Kernel dependence analysis and graph structure morphing for novelty detection with high-dimensional small size data set

Kernel dependence analysis and graph structure morphing for novelty detection with high-dimensional small size data set

Reza Mohammadi-Ghazi[a], Roy E. Welsch[b], Oral Büyüköztürk[a,*]

[a] *Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA*
[b] *Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA*

Abstract

In this study, we propose a new approach for novelty detection that uses kernel dependence techniques for characterizing the statistical dependencies of random variables (RV) and use this characterization as a basis for making inference. Considering the statistical dependencies of the RVs in multivariate problems is an important challenge in novelty detection. Ignoring these dependencies, when they are strong, may result in inaccurate inference, usually in the form of high false positive rates. Previously studied methods, such as graphical models or conditional classifiers, mainly use density estimation techniques as their main learning element to characterize the dependencies of the relevant RVs. Therefore, they suffer from the curse of dimensionality which makes them unable to handle high-dimensional problems. The proposed method, however, avoids using density estimation methods, and rather, employs a kernel method, which is robust with respect to dimensionality, to encode the dependencies and hence, it can handle problems with arbitrarily high-dimensional data. Furthermore, the proposed method does not need any prior information about the dependence structure of the RVs; thus, it is applicable to general novelty detection problems with no simplifying assumption. To test the performance of the proposed method, we apply it to realistic application problems for analyzing sensor networks and compare the results to those obtained by peer methods.

Keywords: Novelty detection, kernel independence, reproducing kernel Hilbert space, two-sample test, Hilbert-Schmidt independence criterion, graphical model, structural health monitoring, sensor network, video camera, camera-based measurement.

1. Introduction

From the pattern recognition point of view, novelty detection can be viewed as a one-class classification that aims to distinguish one well sampled class from all other possible classes for which the available data is insufficient to build an explicit model for the latter [1]. Important methods in this regard are the k-nearest neighbors [2], one-class support vector machines [3], neural networks [4], density estimation and clustering [5], and decision tree based techniques such as one-class random forests [6]. The data from the observed class are usually represented in terms of certain features which can be modeled as random variables (RV); and the

*Corresponding author
Email address: `rezamg@mit.edu`, `rwelsch@mit.edu`, `obuyuk@mit.edu` (Oral Büyüköztürk)

above-mentioned methods are usually most effective when the statistical dependencies of the relevant RVs are weak as these techniques ignore such dependencies. As a result these methods may provide inaccurate predictions, especially in the form of high false positive rates, in applications such as saliency detection in image processing [7], analyzing sensor networks [8], structural health monitoring and damage detection [8, 9, 10], where the dependencies of RVs are not negligible.

To consider the statistical dependencies of RVs in novelty detection, previous studies applied methods such as statistical graphical models [7, 8, 11] and conditional classifiers [10]. These techniques mainly use density estimation for characterizing the dependencies between the RVs of the problem. Noting that the density estimation techniques usually suffer from the curse of dimensionality [12], the novelty detection techniques which use density estimation may not be able to handle high-dimensional problems. Moreover, some of these techniques have limited applications due to simplifying assumptions they make or specific prior information they need about the dependence structure of RVs [8]. These issues motivate the objective of this study which is to develop suitable novelty detection algorithm with the capability of considering statistical dependencies of relevant RVs in high-dimensional problems.

To address the objective of this study we develop a kernel dependence novelty detection (KDND) algorithm that uses kernel two-sample tests [13, 14, 15, 16] and kernel independence analysis [17, 18, 19] as the basis for making inference. Our proposed KDND method aims to detect novel realizations of RVs with respect to a baseline by tracking the changes in the pairwise dependence structure of RVs. This dependence structure is learned by using a kernel dependence criterion [17, 18] that is robust with respect to dimensionality of data. By doing so, the contributions of our study can be summarized as follows: (1) We formulate a KDND classifier that is capable of considering the dependencies of RVs in arbitrarily high-dimensional novelty detection problems without any prior information about these dependencies; (2) We experimentally evaluate the proposed method in structural health monitoring (SHM) problems, one on a full scale steel structure, and compare the results with other techniques.

The paper is organized as follows. First, a review of kernel two-sample tests and kernel dependence analysis is presented in Section 3. Then, in Section 4 we describe the problem and formulate our proposed KDND method followed by an implementation of this method in Section 5. Section 6 provides a discussion about how to relax some of the assumption we made for formulating the proposed method as well as more details about the proposed formulation of the classifier. The results of the experimental evaluation of the proposed algorithm and its comparison with other methods are presented in Section 7. Finally, we conclude with a summary of our findings and a discussion of future research directions.

2. Notation

Through out this paper, the RVs are denoted by sans-serif fonts, e.g., x, and deterministic quantities such as the realizations of RVs are denoted using serifed fonts, e.g., x.

To exclude particular entries from a set we use "\" followed by the indices of the entries to be excluded. For instance, if $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$, then $\mathbf{Z} \setminus \{\mathbf{z}_2\} = \{\mathbf{z}_1, \mathbf{z}_3\}$. $|\cdot|$ is also the cardinality of a set.

Probability distributions are denoted by p with a subscript denoting the RV; e.g., p_{y} is the probability density of the RV y. The probability of an event e is denoted by $\mathbb{P}(e)$.

Finally, through out this paper, we use the *standard* as the opposite of *novel*, e.g., a standard data point with respect to a baseline distribution is the one that is drawn from that distribution. For the Gaussian distribution, we always use the term *Gaussian*.

3. Review of kernel two-sample test and kernel independence analysis

In this study, we use the kernel two-sample test and kernel independence analysis as the building blocks for formulating our proposed KDND algorithm. Therefore, we first provide a review of these methods to be used in the following sections.

A two-sample test is a statistical test to infer whether two data sets are drawn from a same probability distribution [15]. One approach to perform such a test is to compare the estimated probability density models of the two data sets. The main issue of this approach is that density estimation techniques suffer from the curse of dimensionality and hence, density-based methods are not usually robust when the size of data set is relatively small compared to its dimensionality.

An alternative approach for two-sample test is kernel-based method that maps data sets into the reproducing kernel Hilbert space (RKHS) and uses appropriate similarity measures in this space to compare the two samples [13, 14]. To further clarify this procedure, consider two RVs $\mathbf{y}_1$ and $\mathbf{y}_2$ with the fixed but unknown probability distributions $p_{\mathbf{y}_1}$ and $p_{\mathbf{y}_2}$. Letting $k(\cdot,\cdot)$ to be a universal kernel associated with the RKHS, an appropriate distance between these distributions can be defined as [13]

$$D\left(p_{\mathbf{y}_1}, p_{\mathbf{y}_2}\right) = \mathbf{E}_{\mathbf{y}_1,\mathbf{y}_1'}[k(\mathbf{y}_1, \mathbf{y}_1^{'})] - 2\mathbf{E}_{\mathbf{y}_1,\mathbf{y}_2}[k(\mathbf{y}_1, \mathbf{y}_2)] + \mathbf{E}_{\mathbf{y}_2,\mathbf{y}_2'}[k(\mathbf{y}_2, \mathbf{y}_2^{'})], \tag{1}$$

where $\mathbf{y}_1^{'}$ and $\mathbf{y}_2^{'}$ are independent copies of $\mathbf{y}_1$, and $\mathbf{y}_2$, respectively, and $\mathbf{E}$ is the expectation operator. $D(\cdot,\cdot)$, which is called the maximum mean discrepancy (MMD), basically measures the similarity of two distributions by comparing the expectation operators that are defined with respect to their distribution (See [13, 19] for more information about this similarity measure). In practice, however, we do not have access to RVs' distributions and we need to estimate the MMD using empirical data with finite samples. For such estimation, let $\mathbf{Y}_1 = \{\mathbf{y}_{11}, \ldots, \mathbf{y}_{1m}\}$ and $\mathbf{Y}_2 = \{\mathbf{y}_{21}, \ldots, \mathbf{y}_{2m}\}$ be two sets of m i.i.d. realizations drawn from $p_{\mathbf{y}_1}$ and $p_{\mathbf{y}_2}$, respectively. It can be shown that an unbiased estimate of $D(\cdot,\cdot)$ is [14]

$$\begin{aligned} D\left(p_{\mathbf{y}_1}, p_{\mathbf{y}_2}\right) \approx \widehat{D}\left(\mathbf{Y}_1, \mathbf{Y}_2\right) = & \\ & \frac{1}{m(m-1)} \sum_{j \neq j'}^{m} k(\mathbf{y}_{1j}, \mathbf{y}_{1j'}) - 2k(\mathbf{y}_{1j}, \mathbf{y}_{2j'}) + k(\mathbf{y}_{2j}, \mathbf{y}_{2j'}). \end{aligned} \tag{2}$$

where $j, j^{'} \in \{1, \ldots, m\}$.

