Journal of Sound and Vibration 595 (2025) 118710

Contents lists available at ScienceDirect

Journal of Sound and Vibration

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

Combining transfer learning and numerical modelling to deal with the lack of training data in data-based SHM

Raja Sekhar Battu [a], Konstantinos Agathos [a], Julián Mauricio Londoño Monsalve [a], Keith Worden [b], Evangelos Papatheou [a,*]

[a] Department of Engineering, University of Exeter, North Park Road, Exeter, EX4 4QF, United Kingdom
[b] Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD, United Kingdom

ARTICLE INFO

Dataset link: 10.6084/m9.figshare.26955016

Keywords:
Structural health monitoring
Machine learning
Damage detection
Artificial neural networks
Domain adaptation

ABSTRACT

Structural health monitoring (SHM) involves continuously surveilling the performance of structures to identify progressive damage or deterioration that might evolve over time. Recently, machine learning (ML) algorithms have been successfully employed in various SHM applications, including damage detection. However, supervised ML algorithms often require labelled data for multiple possible damage states of the structure for successful damage identification. Although it may be feasible to gather such data for low-value structures, obtaining damage data for expensive structures such as aircraft could be highly challenging. Herein, this data insufficiency is addressed by combining Finite Element (FE) models with domain adaptation, specifically transfer component analysis (TCA) and joint domain adaptation (JDA). The proposed methodology is showcased in two case studies, a Brake–Reuß beam, where damage scenarios correspond to different torque settings on a lap joint and a wingbox laboratory structure where damage is introduced as saw-cuts. Supervised learning algorithms in the form of Artificial Neural Networks (ANNs) and K-Nearest Neighbours (KNNs) are trained based on FE data after domain adaptation is applied and are then tested with the experimental data. It is shown that even though the performance of classifiers in distinct scenarios of dual, three, four and five-class cases is sensitive to choices in the training stage, the use of TCA or JDA allows for the use of FE data for training and significantly reduces the need for expensive experimental damage data to be used for training. These results can pave the way for a broader use of ML algorithms in SHM of critical and/or expensive structures.

1. Introduction

Monitoring structures for early signs of degradation can help prevent a catastrophic failure. This surveillance can also contribute to structural-integrity preservation in accordance with standard design life. In recent years, a combination of SHM and machine learning algorithms for diagnosis has gained prominence. The two Los Alamos laboratory reports by Doebling et al. and Sohn et al. provided detailed literature surveys on health, condition monitoring techniques, and methodologies [1,2]. In addition, machine-learning and pattern-recognition algorithms have greatly aided SHM in enhancing damage identification [3,4]. A recent review by Malekloo et al. presented a thorough overview of various machine-learning methods and algorithms applied in SHM [5].

If higher levels of damage identification (location, severity), are to be achieved using machine learning, then some form of supervised-learning algorithms may become necessary. However, such algorithms will require labelled datasets representing every

* Corresponding author.
E-mail address: e.papatheou@exeter.ac.uk (E. Papatheou).

https://doi.org/10.1016/j.jsv.2024.118710
Received 7 December 2023; Received in revised form 23 August 2024; Accepted 29 August 2024
Available online 1 September 2024

potential type of damage to the system or structure. Obtaining such data could be highly challenging for expensive structures. Strategies in the form of experimental pseudo-faults (or damage proxies), such as added masses [6,7], have shown before that training supervised-learning algorithms to identify damage without actually damaging a structure is indeed possible. However, they also showed that it is not trivial to know which classifier will work well in the experimental damage data in advance [7]. Numerical models may be used for the training of machine learning (ML) algorithms (see for example [8,9]). In the current study, it is shown that ML algorithms exclusively trained with FE data do not always perform well when tested with experimental data. In order to improve the performance of these algorithms without requiring expensive experimental data, transfer learning methods, specifically transfer component analysis (TCA) and joint domain adaptation (JDA), are therefore employed along with FE models.

