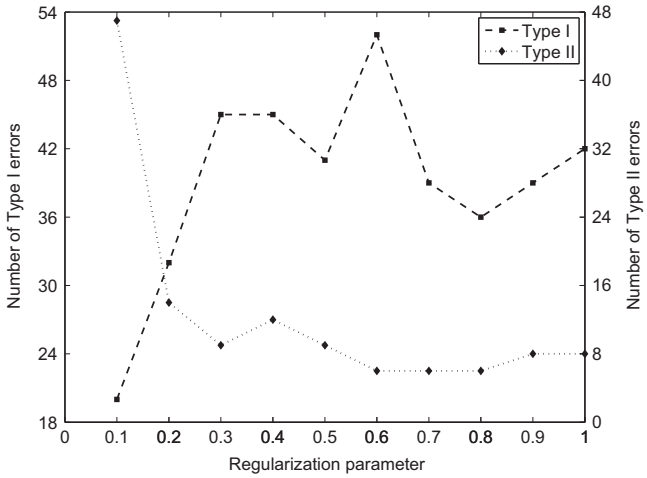

Fig. 10. Classification performance (Type I and Type II errors) as a function of the regularization parameter ($\nu = 0.1, \ldots, 1$), using the one-class SVM algorithm.

have learnt the baseline condition during the training phase and they can generalize well for new undamaged data not used in the training phase.

Nevertheless, all kernel-based algorithms perform relatively well on these standard data sets with percent of total misclassifications (Type I and Type II errors) ranging between 2.64 percent and 3.44 percent of the total number of test observations, which can be considered an acceptable result. Additionally, based on Table 2 and for these specific data sets, one can infer that the one-class SVM and SVDD algorithms are preferred when one wants to minimize false-negative indications of damage and when life safety issues are the main reason for deploying a SHM system. On the other hand, the KPCA and GKPCA algorithms are more appropriate when one wants to minimize false-positive indications of damage without increasing, significantly, the false-negative indications of damage, and reliability issues are driving a SHM system. However, the KPCA and GKPCA have shown to have better generalization performance, which is a very important advantage for real-world applications, where the thresholds are defined based on undamaged data used in the training phase.

Finally, note that the classification performance of the one-class SVM and SVDD algorithms can potentially be improved by adjusting the regularization parameters $\boldsymbol{\nu}$ and $\mathbf{C}$, respectively. To highlight the influence of the regularization parameter assumed on both algorithms, Fig. 9 shows the distance between the mean of the DIs from the undamaged and damaged conditions along with the number of Type II errors as a function of the regularization parameter $\boldsymbol{\nu}$. Based on the distance between means and the minimization of number of Type II errors, the appropriate regularization parameter is 0.8 (based on the converging point) and approximately 0.0028 for the one-class SVM and SVDD algorithms, respectively. This result gives an indication that the regularization parameter assumed in the previous analysis is nearly the optimal solution.

Nonetheless, alternative solutions can be found in Fig. 10. Basically, when ν is equal to 0.2, 0.7 and 0.9, the total errors are 46 (3.68 percent), 45 (3.6 percent) and 47 (3.76 percent), respectively, which are still less than the total errors outputted by the AANN, FA, MSD and SVD. Note that when the regularization parameter increases, the number of support vectors also increases. Therefore, the choice of a regularization parameter with a small value may lead to underfitting as well as the choice of a regularization parameter with high value may lead to overfitting.

5. Summary and conclusions

In this paper, the performance of four kernel-based algorithms (one-class SVM, SVDD, KPCA and GKPCA) for structural damage detection, under varying operational and environmental conditions, was compared using benchmark data sets from a well-known base-excited three-story frame structure. The data sets are characterized by 17 different structural state conditions, including linear changes caused by varying stiffness and mass-loading conditions as well as nonlinear effects caused by damage. Different levels of damage were created by adjusting the gap between a suspended column and a bumper.