The MMD, as it was mentioned before, is a measure of similarity between two arbitrary distributions; therefore, it can be extended to obtain a measure of dependency of RVs. To do that, first remember that two RVs $\boldsymbol{\mathsf{y}}_1$ and $\boldsymbol{\mathsf{y}}_2$ are statistically independent if their joint distribution equals the product of their marginal distributions, i.e., $p_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_2} = p_{\boldsymbol{\mathsf{y}}_1} p_{\boldsymbol{\mathsf{y}}_2}$. Therefore, comparing $p_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_2}$ and $p_{\boldsymbol{\mathsf{y}}_1} p_{\boldsymbol{\mathsf{y}}_2}$ can provide information about the dependency of $\boldsymbol{\mathsf{y}}_1$ and $\boldsymbol{\mathsf{y}}_2$. Performing this comparison via the MMD, a measure of independence between the two RVs is obtained. This independence measure is called Hilbert-Schmidt independence criterion (HSIC) and estimated as follows [13, 18]

$$\eta_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_2} = \frac{1}{m^2} tr \left(\mathbf{H} \cdot \mathbf{K}_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_1} \cdot \mathbf{H} \cdot \mathbf{K}_{\boldsymbol{\mathsf{y}}_2\boldsymbol{\mathsf{y}}_2} \right), \tag{3}$$

where $tr(\cdot)$ is the matrix trace; $\mathbf{K}_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_1} = [k(\mathbf{y}_{1j}, \mathbf{y}_{1j'})]$ and $\mathbf{K}_{\boldsymbol{\mathsf{y}}_2\boldsymbol{\mathsf{y}}_2} = [k(\mathbf{y}_{2j}, \mathbf{y}_{2j'})]$; and $\mathbf{H} = \mathbb{I}_m - 1/m$ with $\mathbb{I}_m$ to be the identity matrix of size m. The HSIC defined in (3) can be normalized as [18]

$$h_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_2} = \frac{\eta_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_2}}{\sqrt{\eta_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_1} \eta_{\boldsymbol{\mathsf{y}}_2\boldsymbol{\mathsf{y}}_2}}}. \tag{4}$$

This normalization maps $\eta_{\boldsymbol{\mathsf{y}}_1\boldsymbol{\mathsf{y}}_2}$ into the interval of $[0, 1]$, which makes it easier to interpret this independence criterion.

The MMD and normalized HSIC are used in the next section as a basis for formulating our KDND classifier.

4. Kernel dependence novelty detection classifier

Assume we have a system with n components that each can have two possible states, *standard* (intact/healthy) and *novel* (damaged), with respect to a baseline data set. Let x_i, $i \in \{1, \ldots, n\}$, be a Bernoulli RV which can take on values in $\{-1, +1\}$ such that $x_i = +1$ if the i^{th} component of the system is intact and $x_i = -1$ if the i^{th} in novel. Assume we can observe all components of the system and represent the observations from the i^{th} component via a feature vector $\boldsymbol{\mathsf{y}}_i \in \mathbb{R}^d$. Note that this feature vector is modeled as a d-dimensional RV, i.e., $\boldsymbol{\mathsf{y}}_i = [\mathsf{y}_{i1}, \ldots, \mathsf{y}_{id}]$, where $\mathsf{y}_{ik}, k \in \{1, \ldots, d\}$, is a univariate RV that represents a single feature that is extracted from the observations of the i^{th} component.

Consider the two sets $\boldsymbol{\mathsf{x}} = \{\mathsf{x}_1, \ldots, \mathsf{x}_n\}$ and $\boldsymbol{\mathsf{Y}} = \{\boldsymbol{\mathsf{y}}_1, \ldots, \boldsymbol{\mathsf{y}}_n\}$, and assume that the relationship between $\boldsymbol{\mathsf{x}}$ and $\boldsymbol{\mathsf{Y}}$ can be captured by a fixed but unknown joint probability distribution denoted by $p_{\boldsymbol{\mathsf{Y}},\boldsymbol{\mathsf{x}}}$. Let us denote the baseline training set by $\mathcal{T}$ which contains m data points, i.e., $\mathcal{T} = \left\{(\mathbf{Y}_j^{(b)}, \mathbf{x}_j^{(b)}; j \in \{1, \ldots, m\}\right\}$ where $\mathbf{Y}_j^{(b)} = \{\mathbf{y}_{1j}^{(b)}, \ldots, \mathbf{y}_{nj}^{(b)}\}$ is the set of feature vectors associated with the j^{th} baseline observation of the system, $\mathbf{x}_j^{(b)} = \{x_{1j}^{(b)}, \ldots, x_{nj}^{(b)}\}$ are the class labels for these observations, and the superscript (b) denotes *baseline*. Note that in novelty detection problems, the baseline data belongs to only one state of the system [1, 20] in which all components are intact and hence, $x_{ij}^{(b)} = +1, \forall (i, j)$. Given $\mathcal{T}$ and a set of $m^{'}$ new realizations of $\boldsymbol{\mathsf{Y}}$, denoted by $\mathcal{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{m'}\}$, the objective is to distinguish the standard and novel components of the system based on $\mathcal{Y}$, i.e., to predict $x_i, \forall i$.

Our proposed approach to satisfy the above-mentioned objective is to track the changes in the dependence structure of the RVs of the problem using kernel dependence methods. In doing so, first we consider a pairwise graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices such that vertex $i \in \mathcal{V}$ is associated with y_i, and the edges $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ represent the pairwise dependencies between each pair of RVs. For the moment, let us assume that the edge potentials of this graph are fixed and unique for each possible realization of x. Thus, we expect that the state transition of a subset of the system's components from intact to novel is reflected in the properties of $\mathcal{G}$. Therefore, finding novel components of the system can be accomplished by comparing the dependencies encoded by two such graphical models which are respectively learned from $\mathcal{T}$ and $\mathcal{Y}$. Note that the assumption about the uniqueness of the edge potential of $\mathcal{G}$ for each realization of x may not always hold. In Section 6 we discuss how to deal with the cases where this assumption is not true.

To proceed with formulating our KDND classifier, let N_i be the set of neighboring vertices of the i^{th} vertex of $\mathcal{G}$. For any edge $(i, i^{'}) \in \mathcal{E}, i^{'} \in N_i$, we use $h_{\mathsf{y}_i \mathsf{y}_{i'}}$ as an estimate of the dependence strength associated with this edge. We compute $h_{\mathsf{y}_i \mathsf{y}_{i'}}$ using (4) and denote it by $h_{ii'}$ for ease of notation. Given the above assumptions and definitions, we define classifier $C(h_{ii'})$ as

$$x_i = C(h_{ii'}) = \operatorname{sign}\left(f(h_{ii'})\right), \tag{5a}$$

$$f(h_{ii'}) = \sum_{i' \in N_i} w_{ii'} \ell_{ii'}, \tag{5b}$$

$$\ell_{ii'} = \operatorname{sign}\left(p^{(b)}_{\mathsf{h}_{ii'}}(h_{ii'}) - p^{*}_{ii'}\right), \tag{5c}$$

where $w_{ii'}$ is the weight that is (pre-)assigned to the edge $(i, i^{'}) \in \mathcal{E}$ and $\sum_{i' \in N_i} w_{ii'} = 1$. $w_{ii'}$ can be viewed as an importance factor of the $(i, i^{'})$ edge in making inferences about the i^{th} component of the system. The reason for considering these weights is to make our formulation applicable to problems where prior information is available on the dependencies of a subset of RVs. In case there is no such information, the weights can be set to $w_{ii'} = 1/|N_i|$ for $(i, i^{'}) \in \mathcal{E}$ and $\forall i^{'} \in N_i$. $p^{(b)}_{\mathsf{h}_{ii'}}(\cdot)$ is the baseline distribution of $\mathsf{h}_{ii'}$, which is the dependence strength associated with the $(i, i^{'})$ edge; and $p^{*}_{ii'}$ is an appropriate likelihood threshold. Note that in this formulation, the edge potentials are modeled as RVs and shown using sans-serif font, i.e., $\mathsf{h}_{ii'}$. The realization of this RV, that is obtained by using $\mathcal{Y}$ along with (4), is shown by serif font, $h_{ii'}$.

To find the novel components of the system, the proposed classifier starts with classifying each edge of $\mathcal{G}$ via (5c). The aim of this preliminary classification is to find the edges that encode significantly different dependence strengths, in a statistical sense, for $\mathcal{T}$ and $\mathcal{Y}$. In doing so, the new edge potentials of the graph, $h_{ii'}$, are calculated using $\mathcal{Y}$ along with (4). Then the likelihood of these potentials with respect to their baseline distribution, $p^{(b)}_{\mathsf{h}_{ii'}}$, is compared with a likelihood threshold $p^{*}_{ii'}$ as stated in (5c). The result of this preliminary classification is $\ell_{ii'}$ which is a Bernoulli RV that can take on values in $\{-1, +1\}$. $\ell_{ii'} = +1$ means that the dependencies between y_i and $\mathsf{y}_{i'}$ for the two data sets $\mathcal{T}$ and $\mathcal{Y}$ are not significantly different. The converse is true for $\ell_{ii'} = -1$ which indicates that the edge potential that is obtained from $\mathcal{Y}$ for the $(i, i^{'})$ edge is statistically different from its baseline. Note that $h_{ii'}$ in (5) is obtained by using $\mathcal{Y}$ along with (4), and $p^{(b)}_{\mathsf{h}_{ii'}}$ is learned from the training set $\mathcal{T}$. We will address the details of learning this classifier in the next section. Note that performing a likelihood test in (5c) is not the only way of classifying the edges. One can use other methods, such as off-the-shelf classifiers such as the SVM to do this task.

After classifying all edges of $\mathcal{G}$, we find the RVs which are most responsible for the discrepancies between the two graphs that are learned from $\mathcal{T}$ and $\mathcal{Y}$. The idea to identify such RVs is to find the vertices whose dependencies with their neighbors have changed the most. This task is carried out in (5b) that counts the number of those incoming edges to a given vertex i which are significantly different from their baseline, i.e., $\ell_{ii'} = -1, i' \in N_i$. This can be viewed as a voting approach where each edge can vote for the two vertices it connects. The advantage of using the voting strategy over other possible prediction methods will be discussed in Section 6. Note that the importance of the incoming edges to a given vertex may not be equal; thus, we consider $w_{ii'}$ to account for such importance and the generality of the formulation. The final step, which is carried out in (5a), is to classify the components of the system based on the majority of the votes that they have received.

Based on the above explanations, three tasks are needed for implmenting the proposed KDND classifier. These tasks are: (1) choosing an appropriate kernel and learning its parameters, (2) learning $p^{(b)}_{\mathsf{h}_{ii'}}$ from the training set, and (3) determining $p^*_{ii'}$. In what follows, we explain our solutions for each of these tasks.