Transfer learning is a branch of machine learning aimed at transferring knowledge from one domain to another. Here, one domain is the numerical model and the other the experimental data. Pan and Yang [10] in their survey on transfer learning, provided the association between transfer learning and various related machine-learning approaches, such as domain adaptation, multitask learning, sample selection bias, and covariate shift. Transfer component analysis (TCA) is a domain-adaptation (DA) technique, which learns from a set of common transfer components between distributions [11]. Domain adaptation via TCA was first introduced by Pan et al. [11], proposing a dimensionality-reduction technique which minimises the distance between the domains with different distributions. A transfer learning-based classification approach in SHM was presented by Chakraborty et al. for classifying fatigue damage in an aluminium lug joint [12]. This approach utilised one sensor close to the damage as the *source* domain and the remaining sensors as the *target* domains for transfer learning in order to improve the performance of damage classification. Nevertheless, this approach transfers the knowledge between data obtained from experimental sensors, which may occasionally be difficult to acquire. Furthermore, this necessitates careful sensor selection for both source and target domains in relatively larger structures with numerous sensors.

Domain adaptation in SHM has been applied before by Gardner et al. [13] utilising feature sets of damped natural frequencies and damping ratios. This study demonstrates success in domain adaptation via Transfer Component Analysis (TCA), Joint Domain Adaptation (JDA) and Adaptation Regularization based Transfer Learning (ARTL) for semi-supervised learning. In a case study, numerical-experimental domain adaptation addressed a dual-class problem, achieving equal performance with TCA and JDA. However, further evaluation is required to assess the efficacy of this approach for multi-class problems within a supervised setting. Poole et al. [14] conducted partial domain adaptation, which introduces the importance of alignment between the source and target domains. They also performed effective knowledge transfer utilising statistical alignment (SA), along with normal-condition alignment and normal-correlation alignment. They proposed SA as a remedy for negative transfer issues in class imbalance and partial domain adaptation (DA) as a useful tool for simplifying nonlinear methods. The main focus was on unsupervised learning DA though. Worden et al. [15] explored recent progress in transfer learning within the context of population-based SHM. The authors provided a summary of the theoretical and methodological basis for population-based SHM (PBSHM), focussing on knowledge transfer within, and among, homogeneous and heterogeneous populations of structures. Bull et al. [16] conducted an investigation in which they used PBSHM on a set of similar but slightly varying structures: a population of aircraft tailplanes. They used transfer learning-especially domain adaptation for damage detection in one of the tailplanes by transferring knowledge from one structure to another. This was successfully done in a dual-class (normal or not) scenario, albeit in real structures. PBSHM was also explored in [17], with an interpretable hierarchical Bayesian strategy on operational fleet data. This approach was able to encode and share domain expertise among diverse subgroups based on use-type, component, or operating conditions, demonstrating improvements in truck-fleet survival analysis and wind-farm power prediction.

Domain Adversarial Neural Networks (DANN) have also been used to implement domain transfer. In [18,19] transfer learning was applied with DANN between simulated and experimental data and was also compared with TCA and Joint Domain Adaptation (JDA). In one of the case studies, authors employed DANN for knowledge transfer of numerical-experimental data in a Three-story laboratory structure with 7 damage states, opting for a simplified lumped mass numerical model rather than an FE model. Furthermore, they omitted the comparison of this technique with TCA and JDA for this particular case study. In a similar application [20], DANN was used to extend the knowledge gained from numerical simulation to real-world data when applied in real buildings. This study utilises XGBoost and CNNs for data fusion of multiple features. They assessed the effectiveness of DANN for numerical-experimental knowledge transfer on large-scale shake-table data with four damage states. However, there is a necessity to choose more suitable features to enhance accuracy. In an alternative application of TCA, information was transmitted between varying temperatures on an aluminium beam while detecting damage, effectively counteracting the influences caused by temperature changes [21]. Finally, in [22], a Multi-Channel Subdomain Adaptive Deep Transfer Learning Model (MSADTL) was used for domain adaptation in three different bridge datasets. From the aforementioned applications, only a few [13,18–20] applied a numerical-experimental knowledge transfer using FE models (with one using simplified models), without successfully extending to multiple damage class training data on expensive structures in supervised learning.