The kernel-based algorithms were shown to be reliable approaches to create a global DI that can separate damaged from undamaged conditions, even when the structure is operating under varying operational and environmental conditions. The comparison between the proposed four kernel-based algorithms and alternative algorithms studied before (AANN, FA, MSD and SVD) permitted one to conclude that the proposed ones have better classification performance as given by the lower number of misclassifications (both Type I and Type II errors). Two fundamental reasons are given to justify the general improvement of the proposed algorithms over the previous ones: (i) the previous algorithms were essentially linear in their formulation, with exception of the AANN; (ii) all the proposed algorithms (SVM, SVDD, KPCA and GKPCA) operate, essentially, in the high-dimensional space, giving them capabilities to model nonlinear patterns presented in the original observation space.

Within the four proposed algorithms, the KPCA and GKPCA outputted better results in terms of minimization of total misclassifications; two fundamental reasons are also proposed for that behavior: (i) all four algorithms map the original observations into the high-dimensional space; however, the KPCA and GKPCA map the original observations into the high-dimensional space, in order to capture the operational and environmental effects with a known percentage, and back to the original space to perform the damage detection; conversely, the SVM and SVDD performs the data normalization by means of a data subset (support vectors) selected in the high-dimensional space, which accounts for an unknown percentage of the variability; (ii) the GKPCA and KPCA perform the damage detection by retaining the principal components that take into account 90 percent of the data variability, which might be useful to discard some sort of noise and singularities from the data that can mask changes caused by damage from changes caused by operational and environmental conditions. This is actually an advantage, as shown by the trade-off between Type I and Type II errors. As the GKPCA and KPCA eliminate some noise from the data, they are not too complex and have better generalization performance.

In particular, the proposed algorithms are independent of the initial conditions, as opposed to the AANN, for instance, where the performance of the neural network is strongly dependent on the choice of the initial parameters. Therefore, if the kernel-based approaches are configured with the same parameters defined in this study, then they should provide the same results. In general, considering different application cases, the proposed algorithms are recommended to remove the influences of varying operational and environmental conditions on damage-sensitive features, especially when nonlinear temperature–stiffness relationship is present. The KPCA and GKPCA algorithms build data models where the principal components account for nearly all the operational and environmental influences, providing a better trade-off between sensitivity and specificity. On the other hand, the one-class SVM and SVDD algorithms generate data models where the normal structural condition used in the training phase is well discriminated by means of the support vectors, improving the detection of abnormal cases. However, the hyperplane and hypersphere separations are unable to generalize well, when compared with the KPCA and GKPCA algorithms, to undamaged data not provided in the training phase, i.e., the support vectors do not account for nearly all normal variability.

A parametric study carried out to establish the relationship between the classification performance and the regularization parameter, in the one-class SVM and SVDD algorithms, permitted one to verify those algorithms maximize the classification performance when the regularization parameter is chosen properly to avoid underfitting and overfitting in the data model generated for feature classification. Note that the cross-validation is the most used method to obtain the optimal regularization parameter, i.e., the best model. In order to do so, it is necessary to divide the data sets into three different subsets: training, validation and testing. In the training phase is considered only data from the undamaged condition to train the algorithms. The validation step is a sequence of iterations with different values for the regularization parameter to obtain the best model for generalization purposes, using both undamaged and damaged data. The test phase of the algorithms is done using the best model obtained in the cross-validation. However, in this study, this procedure was not taken into account, in order to be consistent with the previous study carried out by the authors.

Additionally, the proposed approaches can also perform damage localization when an AR model is applied, individually, on each sensor and the damage detection process is performed at the sensor level. In this paper, the AR model is applied on each sensor, but the AR parameters from the four sensors were concatenated into a single feature vector. Therefore, the damage detection is performed globally.

Finally, in the context of data normalization for damage detection, this study addressed the implementation and comparison of machine learning algorithms to establish the normal condition as a function of the operational and environmental variability. It is important to note that none of these algorithms require a direct measure of the sources of variability (e.g., traffic loading and temperature). Instead, the algorithms rely only on measured response time-series data acquired under varying operational and environmental effects.

Acknowledgments

The authors acknowledge the support received from CNPq (454483/2014-7), CAPES and Vale S.A. (FAPESPA-112794/2010).