5. Implementation of the proposed KDND classifier

5.1. Choice of kernel and determining its parameters

Using a universal kernel in the sense of [21] is a necessary condition for derivation of the MMD and HSIC [13, 14]. Based on the discussion provided in [22], we chose a Gaussian kernel to be used in this study. To use this kernel, consider $\mathbf{y}_{ij} = [y_{ij1}, \ldots, y_{ijd}]$ as the j^{th} realization of $\mathbf{y}_i$, and $\mathbf{y}_{i'j'} = [y_{i'j'1}, \ldots, y_{i'j'd}]$ as j'^{th} realization of $\mathbf{y}_{i'}$. Note that y_{ijk} is the k^{th} feature that is extracted from the j^{th} observation of the response of the i^{th} system's component. The dissimilarity between the two realizations $\mathbf{y}_{ij}$ and $\mathbf{y}_{i'j'}$ with respect to an isotropic Gaussian kernel is

$$k(\mathbf{y}_{ij}, \mathbf{y}_{i'j'}) = \exp\left(-\frac{\|\mathbf{y}_{ij} - \mathbf{y}_{i'j'}\|^2}{2\sigma^2}\right), \tag{6}$$

where σ is the kernel width and $\|\cdot\|$ is vector norm. In practice, the median distance between the aggregate data points can be used as an estimate of σ [17, 18, 19].

Previous studies on kernel two-sample tests usually used the above-mentioned form of the Gaussian kernel. However, the isotropic property may not effectively capture the dissimilarities of data points in multivariate problem where the distributions of the feature y_{ik}, that were defined in Section 4, are different. To address this problem, we suggest using an anisotropic Gaussian kernel with the form of

$$k(\mathbf{y}_{ij}, \mathbf{y}_{i'j'}) = \exp\left(-\frac{1}{2}(\mathbf{y}_{ij} - \mathbf{y}_{i'j'})^T \mathbf{\Sigma}^2 (\mathbf{y}_{ij} - \mathbf{y}_{i'j'})\right), \tag{7}$$

where $\mathbf{\Sigma} = \text{diag}(1/\sigma_1, \ldots, 1/\sigma_d)$ with σ_k, $k \in \{1, \ldots, d\}$, is the kernel width along the k^{th} dimension which corresponds to the k^{th} feature. The superscript T is matrix/vector transpose. Note that the kernel matrix that is obtained from (7) is strictly positive definite because $\mathbf{\Sigma}^2$ is diagonal with positive diagonal elements; therefore, this anisotropic kernel is universal [23, 24].

The anisotropic kernel in (7) considers a unique kernel width for each dimension of the multivariate RVs y_i and $\mathsf{y}_{i'}$ and hence, it is expected to better capture the shape of data. However, the trade-off of using such kernel is to learn d parameters instead of one. More information about anisotropic Gaussian kernels and learning its parameters for various applications can be found in [25, 26, 27]. In this study, we propose using the median distance of aggregate data points along the k^{th} direction as an estimate of σ_k. This proposition basically uses the conventional method suggested in [17, 18, 19] for each feature individually to estimate its corresponding kernel width.

5.2. Learning $p^{(b)}_{\mathsf{h}_{ii'}}$

Our approach here is to generate multiple realizations of $\mathsf{h}_{ii'}$ and learn its distribution accordingly. For a given $(i, i') \in \mathcal{E}$, consider $(\mathbf{y}^{(b)}_{ij}, \mathbf{y}^{(b)}_{i'j'}), \forall (j, j')$, which are all permutations of training sample points for the two vertices i and i'. The dissimilarity between each pair of data points corresponding to a unique (j, j') with respect to an anisotropic Gaussian kernel can be obtained using (7). The calculated dissimilarities for all pairs of (j, j') can be assembled in an $m \times m$ kernel matrix $\mathbf{K}_{ii'}$. The procedure can be followed to assemble $\mathbf{K}_{ii}$ and $\mathbf{K}_{i'i'}$ for (i, i) and (i', i'), respectively. Using these matrices along with (3) and (4) provides a single realization of $\mathsf{h}_{ii'}$, whereas we need multiple such realizations for learning $p^{(b)}_{\mathsf{h}_{ii'}}$.

In order to generate multiple samples for $\mathsf{h}_{ii'}$, we use bootstrap aggregation (bagging). For doing so, we randomly pick $\widetilde{m}$ number of training data points. The dissimilarities between the chosen data points can be determined using (7), and assembled to form new kernel matrices which are called sampled kernel matrices and denoted by $\widetilde{\mathbf{K}}_{ii'}$, $\widetilde{\mathbf{K}}_{ii}$, and $\widetilde{\mathbf{K}}_{i'i'}$ for $\mathbf{K}_{ii'}$, $\mathbf{K}_{ii}$, and $\mathbf{K}_{i'i'}$, respectively. Note that, for computational efficiency, the sampled kernel matrices can be formed by finding and assembling the corresponding elements of $\mathbf{K}_{ii'}$, $\mathbf{K}_{ii}$, and $\mathbf{K}_{i'i'}$ to the randomly chosen data points. Using the sampled kernel matrices along with (3) and (4), a new realization of $\mathsf{h}_{ii'}$ is obtained. By running this procedure multiple times, a set of realizations for $\mathsf{h}_{ii'}$ can be generated. Finally, we use Gaussian mixture models (GMM) to learn $p^{(b)}_{\mathsf{h}_{ii'}}$ from the generated realizations of $\mathsf{h}_{ii'}$. Note that the curse of dimensionality is not a problem in using GMM for learning $p^{(b)}_{\mathsf{h}_{ii'}}$, because it is a univariate distribution and we can generate as many realizations of $\mathsf{h}_{ii'}$ as needed through the bagging procedure.

5.3. Determining $p^*_{ii'}$

If there exists a prior distribution for the novel realizations of Y, the parameter $p^*_{ii'}$ can be determined accordingly. Otherwise, in the absence of such prior information, hypothesis testing can be used. In this case, $p^*_{ii'}$ is the likelihood threshold associated with a $100(1-\alpha)\%$ confidence bound for $p^{(b)}_{\mathsf{h}_{ii'}}$, where α is a predefined significance level. Note that the notion of confidence interval for non-symmetric or multi-modal distributions is controversial [28]. Thus, we propose determining $p^*_{ii'}$ by using the notion of a high density region (HDR), which is described as follows.

For a probability distribution $p_{\mathsf{z}}(z)$ of RV z, the $100(1-\alpha)\%$ HDR is defined as the subset $R(p^*_\alpha)$ of the sample space of z such that [29]

$$R(p_\alpha) = \{z : p_{\mathsf{z}}(z) \geq p^*_\alpha\}, \tag{8}$$

where p^*_α is the largest value such that $\mathbb{P}\left(\mathsf{z} \in R(p^*_\alpha)\right) \geq 1-\alpha$. This is schematically shown in Figure 1 for a univariate distribution. By applying this concept to our problem, $p^*_{ii'}$ becomes the $100(1-\alpha)\%$ HDR of $p^{(b)}_{\mathsf{h}_{ii'}}$.

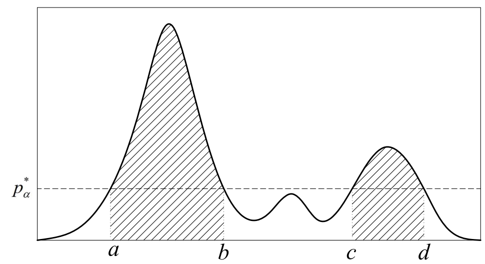


Figure 1: The $100(1-\alpha)\%$ HDR for z is $R(p^*_\alpha) = [a,b] \cup [c,d]$ if $\mathbb{P}\left(\mathsf{z} \in R(p^*_\alpha)\right) \geq 1-\alpha$ [29]

6. Discussions

In this section, we clarify how to solve the problem when the main assumption we made in Section 4 do not hold. We also discuss alternative approaches for edge classification instead of density estimation, the reasons for using the averaging in (5b), and how to use this classifier when the data sets are unbalanced.

6.1. Using the KDND method when the main assumptions do not hold

To formulate the proposed KDND method we assumed that the changes in the state of some components of the system, from standard/intact to novel, affect the edge potentials of $\mathcal{G}$. This assumption may not always be true as it is theoretically possible that there exist special state transformations which change the distributions of the relevant RVs in such a way that their dependencies are retained unaffected. Although such special cases are unlikely in practice due to noise and complexities of the systems being monitored, we address a solution for this issue to ensure the robustness of our method.

For the situation that was described above, using the HSIC is no longer effective in detecting the novel realizations of the RVs. But, we can still use MMD as suggested in section 7.5 of [14] to detect any changes in the marginal distributions of the RVs. MMD, as described in this reference, is capable of detecting the novel realizations of RVs without suffering from the curse of dimensionality; however, this method may result in high false positive rates due to ignoring the dependencies of RVs. Thus, we suggest performing a preliminary classification using the MMD to ensure capturing the state changes in the system. Once some state transformations are detected in this preliminary analysis, we can use the proposed KDND algorithm to improve the accuracy of our predictions.

6.2. Alternative methods for learning $p^{(b)}_{\mathsf{h}_{ii'}}$

As was explained in section 4, we use likelihood tests in (5c) to classify each edge of $\mathcal{G}$. This requires learning $p^{(b)}_{\mathsf{h}_{ii'}}$ for which we proposed using GMM; however, this is not the only approach for classifying the edges. In fact, any novelty detection classifier can be trained using the realizations of $\mathsf{h}_{ii'}$, that are obtained via bagging (see section 5.2), and used instead of the likelihood test in (5c).