In this article, domain adaptation using Transfer Component Analysis (TCA) and Joint Domain Adaptation (JDA) combined with FE data is shown to be effective in addressing the lack of expensive training data in ML. The proposed methodology is demonstrated via two case studies. The first involves a Brake–Reuß beam, addressing dual-class and three-class problems. The second case study focuses on a wingbox structure, addressing a four-class problem. The case studies incorporate the damage scenarios of loosened bolts and saw-cuts, respectively. Training of supervised-learning algorithms exclusively utilised numerical data after the application of TCA or JDA, without incorporating any experimental data. The resulting classifiers were subsequently validated on experimental data to assess the overall performance of the approach.

The remainder of the article is organised as follows: The theoretical representations of both TCA and JDA algorithms are outlined in Sections 2 and 3 respectively. Section 4 describes the proposed methodology while Section 5 examines the application of the

proposed framework on a relatively-simple structure with dual-class and three-class problems and delivers the results in the form of the first case study. Section 6 depicts a relatively complex reinforced panel representing a wingbox structure, with a four-class problem. Conclusions are drawn in the last section.

2. Transfer component analysis

Transfer component analysis was first proposed by Pan et al. [11], who implemented a novel feature extraction while projecting the features to a new latent space. A domain can be defined as an object which consists of feature space and a marginal probability distribution over feature data [10]. Once a domain is specified, the task can be defined, consisting of a label space and an objective predictive function that can be learned from training data (see [10]).

$$\begin{aligned} &Domain && D = \{\mathcal{X}, P(X)\} \\ &Task && T = \{\mathcal{Y}, f(.)\} \end{aligned}$$

where, $\mathcal{X}$ is the input feature space, $P(X)$ is the marginal probability distribution of the feature data, $\mathcal{Y}$ is the label space, and $f(.)$ is the predictive function (in a classification problem, one may adopt $f(.) = P(y|X)$). 'Task' refers to the learning objective or problem to be solved. The knowledge transfer between the source domain (D_s) and the target domain (D_t) starts with the selection of feature sets of respective domains. Let $D_s = \{X_s = [x_1, x_2, \ldots., x_n], \mathcal{P}(X_s)\}$ and $D_t = \{X_t = [X_1, X_2, \ldots .X_n], \mathcal{Q}(X_t)\}$ be the features and probability distributions of the source and target domains. Labelled data in TCA serves a crucial purpose in directing the alignment process and enabling knowledge transfer from the labelled source domain to the unlabelled target domain. The labels of the features in respective domains are denoted as Y_s and Y_t. Domain adaptation is the process of enhancing the target predictive function by utilising the knowledge from the source and target domains [13]. This process aims at minimising the distance between distributions (both domains) in a shared latent space while focussing on the specific task. A significant assumption for most domain adaptation methods is $\mathcal{P} \neq \mathcal{Q}$, but sometimes $P(Y_s|X_s) = P(Y_t|X_t)$. Additional general preliminary assumptions of the approach can be found in [11].

Minimising the Maximum Mean Discrepancy (MMD) between the source and target domains is a technique utilised to accomplish appropriate projection. Furthermore, the source and target domains can be aligned inside a new latent feature space by using a kernel technique to minimise the Maximum Mean Discrepancy (MMD) across domains. MMD is a distance measure between the empirical means of two distributions in a reproducing kernel Hilbert space (RKHS), where the RKHS is principally determined by the kernel function. Let ϕ be the kernel-induced feature map, and $X_s = \{x_{s_i}\}$, $X_t = \{x_{t_i}\}$ be samples from the source and target distributions. The empirical estimate of the distance between two samples is,

$$MMD(X_s, X_t) = ||1/n_1 \sum_{i=1}^{n_1} \phi(x_{s_i}) - 1/n_2 \sum_{i=1}^{n_2} \phi(x_{t_i})||^2_{\mathcal{H}} = tr(\mathbf{GM}) \tag{1}$$

where $||.||_{\mathcal{H}}$ is the RKHS norm, the learned kernel matrix is denoted as $\mathbf{G}$, and the MMD matrix is denoted as M. Thus, from Eq. (1), it is evident that kernel mapping plays a vital role in reducing the distance between $P(\phi(X_s))$ and $P(\phi(X_t))$. The learned kernel matrix $\mathbf{G} = [\phi(x_i)^\top \phi(x_j)]$ is represented with embedded-space Gramm matrices as follows,

$$\mathbf{G} = \begin{bmatrix} G_{s,s} & G_{s,t} \\ G_{t,s} & G_{t,t} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)} \tag{2}$$