References

[1] E. Figueiredo, I. Moldovan, M.B. Marques, Condition Assessment of Bridges: *Past, Present, and Future—A Complementary Approach*, in: Universidade Católica Editora, Portugal, 2013.
[2] B. Glisic, D. Inaudi, *Fibre Optic Methods for Structural Health Monitoring*, John Wiley & Sons, Inc., Hoboken, NJ, United States, 2007.
[3] C.R. Farrar, K. Worden, An introduction to structural health monitoring, *Philosophical Transactions of the Royal Society: Mathematical, Physical & Engineering Sciences* 365 (1851) (2007) 303–315, http://dx.doi.org/10.1098/rsta.2006.1928.
[4] H. Sohn, C.R. Farrar, Damage diagnosis using time series analysis of vibration signals, *Smart Materials and Structures* 10 (3) (2001) 446–451, http://dx.doi.org/10.1088/0964-1726/10/3/304.
[5] K. Worden, J.M. Dulieu-Barton, An overview of intelligent fault detection in systems and structures, *Structural Health Monitoring* 3 (1) (2004) 85–98, http://dx.doi.org/10.1177/147592170404186.
[6] S.G. Mattson, S.M. Pandit, Statistical moments of autoregressive model residuals for damage localisation, *Mechanical Systems and Signal Processing* 20 (3) (2006) 627–645, http://dx.doi.org/10.1016/j.ymssp.2004.08.005.
[7] E. Figueiredo, M. Todd, C. Farrar, E. Flynn, Autoregressive modeling with state-space embedding vectors for damage detection under operational variability, *International Journal of Engineering Science* 48 (10) (2010) 822–834, http://dx.doi.org/10.1016/j.ijengsci.2010.05.005.
[8] E.J.F. Figueiredo, Damage Identification in Civil Engineering Infrastructure under Operational and Environmental Conditions, Doctor of Philosophy in Civil Engineering, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal, 2010.
[9] K. Worden, G. Manson, The application of machine learning to structural health monitoring, *Philosophical Transactions of the Royal Society: Mathematical, Physical & Engineering Sciences* 365 (1851) (2007) 515–537, http://dx.doi.org/10.1098/rsta.2006.1938.
[10] K. Worden, C.R. Farrar, G. Manson, G. Park, The fundamental axioms of structural health monitoring, *Philosophical Transactions of the Royal Society: Mathematical, Physical & Engineering Sciences* 463 (2082) (2007) 1639–1664, http://dx.doi.org/10.1098/rspa.2007.1834.
[11] H. Sohn, K. Worden, C.R. Farrar, Statistical damage classification under changing environmental and operational conditions, *Journal of Intelligent Material Systems and Structures* 13 (9) (2002) 561–574, http://dx.doi.org/10.1106/104538902030904.
[12] K. Worden, A.J. Lane, Damage identification using support vector machines, *Smart Materials and Structures* 10 (3) (2001) 540–547, http://dx.doi.org/10.1088/0964-1726/10/3/317.
[13] L. Bornn, C.R. Farrar, G. Park, K. Farinholt, Structural health monitoring with autoregressive support vector machines, *Journal of Vibration and Acoustics* 131 (2) (2009) http://dx.doi.org/10.1115/1.3025827. 021004-1–9.
[14] Y. Kim, J.W. Chong, K.H.C.J. Kim, Wavelet-based AR-SVM for health monitoring of smart structures, *Smart Materials and Structures* 22 (1) (2013) 1–12, http://dx.doi.org/10.1088/0964-1726/22/1/015003.
[15] J.W. Chong, Y. Kim, K.H. Chon, Nonlinear multiclass support vector machine-based health monitoring system for buildings employing magnetorheological dampers, *Journal of Intelligent Material Systems and Structures* 25 (12) (2014) 1456–1468, http://dx.doi.org/10.1177/1045389X13507343.