6.3. Reason for using the voting strategy in (5b)

In the formulation of our proposed KDND classifier, $\ell_{ii'}$ is a binary RV which is the vote of edge $(i, i^{'}) \in \mathcal{E}$ for the i^{th} and $i^{'th}$ vertices. An alternative for this voting strategy would be the direct use of the likelihood difference, $p^{(b)}_{\mathsf{h}_{ii'}}(h_{ii'}) - p^{*}_{ii'}$, in our predictions. Due to the inverse relationship of the HSIC's magnitude with its variance [14], making inferences based on the explicit value of $p^{(b)}_{\mathsf{h}_{ii'}}(h_{ii'})$ results in the dominance of those edges of $\mathcal{G}$ whose potentials are small. This is equivalent to assigning more weight to the weakly dependent RVs in the process of decision making. By doing so on the extreme case, we only consider the RVs which are almost independent, and this is not consistent with the objective of this study. For more information about the variance analysis of the MMD and HSIC the readers are referred to [14]. For an example showing the behavior of these two measures in comparing Gaussian RVs see Appendix Appendix A.

6.4. Dealing with unbalanced data sets

It is common in practice that the baseline and test data sets are not equally sized, i.e., $m \neq m^{'}$. This results in different error rates in computing the dependence strength of RVs for $\mathcal{T}$ and $\mathcal{Y}$ [14]. Herein, we provide two solutions to get around this problem. The first solution is applicable if $m > m^{'}$ and can be applied to the bagging procedure for learning the underlying distribution of $\mathsf{h}_{ii'}$. To do that, we suggest simply choosing $\widetilde{m} = m^{'}$ for each bagging iteration. Otherwise, if $m^{'} > m$ or the sizes of data sets are too small, we can use over sampling methods such as the synthetic minority over sampling technique [30].

7. Experimental evaluation

To evaluate the efficacy of the proposed method, we applied it to realistic problems of analyzing sensor network data in SHM applications. In vibration-based SHM, the dynamic behavior of structures is used as a basis for evaluating their health and safety. For this purpose, their vibrational responses are measured and the measurements are compared with a baseline that represents the intact state of the system. This baseline can be a set of theoretical facts about the behavior of structures or an empirical data set that has been provided from the same structure at the intact or reliable state. Noting that there may be an infinite number of damaged states for a structure, the SHM problem clearly fits the novelty detection framework [31, 32].

One of the main goals of this comparison is to detect and localize damages in the structure [33]. In recent years, statistical methods and machine learning techniques have assisted the researchers in achieving this goal [31, 32, 34, 35]. SVM [36], clustering techniques [37], and deep learning [38] are examples of such methods.

However, the predictions from most of these techniques are inaccurate if the size of data set is small, the relevant RVs are strongly dependent, or the dimensionality of the data is extremely large [12]. Our proposed KDND classifier has been specifically designed for handling these issues and hence, can be considered as an approach to deal with real-world SHM applications.

In the following two sections, we present the application of our proposed method in monitoring a plate structure and a full scale steel structure. In each of these sections we first describe the experimental setup and the data acquisition system we used to measure the response of the structures. Then, we show the damage localization result of the proposed KDND algorithm, and its comparison with peer novelty detection methods.

7.1. Plate structure

7.1.1. Experimental Setup

This structure was a steel plate with dimensions of 60 cm $\times$ 5.08 cm $\times$ 0.64 cm, fixed to a massive concrete base using four bolts. The experimental setup for measuring the plate involved a shaker that is attached to the top of the plate, high-speed camera, and extra lighting, as shown in Figure 2. The video camera and the extra lighting are used along with the phased-based optical flow approach [39, 40] to extract the displacement field of the plate from video.

A summary of how the displacement extraction algorithm works is as follows. The complex steerable pyramid filters are used to obtain the local amplitude and phase at multiple orientations and physical length scales [41]. As was shown in previous works [41], the motion of constant phase contours corresponds to motion signals in the video. Therefore, we use the decomposed local phase signals to calculate the displacement signal at every pixel in the video. More details about the video decomposition and the calculation of displacements is contained in [42]. Note that there are few assumptions in calculating the displacement signal from objects in video. The first assumption is that the motion must be small and on the order of one pixel or smaller; otherwise, the local filters stop working. Secondly, displacements are only well defined at edges or textured regions in the video. In order to satisfy this condition, we applied a speckled pattern on the steel plate by first painting it white and then spraying a random pattern of black paint on the plate in a spotty manner. The last assumption for the measurement is that the lighting is constant. Flickering lights, such as fluorescent lighting, introduce an apparent motion signal into the video with the same variation in time as the lighting. To prevent this from happening, the object under test is flood illuminated with several bright battery powered lamps so that the lighting stays consistent.

To run the tests, the shaker excited the plate with a white Gaussian noise waveform in a horizontal direction in the video. After allowing some time for the excitation to reach steady state, 3.5 seconds of video were recorded at 2000 frames per second with a resolution of 1736$\times$244 pixels using the high speed camera. This was repeated for the damaged plate with a machined crack towards its base. The details of the intact and notched plates, and screenshots from the input videos are shown in Figure 3. To extract the displacements of the plate, 100 pixels are chosen, on a grid of 5 $\times$ 20, as the pseudo sensor locations on this structure.

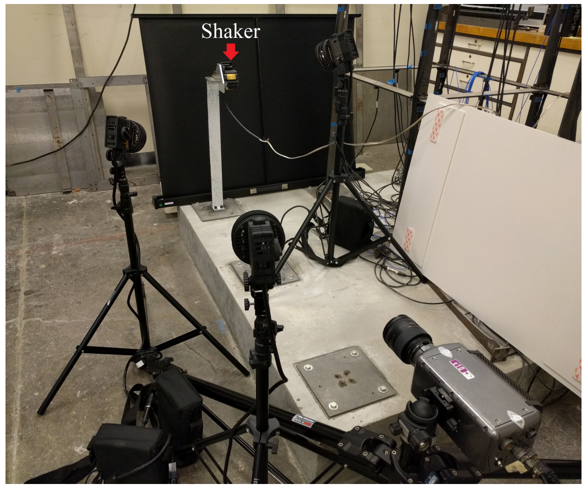


Figure 2: Picture of the experimental setup for the plate, showing the plate fixed to a concrete base, the shaker bolted to the top of the plate, high-speed camera, and extra lighting

The displacements of these locations were calculated from the recorded video using the phase-based method described above. Each pixel's displacement time history was windowed in time to produce 22 pseudo tests that each consisted of 800 time points. The reason for dividing one measurement into multiple windows is that in the real world it is unlikely to have multiple extreme events in civil structures. However, we may have few extreme events that last for about 10 to 20 seconds or even more.

7.1.2. Damage sensitive features and evaluation criteria

The Fourier coefficients in the frequency range of 1 to 400 Hz are used as the damage sensitive features in this example. As a result, the dimensionality of the feature space becomes 400 while we have only 22 pseudo tests. Many of the widely used novelty detection algorithms, such as the one-class SVM, cannot be learned for such a data set in which the number of features significantly exceeds the number of sample points. Therefore, to evaluate the efficacy of the proposed KDND method, its classification results were compared to a one-class gradient boosting algorithm [6, 12] which is robust with respect to dimensionality of data sets.

For comparing the results of the proposed method and peer methods we used the *false positive reduction* and *true positive improvement* criteria which were defined in [8]. The definition of false positive (FP) and true positive (TP) are controversial in SHM application as it is unlikely that the damage occurs exactly at the sensor locations. Therefore, the FP for SHM is defined in [8] as the detection of damage at sensor locations which may not coincide with the damage location nor the closest neighboring sensor locations. The TP is also defined as detecting the damage at the sensor location where the damage is located or its closest neighboring sensor locations. Assume $I = \{1, \cdots, n\}$ with n to be the number of sensors in a network, D_a is the set of sensor locations which are either at the damage location or its closest neighborhood, and D_a^c is the complement of D_a with respect to I. Also assume that D_k and D_g are the set of damaged locations which are detected by the proposed KDND method and the alternative novelty detection method, respectively. Then,

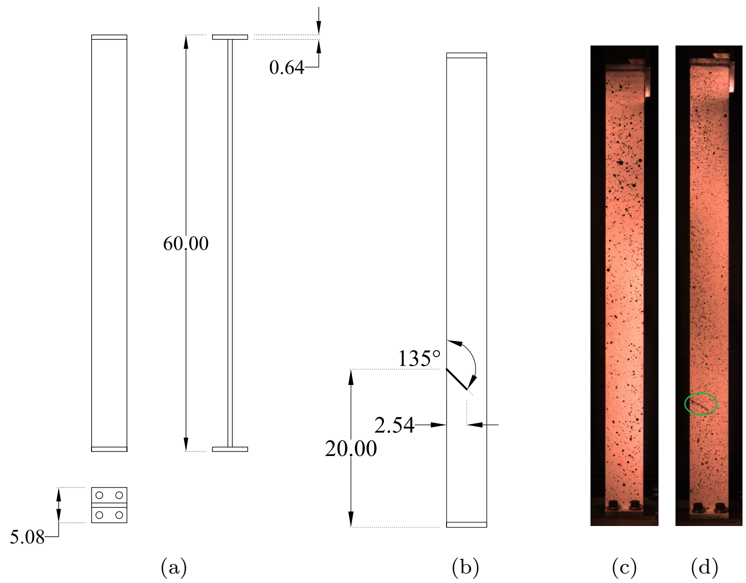


Figure 3: Details of the plate structures (dimensions are in centimeters): a) detail of the plates, b) location of the machined crack on the damaged plate, c) screenshot of input video for the intact plate, (d) screenshot of input video for the damaged plate with crack

for a new measurement the FP reduction (FPR) and TP improvement (TPI) criteria are defined as