The MMD matrix is defined as,

$$\mathbf{M}_{ij} = \begin{cases} 1/n_1^2 & \text{if } x_i, x_j \in X_s \\ 1/n_2^2 & \text{if } x_i, x_j \in X_t \\ -(1/n_1 n_2) & \text{otherwise} \end{cases} \tag{3}$$

The kernel matrix $\mathbf{G}$, as indicated in Eq. (2), serves as the initial step for the parametric kernel mapping, which enables feature extraction based on kernels. The decomposition of this matrix can be represented as an empirical kernel map $(\mathbf{G}\mathbf{G}^{-1/2})(\mathbf{G}^{-1/2}\mathbf{G})$. The empirical kernel map features are transformed to the m-dimensional latent feature space using the matrix $\widetilde{\mathbf{W}} \in \mathbb{R}^{(n_1+n_2)\times m}$ (where $m \ll n_1 + n_2$). As a consequence, the resulting kernel matrix emerges as,

$$\widetilde{\mathbf{G}} = (\mathbf{G}\mathbf{G}^{-1/2}\widetilde{\mathbf{W}})(\widetilde{\mathbf{W}}^\top \mathbf{G}^{-1/2}\mathbf{G}) = \mathbf{G}\mathbf{W}\mathbf{W}^\top\mathbf{G} \tag{4}$$

where, $\mathbf{W} = \mathbf{G}^{-1/2}\widetilde{\mathbf{W}}$. The distance between empirical means of two domain distributions X_S and X_T can be redefined using the definition of the MMD in Eq. (1) and the kernel matrix $\widetilde{\mathbf{G}}$ as,

$$Dist(p(X_S), p(X_T)) = tr((\mathbf{G}\mathbf{W}\mathbf{W}^\top\mathbf{G})\mathbf{M}) = tr(W^\top\mathbf{G}\mathbf{M}\mathbf{G}\mathbf{W}) \tag{5}$$

To minimise the distance metric in Eq. (5), a regularisation term $tr(\mathbf{W}^\top\mathbf{W})$ is typically needed to manage the complexity of $\mathbf{W}$.

When all of the above processes are used in the context of unsupervised TCA, the outcome is a kernel learning problem that can be used as a dimensionality-reduction strategy for domain adaptation. This method is effective in maintaining data variance and decreasing the distance between domain distributions. The kernel problem then becomes,

$$\begin{aligned} &min_{\mathbf{W}} && tr(\mathbf{W}^\top\mathbf{G}\mathbf{M}\mathbf{G}\mathbf{W}) + \mu\ tr(\mathbf{W}^\top\mathbf{W}) \\ &s.t. && \mathbf{W}^\top\mathbf{G}\mathbf{H}\mathbf{G}\mathbf{W} = \mathbf{I}_m \end{aligned} \tag{6}$$

where $\mu > 0$ is a tradeoff parameter, and $\mathbf{I}_m \in \mathbb{R}^{(m\times m)}$ is the identity matrix. $\mathbf{I}_m$ is further abbreviated as $\mathbf{I}$ for simplicity. This problem can be solved by a trace optimisation problem such as,

$$max_{\mathbf{W}} \quad tr((\mathbf{W}^{\top}(\mathbf{GMG} + \mu\mathbf{I})\mathbf{W})^{-1}\mathbf{W}^{\top}\mathbf{GHGW}) \tag{7}$$

The solution to the optimisation problem in Eq. (7) provides a transformation matrix $\mathbf{W}$, which is composed of m leading vectors chosen from eigenvalue decomposition of the matrix $(\mathbf{GMG}\mu\mathbf{I})^{-1}\mathbf{GHG}$, where $m \leq n_1 + n_2 - 1$ [11].