[16] N.L. Khoa, B. Zhang, Y. Wang, F. Chen, S. Mustapha, Robust dimensionality reduction and damage detection approaches in structural health monitoring, *Structural Health Monitoring* 13 (4) (2014) 406–417, http://dx.doi.org/10.1177/1475921714532989.
[17] A. Mita, H. Hagiwara, Quantitative damage diagnosis of shear structures using support vector machine, *KSCE Journal of Civil Engineering* 7 (6) (2003) 683–689, http://dx.doi.org/10.1007/BF02829138.
[18] H.-X. He, W. ming Yan, Structural damage detection with wavelet support vector machine: introduction and applications, *Structural Control and Health Monitoring* 14 (1) (2007) 162–176, http://dx.doi.org/10.1002/stc.150.
[19] A. Cury, C. Crémona, Pattern recognition of structural behaviors based on learning algorithms and symbolic data concepts, *Structural Control and Health Monitoring* 19 (2) (2012) 161–186, http://dx.doi.org/10.1002/stc.412.
[20] H. HoThu, A. Mita, Damage detection method using support vector machine and first three natural frequencies for shear structures, *Open Journal of Civil Engineering* 3 (2) (2013) 104–112, http://dx.doi.org/10.4236/ojce.2013.32012.
[21] C.M. Wen, S.L. Hung, C.S. Huang, J.C. Jan, Unsupervised fuzzy neural networks for damage detection of structures, *Structural Control and Health Monitoring* 14 (1) (2007) 144–161, http://dx.doi.org/10.1002/stc.116.
[22] C.R. Farrar, K. Worden, *Structural Health Monitoring: A Machine Learning Perspective*, John Wiley & Sons, Inc., Hoboken, NJ, United States, 2013.
[23] E. Figueiredo, G. Park, C.R. Farrar, K. Worden, J. Figueiras, Machine learning algorithms for damage detection under operational and environmental variability, *Structural Health Monitoring* 10 (6) (2011) 559–572, http://dx.doi.org/10.1177/1475921710388971.
[24] M.A. Torres-Arredondo, D.A. Tibaduiza, L.E. Mujica, J. Rodellar, C.-P. Fritzen, Data-driven multivariate algorithms for damage detection and identification: *evaluation and comparison*, *Structural Health Monitoring* 13 (1) (2014) 19–32, http://dx.doi.org/10.1177/1475921713498530.
[25] T. Nguyen, T.H. Chan, D.P. Thambiratnam, Controlled Monte Carlo data generation for statistical damage identification employing Mahalanobis squared distance, *Structural Health Monitoring* 13 (4) (2014) 461–472, http://dx.doi.org/10.1177/1475921714521270.
[26] S. Hakim, H.A. Razak, Modal parameters based structural damage detection using artificial neural networks—a review, *Smart Structures and Systems* 14 (2) (2014) 159–189, http://dx.doi.org/10.12989/sss.2014.14.2.159.
[27] T.-Y. Hsu, C.-H. Loh, Damage detection accommodating nonlinear environmental effects by nonlinear principal component analysis, *Structural Control and Health Monitoring* 17 (3) (2010) 338–354, http://dx.doi.org/10.1002/stc.320.
[28] J. Kullaa, Is temperature measurement essential in structural health monitoring?, *Proceedings of the 4th International Workshop on Structural Health Monitoring: From Diagnostic & Prognostics to Structural Health Monitoring*, DEStech Publications, Stanford, CA, USA, 1993, pp. 717–724.
[29] K. Worden, G. Manson, N.R.J. Fieller, Damage detection using outlier analysis, *Journal of Sound and Vibration* 229 (3) (2000) 647–667, http://dx.doi.org/10.1006/jsvi.1999.2514.
[30] R. Ruotolo, C. Surace, Using SVD to detect damage in structures with different operational conditions, *Journal of Sound and Vibration* 226 (3) (1999) 425–439, http://dx.doi.org/10.1006/jsvi.1999.2305.
[31] H. Kantz, T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge University Press, Cambridge, United Kingdom, 2003.