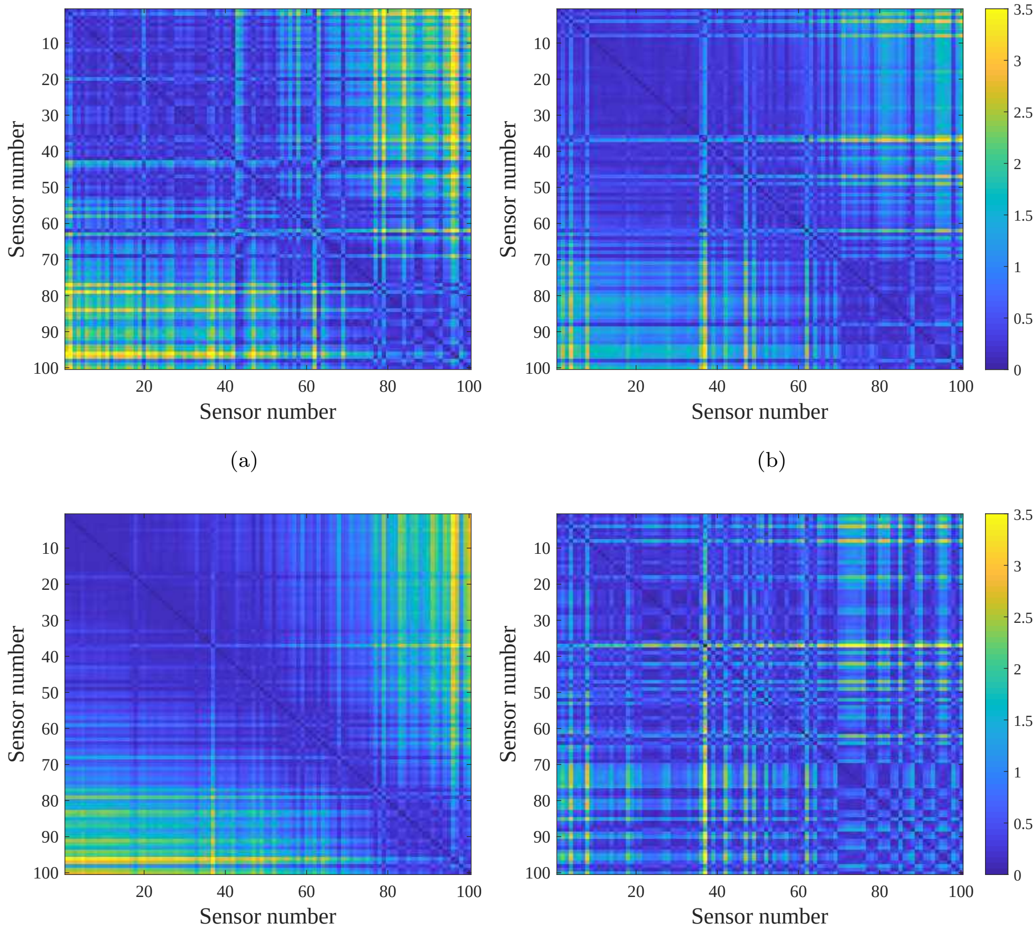
$$FPR = \frac{|D_g \cap D_a^c| - |D_k \cap D_a^c|}{|D_a^c|} \tag{9a}$$

$$TPI = \mathbf{1}_{D_k \cap D_a \neq \emptyset} - \mathbf{1}_{D_g \cap D_a \neq \emptyset} \tag{9b}$$

where $|\cdot|$ is the cardinality of a set, and $\mathbf{1}_{(\cdot)}$ is the indicator function. FPR essentially shows how much the FP is reduced by using the proposed method compared to the gradient boosting algorithm. TRI shows which algorithm can/cannot localize the damage.

7.1.3. Fitting the KDND models and damage detection results

The graph $\mathcal{G}$ in this example is considered as a fully connected pairwise graphical model in which, a generic edge $(i, i^{'})$ encodes the dependencies of the displacement responses at the i^{th} and $i^{'th}$ pseudo sensor locations. Note $\mathbf{y}_i, i \in \{1, \cdots, 100\}$, which is the feature vector associated with the response of the i^{th} pseudo sensor is a 22×400 matrix in this experiment. Using these vectors, we followed the procedure described in Section 5 to learn the KDND model. Due to the lack of prior information about the dependencies of the sensor measurements, the edge weights are considered to be $w_{ii'} = 1/|N_i|$ for $(i, i^{'}) \in \mathcal{E}$, $\forall i \in \mathcal{V}$ and $\forall i^{'} \in N_i$. We also chose an anisotropic Gaussian kernel which results in computing 400 different kernel widths for each edge of the graph. Figure 4 shows the kernel width for four of the features (out of 400) and all edges. The kernel width between sensors i and j is colored at the intersection of the i^{th} row and the j^{th} column of these plots. Note that these graphs are symmetric because there is no difference between the edges (i, j) and (j, i) for computing the kernel width. Also, the diagonal entries are zero in these plots, because we do not consider self-looping in $\mathcal{G}$ and hence, we do not compute the associated kernel width. These plots imply that the kernel width can be quite different for different features and different subsets of the edges.



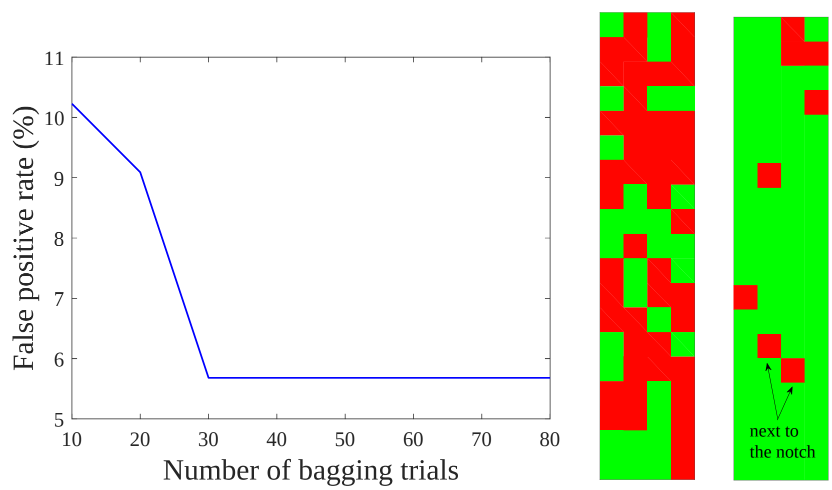
(a) (b) (c) (d)

Figure 4: Kernel width for (a) first feature, (b) 10^{th} feature, (c) 200^{th} feature, (d) 100^{th} feature.

For learning $p^{(b)}_{\mathsf{h}_{ii'}}$, we first set $\alpha = 0.05$ as the HDR significance level for decision making and then started fitting KDND models with various bagging trials starting from 10 and increased this number until the prediction error converges. The result of this procedure is shown in Figure 5(a). It follows from this plot that the FP rates does not change after the bagging trials reach 30.

In this case study we chose the one-class gradient boosting method as the alternative method to be compared to the KDND. The main reason for choosing the gradient boosting method in this part is the robustness of decision tree based algorithms with respect to the dimensionality of learning problems. Also, most of other widely used novelty detection algorithms, such as the one-class SVM, are not applicable to this application example due to the high dimensionality of the feature space. To learn the one-class gradient boosting classifier, we followed the approach that is explained in [6, 12].

The damage localization result of the proposed algorithm and its comparison with the predictions of the one-class gradient boosting algorithm are shown in Figures 5(b) and 5(c). In Figure 5(b), the average classification results of the gradient boosting algorithm for the 22 pseudo tests on the damaged structure are color coded. The red and green colors in these plots are, respectively, used to show whether a pseudo sensor location is predicted as damaged or intact. It follows from the results shown in Figure 5 that both algorithms can detect the damage, but their damage localization accuracies are significantly different. The localization results of the gradient boosting algorithm are almost inconclusive, while the proposed KDND algorithm can perfectly localize the damage by detecting two pseudo sensor locations right above the notch. The proposed algorithm also has some false positives, mainly at the top of the plate. These false alarms might have been due to the loss of the speckled pattern in that zone on the plate and hence inaccurate displacement extraction, especially since the lightening is weaker at the top of the plates compared to the other areas (see Figure 3(c)). This suggests that more attention needs to be paid when using video-based measurements in practice as this method is quite sensitive to lightening. Comparison of the two algorithms using the criteria defined in Section 7.1.2 shows that the proposed algorithm reduces the FP rate by 24% while keeping the TP rate the same. This means that by using the KDND method we localize the anomalies more accurately without sacrificing the TP.



(a) (b) (c)

Figure 5: KDND training and damage localization results; (a) Variation of false positive rates versus the number of bagging trials, (b) damage detection results of gradient boosting, (c) damage detection results of the proposed KDND algorithm. In (a) and (b), the colors show the binary classification results where red and green colors at a pseudo sensot location, respectively, means that location is predicted as damaged or intact.

7.2. Full scale steel structure

7.2.1. Experimental Setup

The full scale structure that we study in this chapter is the upper half of a truncated telecommunication tower. The dimensions of this tower are $83 \times 83 \times 239$ inches and it consists of L-shaped steel elements with bolted connections as shown in Figure 6. The structure has three stories, which are all similar in shape and configuration, and a cab floor which is shown in Figure 6(c). The beams are connected to the columns using one bolt at each side. The bracing-column connections are also made by one bolt, but the connection between a pair of bracings when they cross are made using three bolts as shown in Figures 6(d) and 6(e). The four faces of the structure are identical; therefore, we have named them as A, B, C, and D to be used when addressing damage scenarios. These faces and the global coordinate system are shown in Figure 6(c).

To measure the dynamic response of the structure it was instrumented by 48 triaxial MEMS accelerometers, as well as two shakers at its top corner for generating excitation along the two perpendicular axis of the structure (Figure 6). The maximum sampling rate of the MEMS sensors was 2 kHz, and the shakers could generate white Gaussian noise excitation in the frequency range of 5 to 350 Hz. The sensor network and the shakers were controlled via a central computer and a data acquisition system that were placed in a trailer next to the structure. Note that the structure was symmetric, but torsional modes could be excited due to the placement of the shakers.