3. Joint domain adaptation

Joint Domain Adaptation (JDA) is an alternative domain adaptation technique similar to TCA. However, JDA jointly utilises both the marginal and conditional distributions to reduce the discrimination between the source and target domains [23]. JDA works under suppositions such as $P(X_s) \neq P(X_t)$ and $P(Y_s|X_s) \neq P(Y_t|X_t)$, in other words, the joint distributions are dissimilar. The goal of the JDA is to minimise the discrepancy between joint distributions to acquire the non-linear transform of the feature space to an RKHS ($\phi : \mathcal{X} \mapsto \mathcal{H}$).

The challenge in achieving joint distribution minimisation comes from conditional distribution. Direct modelling is only possible if there exist labels in the target domain. Since there are no labels in the target data direct modelling is highly impossible. One approach to tackle this hurdle is by utilising base classifiers (*e.g.* Support Vector Machine (SVM), K-nearest neighbour (KNN)) trained on labelled source data, and applying them to unlabelled target data. Additionally, posterior probabilities are intricate to obtain, and this method explores class-conditional distributions $P(Xs|Ys), P(Xt|Yt)$ instead. The estimates of the target labels (pseudo labels) from a base classifier and true source labels are then utilised to match the class-conditional distributions $P(Xs|Ys = c)$ and $P(Xt|Yt = c)$ where c $\in \{1, \ldots, C\}$ in the label set $\mathcal{Y}$ [23]. In this method, the distance of these class-conditional distributions (kernel embedded) is calculated by MMD as,

$$Dist(p(X_S), p(X_T)) + Dist(P(Y_s|X_s), P(Y_t|X_t)) \approx tr(\mathbf{W}^{\top}\mathbf{G_k}\mathbf{M_c}\mathbf{G_k}\mathbf{W}) \tag{8}$$

where, $\mathbf{G_k}$ denotes kernel matrix for joint distributions, $\mathbf{M_c}$ is the MMD matrix and notably $c = 0$ produces the MMD in TCA. Therefore, the MMD matrix while minimising becomes,

$$(\mathbf{M_c})_{ij} = \begin{cases} \frac{1}{n_s^{(c)} n_s^{(c)}} & \text{if } x_i, x_j \in D_s^{(c)} \\ \frac{1}{n_t^{(c)} n_t^{(c)}} & \text{if } x_i, x_j \in D_t^{(c)} \\ \frac{-1}{n_s^{(c)} n_t^{(c)}} & \begin{cases} x_i \in D_s^{(c)}, x_j \in D_t^{(c)} \\ x_j \in D_s^{(c)}, x_i \in D_t^{(c)} \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where, $D_s^{(c)} = \{x_i : x_i \in D_s \wedge y(x_i) = c$ source data, $y(x_i)$ is true label of x_i and $n_s^{(c)} = |D_s^{(c)}|$. Correspondingly, $D_t^{(c)} = \{x_j : x_j \in D_t \wedge \hat{y}(x_j) = c\}$ are the examples from class c in target data. $\hat{y}(x_j)$ is the predicted pseudo labels of x_j, and $n_t^{(c)} = |D_t^{(c)}|$.

Similar to TCA, the eigenvalue problem where the optimal W is obtained from the eigenvectors corresponding to the smallest eigenvalues is from,

$$(\mathbf{G_k}\sum_{c=o}^{C}\mathbf{V_c}\mathbf{G_k} + \mu\mathbf{I})\mathbf{W} = \mathbf{G_k}\mathbf{HG_k}\mathbf{W}\phi \tag{10}$$

The transformed feature space can be calculated by $\mathbf{Z} = \mathbf{G_k}\mathbf{W} \in \mathbb{R}^{(N_s+N_t)\times n}$.