[32] E. Figueiredo, J. Figueiras, G. Park, C.R. Farrar, K. Worden, Influence of the autoregressive model order on damage detection, *Computer-Aided Civil and Infrastructure Engineering* 26 (3) (2011) 225–238, http://dx.doi.org/10.1111/j.1467-8667.2010.00685.x.
[33] R. Yao, S.N. Pakzad, Autoregressive statistical pattern recognition algorithms for damage detection in civil structures, *Mechanical Systems and Signal Processing* 31 (2012) 355–368, http://dx.doi.org/10.1016/j.ymssp.2012.02.014.
[34] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, *Time Series Analysis: Forecasting and Control*, 4th ed. John Wiley & Sons, Inc., Hoboken, NJ, United States, 2008.
[35] R. Kothamasu, J. Shi, S.H. Huang, H.R. Leep, Comparison of selected model evaluation criteria for maintenance applications, *Structural Health Monitoring* 3 (3) (2004) 213–224, http://dx.doi.org/10.1177/1475921704042696.
[36] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ACM, Pittsburgh, PA, USA, 1992, pp. 144–152. http://dx.doi.org/10.1145/130385.130401.
[37] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning* 20 (3) (1995) 273–297, http://dx.doi.org/10.1023/A:1022627411411.
[38] C.-C. Chang, C.-J. Lin, LIBSVM: *a library for support vector machines*, *ACM Transactions on Intelligent Systems and Technology* 2 (3) (2011) 1–27, http://dx.doi.org/10.1145/1961189.1961199.
[39] S.S. Keerthi, C.-J. Lin, Asymptotic behaviors of support vector machines with gaussian kernel, *Neural Computation* 15 (7) (2003) 1667–1689, http://dx.doi.org/10.1162/089976603321891855.
[40] B. Schölkopf, J.C. Platt, J.C. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, *Neural Computation* 13 (7) (2001) 1443–1471, http://dx.doi.org/10.1162/089976601750264965.
[41] B. Schölkopf, A.J. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, *Neural Computation* 12 (5) (2000) 1207–1245, http://dx.doi.org/10.1162/089976600300015565.
[42] D.M.J. Tax, R.P.W. Duin, Support vector data description, *Machine Learning* 54 (1) (2004) 45–66, http://dx.doi.org/10.1023/B:MACH.0000008084.60811.49.
[43] W.-C. Chang, C.-P. Lee, C.-J. Lin, A Revisit to Support Vector Data Description (SVDD), Technical Report, National Taiwan University, Taipei, Taiwan, 2013.
[44] B. Schölkopf, A. Smola, K.-R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* 10 (5) (1998) 1299–1319, http://dx.doi.org/10.1162/089976698300017467.
[45] B. Schölkopf, A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, Cambridge, MA, United States, 2001.
[46] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, G. Rätsch, Kernel PCA and de-noising in feature spaces, *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems 11*, MIT Press, Denver, CO, USA, 1999, pp. 536–542.
[47] V. Franc, V. Hlavac, *Greedy algorithm for a training set reduction in the kernel methods*, *Computer Analysis of Images and Patterns*, Springer, Groningen, Netherlands, 2003, 426–433, http://dx.doi.org/10.1007/978-3-540-45179-2_53.
[48] E. Figueiredo, G. Park, J. Figueiras, C. Farrar, K. Worden, Structural Health Monitoring Algorithm Comparisons Using Standard Data Sets, Technical Report LA-14393, Los Alamos National Laboratory, Los Alamos, NM, United States, 2009.
[49] C.R. Farrar, P.J. Cornwell, S.W. Doebling, M.B. Prime, Structural Health Monitoring Studies of the Alamosa Canyon and I-40 Bridges, Technical Report LA-13635-MS, Los Alamos National Laboratory, Los Alamos, NM, United States, 2000.
[50] B. Peeters, G.D. Roeck, One-year monitoring of the Z24-Bridge: *environmental effects versus damage events*, *Earthquake Engineering & Structural Dynamics* 30 (2) (2001) 149–171, http://dx.doi.org/10.1002/1096-9845(200102)30:2⟨149::AID-EQE1⟩3.0.CO;2-Z.
[51] E. Figueiredo, E. Flynn, Three-Story Buildings Structure to Detect Nonlinear Effects, Technical Report. SHMTools, Los Alamos National Laboratory, Los Alamos, NM, United States (2009).