7.2.2. Experimental tests and damage scenarios

We considered five different damage scenarios in addition to testing the intact structure. The first damage scenario was introduced by loosening a bolt in a beam-column connection at the location shown in Figure 7(a). The second damage scenario was introduced by replacing a reduced cross section element with one of the diagonal elements of the tower. Figures 7(b) and 7(d), respectively, show the location of this damage scenario and the reduced cross section element. To build the reduced cross section element, both flanges of a diagonal element were machined to reduce the element's cross section throughout its length. The third damage scenario was made by taking out the reduced cross section element from the structure. Basically, the location of this damage was the same as in the second scenario, but for the third case, the element was taken out. The fourth and fifth damage scenarios were simulating the presence of multiple damages on a structure. For these damage scenarios we removed the beams that are shown in Figure 7(c) from face C of the structure, and combined this scenario with the second and the third scenarios. Table 1 summarizes the tested damage scenarios on this structure.

To establish a baseline data set for the intact state, the structure was tested four times. In each test, the structure was excited under a pink noise excitation with the spectrum that is shown in Figure 8 for two minutes. Then, each test was segmented, by windowing with overlaps, to generate 14 pseudo tests. The reason for this segmentation of the structural response was to model real world scenario, as it was explained in Section 7.1.1. Each damage scenario was also tested two times with the same manner as testing the intact structure. Therefore, we provided 56 and 28 pseudo tests, respectively, for the intact and each damage

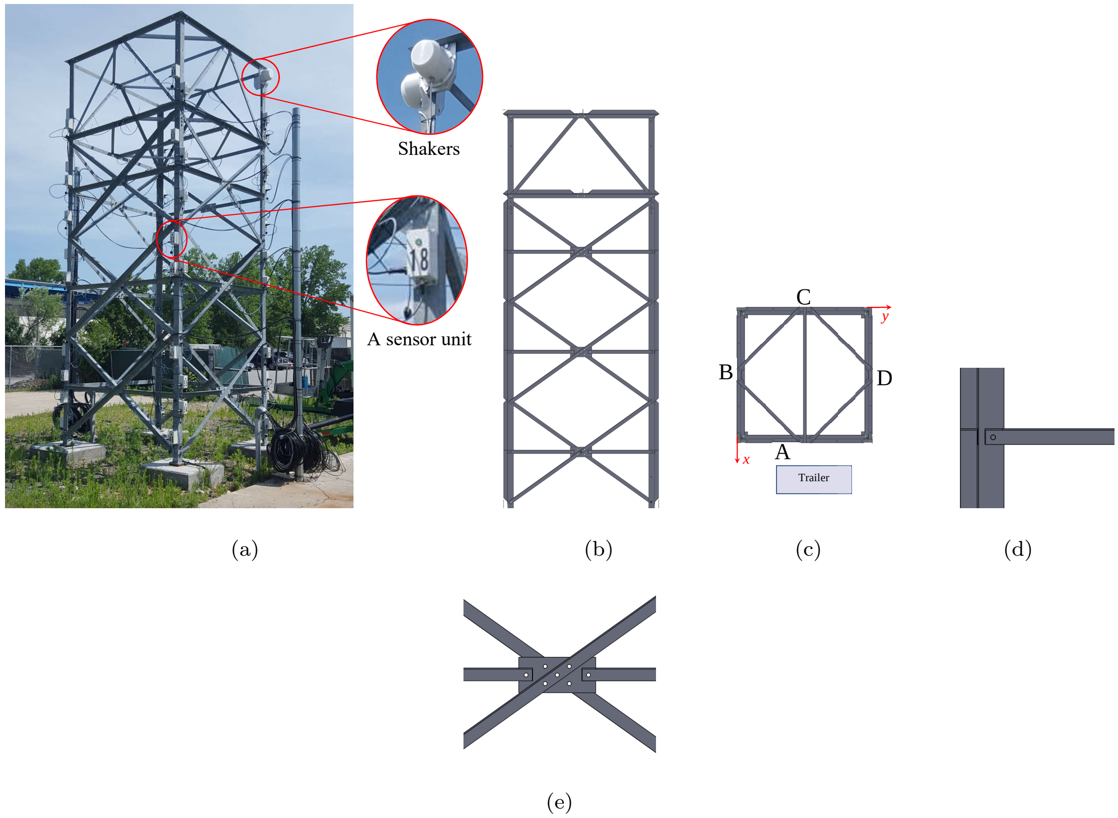
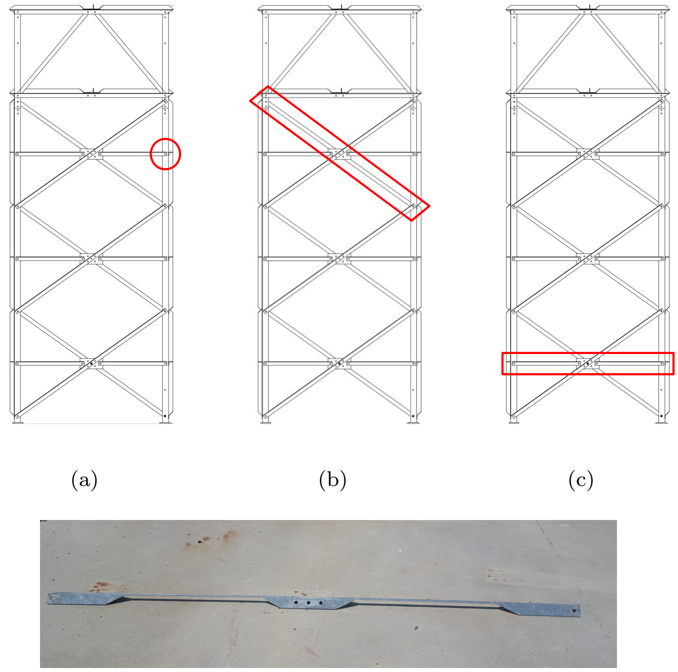


Figure 6: Large scale structure: (a) structure, sensor units, and shakers, (b) side view, (c) roof plan of the top floor, the location of the trailer with respect to the structure, and the name of each of its faces, (d) beam-column connection, (e) bracing connection .

scenarios. For damage sensitive features, we considered the autoregressive (AR) coefficients. To determine the order of the AR model, we followed the AR model selection approach using the Akaike information criterion (AIC) that is suggested in [43]. Figure 9 shows the variation of the AIC for different AR model orders. Each line in this plot corresponds to the AR model selection result of a specific sensor. It follows from these results that the variations of the AIC are negligible for model orders that are larger than 16 for all sensors' measurements; thus, we choose this number as the AR model order in our study. The weights $w_{ii'}$ were chosen similar to the previous experiment due to the lack of prior information about the dependencies of sensor measurements, i.e. $w_{ii'} = 1/|N_i|$ for $(i, i^{'}) \in \mathcal{E}$, $\forall i \in \mathcal{V}$ and $\forall i^{'} \in N_i$ For comparing the performance of the KDND with other methods on this structure we use the evaluation criteria that were defined in section 7.1.2.

7.2.3. Detection results and comparison

For learning the KDND, we followed the same procedure that was explained in section 7.1.3 for the plate structure. For evaluating the performance of the proposed method on this structure, we compared the KDND with the one-class gradient boosting, one-class SVM, and one-class clustering via GMM. We followed [44, 3] for learning the SVM models and [5] for learning the clustering model. The detection results of these



(a) (b) (c)

(d)

Figure 7: Location of damages on the structure. (a) location of the loosened bolt at the beam-column connection on face A, (b) location of the reduced cross section element on face A, (c) location of the removed beam on face C of the structure for damage scenarios 4 and 5, (d) the reduced cross section element

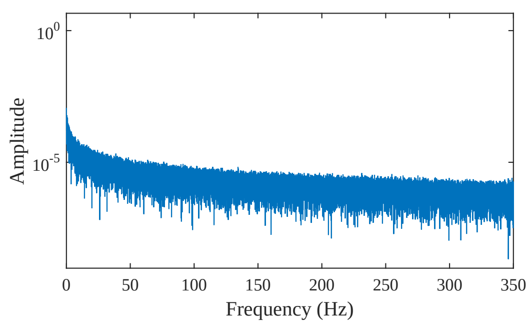


Figure 8: The spectrum of the pink noise excitation

algorithms for the first damage scenario are illustrated in Figure 10. The sensor locations are shown with circles in the plots of this figure. In Figure 10(a) the actual location of damage is marked by a red circle, and the black lines and circles, respectively, show the intact elements and sensor locations. Figures 10(b) to 10(e) show the detection results of the four above-mentioned methods. The red and blue circles in these plots are the sensor locations which are predicted as damaged and intact, respectively. It follows from these plots that all techniques can detect the damage; however, the localization results are significantly different. Basically, the localization with the gradient boosting is inconclusive. The results of SVM and clustering are

Table 1: Summary of the tested damage scenarios

Scenario No.	Description
1	Bolt loosening at a beam-column connection at the location shown in Figure 7(a) on face A
2	Reduced cross section element at the location shown in Figure 7(b) on face A
3	Element removal at the location shown in Figure 7(b) on face A
4	Multiple damage scenario by combining scenario #2 and element removal at the location shown in Figure 7(c) on face C
5	Multiple damage scenario by combining scenario #3 and element removal at the location shown in Figure 7(c) on face C