4. Proposed methodology

Fig. 1 depicts the flow diagram of the proposed methodology, illustrating the incorporation of numerical models and domain-adaptation in SHM. A numerical model is necessary to provide the surrogate data for training damage classifiers. For this purpose, finite-element models are created utilising the equivalent CAD models of the Brake–Reuß beam and Wingbox structure. Subsequently, These FE models are updated iteratively, employing a sensitivity-based model-updating technique based on vibration test data, as detailed in [24]. The approach then follows a conventional machine-learning implementation using features (entirely selected on FE data), followed by the application of TCA or JDA.

Initially, as this research focusses on frequency-domain features, frequency response functions (FRFs) from both FE models (experimentally validated) and experimental measurements are extracted. These FRFs and transmissibility ratios are then employed for manual feature selection. As a consequence, features from the FE and experimental domains serve as input to the domain-adaptation algorithm. Only experimental data from the 'normal' state were used at this stage. Finally, the output features from domain adaption (TCA or JDA) are incorporated to train multiple classifiers for damage detection.

Feature selection or extraction arguably plays a crucial role in the performance of the trained classifiers. Here, a feature is considered to be any set of values derived or calculated from experimental or simulated data which can help discriminate among different classes. Similar to [6,7] a manual feature-selection technique is employed here by visualising the numerically generated (FE) FRFs or transmissibilities to identify the most relevant features in relation to each damage scenario. The overall objective was

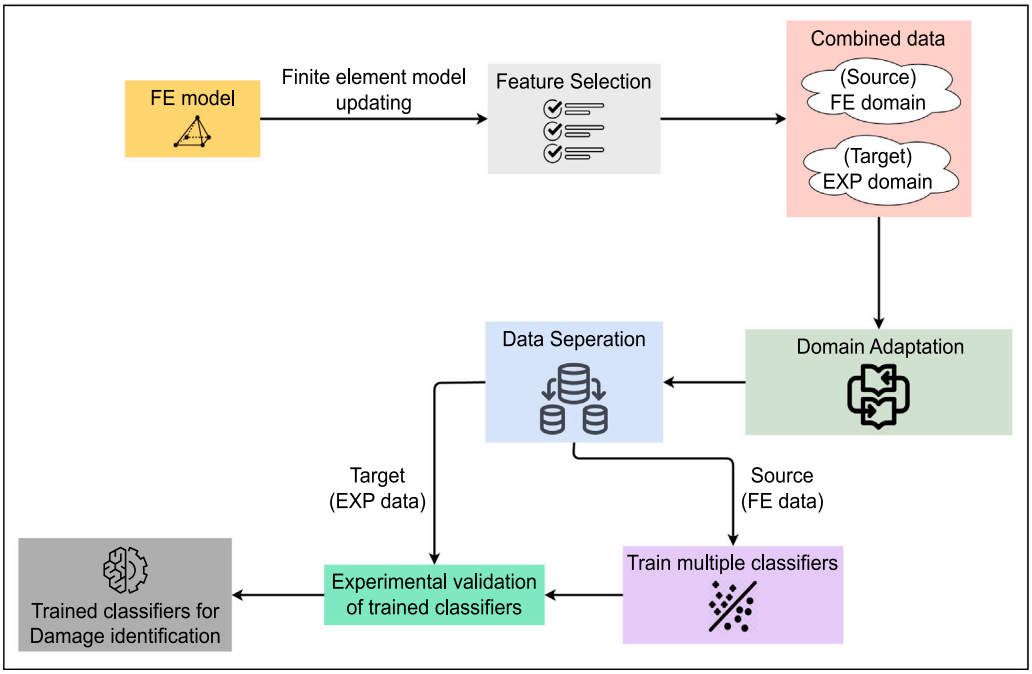


Fig. 1. Flow diagram of the proposed methodology.

the selection of at least one discriminating feature per damage class. Once the features are chosen, they were copied 500 times and a Gaussian noise with a signal-to-noise ratio of 5 was added to create a feature dataset ready for further analysis and classification.

The 'normal' state data from the source and target domains are fed into the TCA algorithm, yielding both the data and transformation matrix in new feature space. The obtained transformation matrix (**W**) (seen in Eq. (7)) can be in turn used to project the samples into the new latent feature space for domain adaptation. Using the output features from this transformation, classifiers can train on source data and can be validated on target data.

The benefit of domain adaptation in SHM is that it can be implemented if the target domain has a small amount of labelled data. The kernel functions such as *linear*, *radial-basis function* (RBF), and *polynomial* kernels can be utilised for domain adaptation [11]. In this article, a linear mapping function or '*linear*' kernel was used in TCA, while a *non-linear* (RBF) kernel was employed in JDA. The further step is to select the width of the kernel which is also depicted as kernel sigma (σ), and the dimension of the latent subspace after projection. In SHM, the choice of the kernel function, and kernel sigma, could change depending on the linearity of the selected features, and the nature of the specific problem being addressed within systems, or structures.

The current work employs an approach for producing damage surrogates that are based on a transformation matrix created from the domain alignment of normal (healthy) data from the source and target domains. The projection begins with a consolidated feature dataset (potential damage) that combines datasets from the source (FE) and target (EXP) domains, which are then processed via a kernel function. The creation of projected datasets in the newly-formed latent feature space involves conducting matrix multiplications between the obtained kernel dataset (potential damage) and the transformation matrix (healthy). This approach is consistently applied to every selected feature within each class to produce datasets for all of the given 'damage' classes. After computing the feature sets for healthy and damaged classes, the Mahalanobis distance between normal (healthy) and abnormal classes can then be determined and used as a measure of 'abnormality' in the dataset [25]. The distance measure utilised for performing a discordancy test on this multivariate data is given by,

$$D_\zeta = (\{\mathbf{X}_\zeta\} - \{\bar{\mathbf{X}}\})^\top [\mathbf{S}]^{-1} (\{\mathbf{X}_\zeta\} - \{\bar{\mathbf{X}}\}) \tag{11}$$

where $\{\mathbf{X}_\zeta\}$, $\{\bar{\mathbf{X}}\}$ and $[\mathbf{S}]$ are outlier datum, mean vector of the sample space and sample covariance matrix respectively. It is the Mahalanobis distance on the selected features that is used as input to the classifiers.

4.1. Classifiers used: ANNs and KNN

The third and final phase of this methodology is to train multiple classifiers. Two supervised learning classifiers are used here for comparison: Artificial Neural Networks (ANNs) in the form of Multi-Layer Perceptrons (MLPs) and K-nearest neighbour (KNN). For the purposes of this work, 40 Multi-Layer Perceptron structures are employed, consisting of one hidden layer and varying hidden units, ranging from 5 to 200 in steps of 5. Additionally, every network structure contains 20 random weight initialisations. In total, there are 800 MLPs that are implemented to train exclusively on the finite-element features. Testing on experimental features was

Table 1
KNN classifier variants with 'K' values optimised for different distance metrics.

'K'	Distance metric	Formula
Bayesian optimisation	Euclidian	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Bayesian optimisation	City block	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Bayesian optimisation	Chebyshev	$\max(\lvert x_i - y_i \rvert)$
Bayesian optimisation	Minkowski (cubic)	$\left(\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p\right)^{\frac{1}{p}}$
Bayesian optimisation	Cosine	$\frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}$

Table 2
Order of experimental tests and bolt configurations of the beam.

Test	Bolt configuration
Normal	All bolts Torque = 20 Nm
Damage class 1	All bolts Torque = 10 Nm
Damage class 2	All bolts Torque = 5 Nm

carried out to validate the efficiency and performance of MLPs. The ability to determine the overall efficiency of SHM depends on the total number of effective MLPs on the experimental data.

For the second classifier, the K-nearest neighbours algorithm (KNN) is used; the 'K' value determines the number of neighbours needed to be considered for the training dataset. This algorithm is generally considered simple to implement; however, its major limitation is its high computational requirement for larger datasets. Over the span of this article, KNN classifiers with five distance metrics are employed as can be seen in Table 1. While Euclidean distance is the most common metric, the rest of the distance metrics are chosen here mainly for the sake of comparison.