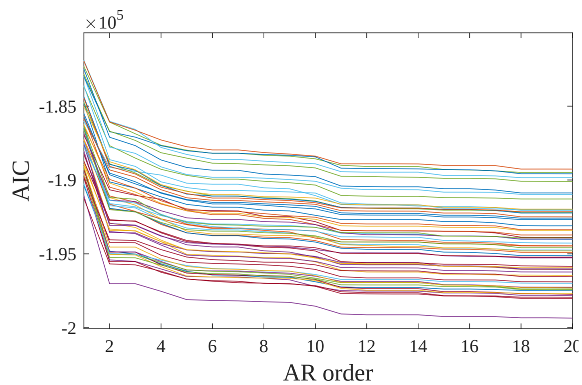
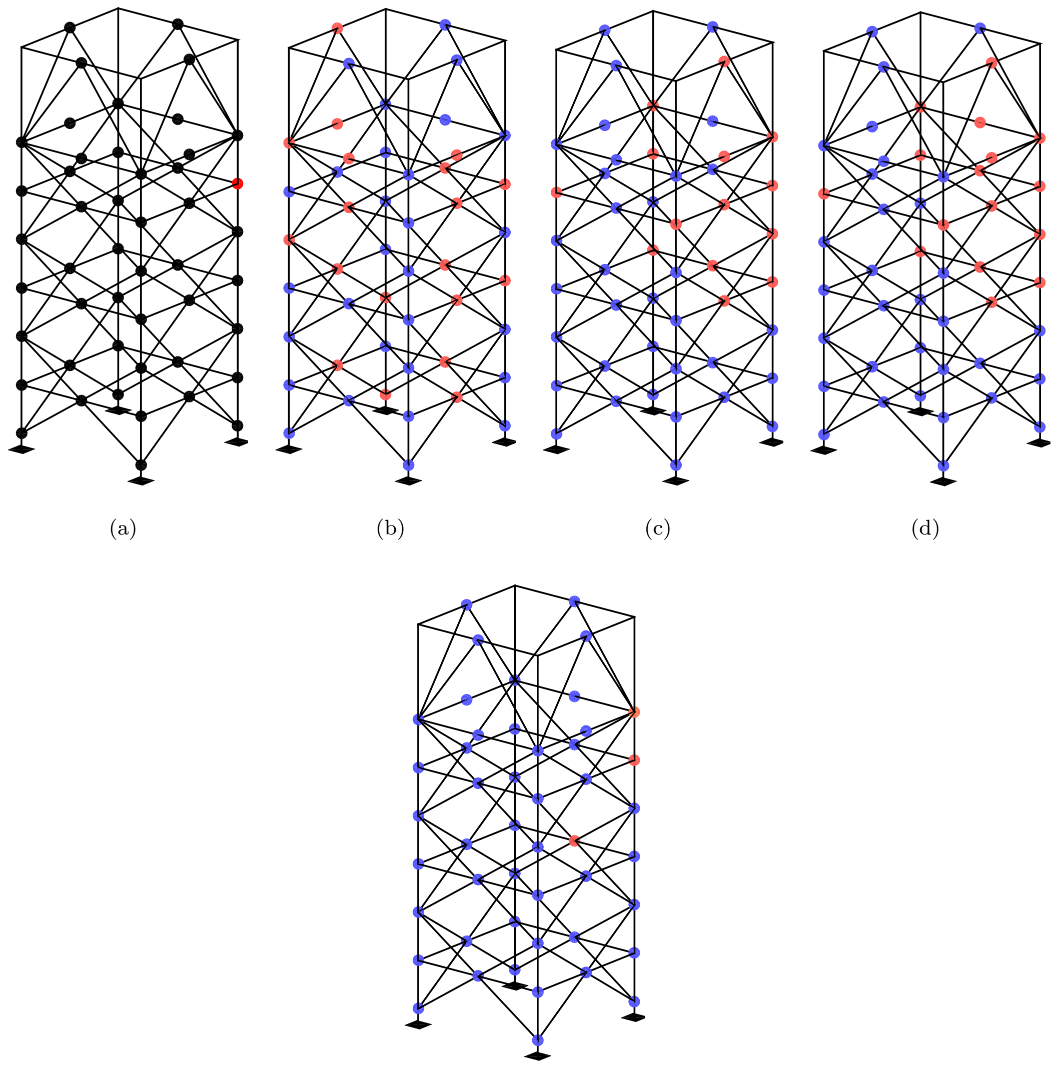


Figure 9: Variation of AIC for different AR model orders for the vibration response at all sensor locations. There are 56 lines in this plot and each line is associated with one sensor location.

more concentrated around the true location of damage, but these techniques also suffer from high false rates. The KDND, on the other hand, provides a more accurate localization result by detecting the true location of damage and one sensor location that is directly connected to the damaged location. This method has only one false detection that is two elements away from the actual location of damage. The predictions of these algorithms for other damage scenarios follow a similar pattern; thus, we skip showing the plots for brevity, and instead, summarize the results in Table 2. It should be noted that all algorithms can correctly detect the damage in all scenarios; therefore, we only report the false positive reduction rates in this table. The information in this table implies that considering the dependencies of RV via the KDND method can effectively reduce the false positive rates between 14% to 33% without affecting the true positive rates.

8. Conclusion

In this paper we have proposed a novelty detection method that uses kernel dependence analysis for considering the statistical dependencies of the problem's RVs to make predictions. The method considers a pairwise graphical model over the RVs and aims to detect statistically significant variations in the parameters



(a) (b) (c) (d)

(e)

Figure 10: Damage detection/localization results for the bolt loosening damage scenario. The red color shows the damaged locations and the blue is used for the intact ones: (a) actual damage location, (b) predictions of one-class gradient boosting algorithm, (c) prediction of one-class SVM, (d) predictions of clustering technique using GMM, (e) predictions of our proposed KDND algorithm.

of this graph as a result of changes in the characteristics of those RVs. The main advantage of graph structure morphing using the kernel dependence technique is its robustness with respect to the dimensionality of the data sets. Therefore, the proposed KDND method is applicable to arbitrarily high dimensional data sets.

The experimental results of applying the proposed algorithm to realistic SHM application problems shows that considering the dependencies of the relevant RVs by tracking their dependence structures can potentially yield more accurate classification results compared to traditional classification based on tracking the changes in the marginal distributions of the RVs. Followed by the results, the KDND method reduced the false positive rates between 14% and 33% compared to peer techniques such as gradient boosting method, which

Table 2: Comparing the performance of KDND with one-class gradient boosting algorithm, one-class SVM, and clustering method using the false positive reduction criteria that was defined in section 7.1.3

	Damage scenarios				
FP reduction compared to (%)	1	2	3	4	5
one-class gradient boosting	32.5	33.3	36.1	29.6	29.6
one-class SVM	20.9	16.6	19.4	14.8	14.8
one-class clustering	20.9	22.2	14.8	22.2	18.5

is another robust method with respect to the dimensionality of data sets, one-class SVM, and clustering via GMM.

The main trade-off of using the proposed technique over the alternative methods is its higher computational demand as a result of computing kernel matrices and iterative operations on such matrices. Also, only a few dimensionality reduction methods, such as random subspace feature selection which are also usually demanding, can be used along with the KDND classifier. Moreover, the formulation of the KDND algorithm requires a fixed set of features to be used for all RVs. This can be viewed as an additional constraint when it comes to feature selection. Thus, developing specific feature selection techniques for the KDND classifier, and similar techniques that impose the same constraint, can be pursued in future studies.

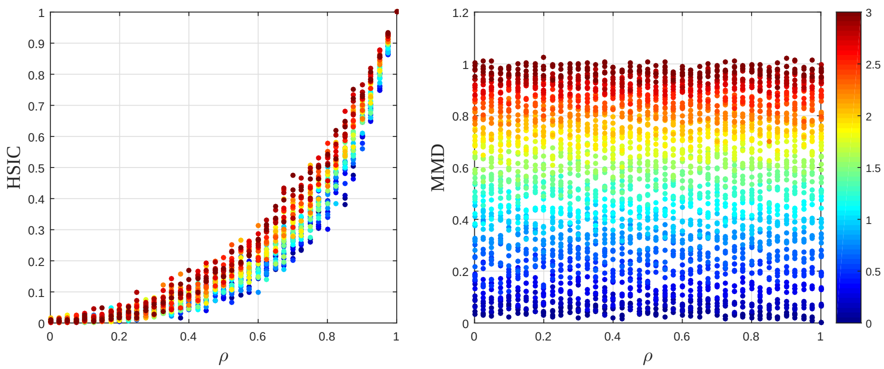
9. Acknowledgment

The authors acknowledge the support provided by Royal Dutch Shell through the MIT Energy Initiative, and thank chief scientists Dr. Dirk Smit and Dr. Sergio Kapusta.

Appendix A. MMD and HSIC for comparing Gaussian RVs

Consider two Gaussian RV $\mathbf{z}_1$ and $\mathbf{z}_2$, where $\mathbf{z}_1 \sim \mathcal{N}(0,1)$ and $\mathbf{z}_2 \sim \mathcal{N}(\mu,\sigma)$, and $\mathcal{N}(.,.)$ denotes Gaussian RV with the mean and variance as its first and second arguments, respectively. The reason for considering Gaussian RVs is that their correlation coefficient is an exact measure of their dependency. To study the variations of HSIC and MMD for Gaussian RVs, we keep the parameters of $\mathbf{z}_1$ unchanged while changing the parameters of $\mathbf{z}_2$. For each new set of parameters of $\mathbf{z}_2$ we draw two sets of samples from $\mathbf{z}_1$ and $\mathbf{z}_2$ and compute their associated HSIC and MMD.

We consider 41 linearly spaced values between 0.0 and 3.0 for changing μ. For each new value of μ, we use 20 different values, which are linearly spaced between 0.0 and 1.0, as the correlation coefficient between $\mathbf{z}_1$ and $\mathbf{z}_2$. For a desired correlation coefficient, we adjust σ accordingly. Thus, the new parameter of $\mathbf{z}_2$ can sit on a grid of 41×20. For each pair of (μ,σ) on this grid, we run 30 simulations by sampling from the distributions of $\mathbf{z}_1$ and $\mathbf{z}_2$ with its new parameters and computing the associated HSIC and MMD of the two RVs. The result of these simulations is shown in Figure A.11. It follows from this figure that the HSIC is



(a) (b)

Figure A.11: Variations of (a) HSIC and (b) MMD as a result of changing the mean value and correlation coefficient of two Gaussian RV. The distance between the mean values of these RV is color coded such that dark blue is used for zero distance between the mean values, red is used for a shift of mean value of 3, and other colors are in between these extreme cases.

monotonically changed from zero to unity as the correlation coefficient, ρ, varies from zero to one; however, the independence criterion is not as sensitive to the shift of mean values of the Gaussian RVs. In contrast to HSIC, the MMD is capable of capturing the translational discrepancy between two Gaussian clusters, while it is not sensitive to the change of their correlation. Therefore, by using both of these measures, as suggested in Section 6.1, we should be able to track the translational discrepancies as well as the change of dependencies between two clusters of data.

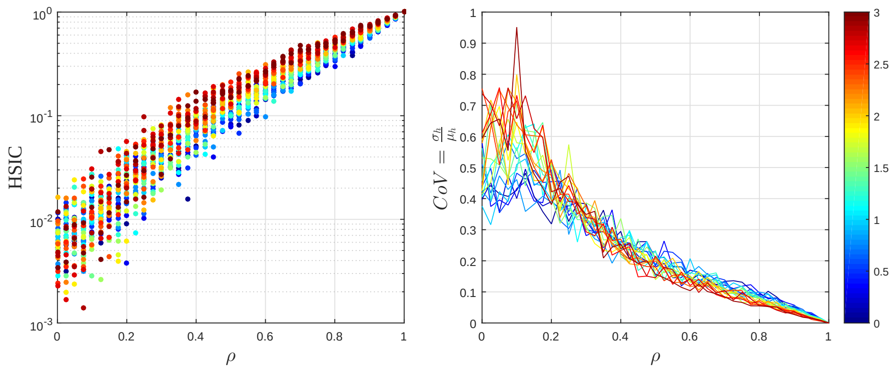
Another important characteristic of these measures is the relation between their magnitude and variance. Figure 12(a) shows the same data as Figure 11(b), but in logarithmic scale. As is shown, the variation of HSIC increases when its magnitude decreases. This can be quantified by the coefficient of variation (CoV) of the HSIC which is shown in Figure 12(b) for different correlation coefficients and shifts of mean values. μ_h and σ_h in this plot are, respectively, the mean value and the standard deviation of the HSIC for a given mean shift and correlation coefficient between $\mathbf{z}_1$ and $\mathbf{z}_2$. Due to the high variations of HSIC for weak dependencies, direct use of the likelihood difference stated in (5c) in our classification problem results in the dominance of the weakly dependent variables in the final decision making. This is in contrast with our main objective; thus, we proposed the voting strategy to avoid the direct use of likelihood ratios in our decision making process.

References

References

[1] M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Processing 99 (2014) 215–249.

[2] F. Angiulli, C. Pizzuti, Support vector method for novelty detection, in: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD '02, 2002, pp. 15–26.



(a) (b)

Figure A.12: The relationship between the magnitude and variance of the HSIC for the Gaussian RVs. (a) variation of HSIC as a function of correlation coefficient and shift of mean values of two Gaussian clusters in log-scale, (b) coefficient of variation for HSIC. The color coded lines and points follow the same rule as in Figure A.11

[3] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, Support vector method for novelty detection, Advances in Neural Information Processing Systems, MIT Press 3 (2000) 582–588.

[4] M. Markou, S.Singh, Novelty detection: a review - part 2: neural network based approaches, Signal Processing, Elsevier 83 (2003) 2499–2521.

[5] P. Paalanen, J.-K. Kamarainen, J. Ilonen, H. Kalviainen, Feature representation and discrimination based on gaussian mixture model probability densities—practices and algorithms, Pattern Recognition, Elsevier 39 (2006) 1346–1358.

[6] C. Desir, S. B. an Caroline Petitjean, H. Laurent, One class random forests, Pattern Recognition, Elsevier 46 (2013) 3490–3506.

[7] Y. Sheikh, M. Shah, Bayesian object detection in dynamic scenes, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE, 2005.

[8] R. Mohammadi-Ghazi, J. Chen, O. Buyukozturk, Pairwise graphical models for structural health monitoring with dense sensor arrays, Journal of Mechanical Systems and Signal Processing 93 (2017) 578–592.

[9] R. Mohammadi-Ghazi, O. Buyukozturk, Non-planar ising graphical model for efficient inference in structural health monitoring, in: 10th International Workshop on Structural Health Monitoring (IWSHM 2015), Stanford, CA, USA, 2015.

[10] R. Mohammadi-Ghazi, Y. M. Marzouk, O. Buyukozturk, Conditional classifiers and boosted conditional gaussian mixture model for novelty detection, Pattern Recognition, Elsevier 81 601–614.

[11] S. Mahamud, Comparing belief propagation and graph cuts for novelty detection, in: Proceedings of

the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), IEEE, 2006.

[12] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer Series in Statistics, 2009.

[13] A. Smola, A. Gretton, L. Song, B. Scholkopf, A hilbert space embedding for distributions, 18th International Conference on Algorithmic Learning Theory, ALT 2007 (2007) 13–31.

[14] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, A. Smola, A kernel method for the two-sample problem, Journal of Machine Learning Research 1 (2008) 1–43.

[15] A. Gretton, K. Fukumizu, Z. Harchaoui, B. K. Sriperumbudur, A fast, consistent kernel two-sample test, Advances in Neural Information Processing Systems 22 (NIPS 2009), Red Hook, NY.

[16] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, A. Smola, A kernel two-sample test, Journal of Machine Learning Research 13 (2012) 723–773.

[17] K. Chwialkowski, A. Gretton, A kernel independence test for random processes, 31th International Conference on Machine Learning, Beijing, China 32.

[18] M. D. Lozzo, A. Marrel, New improvements in the use of dependence measures for sensitivity analysis and screening, Journal of Statistical Computation and Simulation (2014) 1–21`doi:arXiv:1412.1414v1`.

[19] S. D. Veiga, Global sensitivity analysis with dependence measures, Journal of Statistical Computation and Simulation 85 (2014) 1283–1305.

[20] V. J. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review 22 (2004) 85–126.

[21] i. Steinwart, The influence of the kernel on the consistency of support vector machines, Journal of Machine Learning Research 2.

[22] T. Hofmann, B. Scholkopf, A. J. Smola, Kernel methods in machine learning, The Annals of Statistics 36 (2008) 1171–1220.

[23] A. Smola, A. Gretton, L. Song, B. Schölkopf, A hilbert space embedding for distributions, in: Algorithmic Learning Theory: 18th International Conference, 2007.

[24] L. Song, Learning via hilbert space embedding of distributions, Ph.D. thesis, University of Sydney (2008).

[25] A. Shamsheyeva, A. Sowmya, The anisotropic gaussian kernel for svm classification of hrct images of the lung, Intelligent Sensors, Sensor Networks and Information Processing Conference, IEEE.

[26] D. Brodic, Optimization of the anisotropic gaussian kernel for text segmentation and parameter extraction, Theoretical Computer Science (TCS) 323 (2010) 140–152.

[27] F. Aiolli, M. Donini, Learning anisotropic rbf kernels, Artificial Neural Networks and Machine Learning, ICANN.

[28] J. Rice, Mathematical Statistics and Data Analysis, 3rd Edition, Duxbury Press: Belmont, CA, 2007.

[29] R. J. Hyndman, Computing and graphing highest density regions, The American Statistician, Taylor and Francis 50 (1996) 120–126.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[31] 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.

[32] C. R. Farrar, K. Worden, Structural Health Monitoring: A Machine Learning Perspective, John Wiley Sons, 2012.

[33] K. Worden, G. Manson, The application of machine learning to structural health monitoring, Philosophical Transactions of the Royal Society A (2007) 515–537.

[34] K. Worden, G. Manson, The application of machine learning to structural health monitoring, Philosophical Transactions of Royal Society A.

[35] Y. Ying, J. H. G. Jr., I. J. Oppenheim, L. Soibelman, J. B. Harley, J. Shi, Y. Jin, Toward data-driven structural health monitoring: Application of machine learning and signal processing to damage detection, Journal of Computing in Civil Engineering 27.

[36] L. Bornn, C. R. Farrar, G. Park, Damage detection in initially nonlinear systems, International Journal of Engineering Science 48 (2010) 909–920.

[37] S. da Silvaa, M. D. Junior, V. L. Junior, M. J. Brennanc, Structural damage detection by fuzzy clustering, Mechanical Systems and Signal Processing 22 (2008) 1636–1649.

[38] M. H. Rafiei, H. Adelib, A novel unsupervised deep learning model for global and local health condition assessment of structures, Engineering Structures 156 (2018) 598–607.

[39] D. J. Fleet, A. D. Jepson, Computation of component image velocity from local phase information, Int. J. Comput. Vision 5 (1) (1990) 77–104. `doi:10.1007/BF00056772`.
URL `http://dx.doi.org/10.1007/BF00056772`

[40] T. Gautama, M. Van Hulle, A phase-based approach to the estimation of the optical flow field using spatial filtering, Neural Networks, IEEE Transactions on 13 (5) (2002) 1127 – 1136. doi:10.1109/TNN.2002.1031944.

[41] E. P. Simoncelli, W. T. Freeman, The steerable pyramid: A flexible architecture for multi-scale derivative computation, in: icip, IEEE, 1995, p. 3444.

[42] N. Wadhwa, M. Rubinstein, F. Durand, W. T. Freeman, Phase-based video motion processing, ACM Trans. Graph. (Proceedings SIGGRAPH 2013) 32 (4).

[43] 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 (2011) 225–238.

[44] S. Khazai, S. Homayouni, A. Safari, B. Mojaradi, Anomaly detection in hyperspectral images based on an adaptive support vector method, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS 8 (2011) 646–650.