The 'K' value for a classifier with selected distance metric is tuned to achieve the optimal performance of the model, evaluated on a validation set (15% of training data). This optimisation procedure is enabled by employing Bayesian optimisation (Built-in optimiser in MATLAB), involving a range of 'K' values between 1 and half of the total number of observations ($n/2$). The programming and training of neural networks and KNN classifiers were executed using MATLAB.

5. Case study I: Brake-Reuß beam

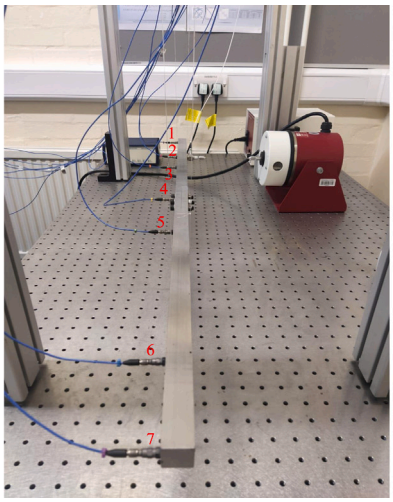
The first chosen structure for the experimental demonstration was a Brake–Reuß beam [26], illustrated in Fig. 2. It consists of two beam parts joined by a central lap joint and constitutes a prismatic beam with a length of 108 cm and a squared cross-section of side 2.54 cm. The joint is secured with three M8 MS bolts, nuts, and washers at a torque of 20 Nm. The dashed lines in Fig. 2(b) indicate the locations of all three bolts on the structure. In terms of damage detection, a bolt pretension of 20 Nm for all three bolts is considered as a baseline or healthy class (normal). By loosening these three bolts, it is feasible to create various damage scenarios.

5.1. Test procedure

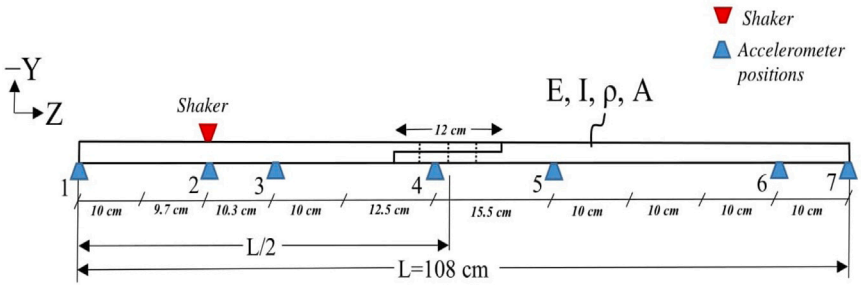
Fig. 2(a) depicts the test apparatus of the experimental procedure. Free-free boundary conditions are approximated by suspending the beam on fishing wires at two locations. Seven accelerometers were located axially to the z-axis, which can be seen numbered in both the experimental setup and the schematic diagrams in Fig. 2. Two National Instruments (NI) 9234 modules with a total of eight channels were utilised to acquire the experimental data. A sampling rate of 25.6 kHz was used for all the measurements. The data-acquisition system employed during the test was a NI DAQ controlled by Labview (Signalexpress) software running on a Dell PC. The sensors used were piezoelectric accelerometers of the PCB type. The excitation force had a duration of 25 s, was generated using a random noise generator and recorded via a force transducer. Frequency response functions (FRFs) were estimated using the H1 estimator with a frequency resolution of 0.3906 Hz and a bandwidth of 0–12800 Hz. Furthermore, the SDTools Matlab package [27] was utilised to carry out experimental modal analysis and extract the natural frequencies and mode shapes of the first nine modes. Table 2 details the order of experimental tests and bolt configurations used for the respective damage classes.

5.2. FE model: Brake-Reuß beam

Fig. 3 shows the FE model used for the Brake–Reuß beam. The model was created using an in-house MATLAB finite element code, while Gmsh was used for meshing and ParaView for post processing. The beam was meshed with linear tetrahedral solid elements with a mesh size of h = 4 mm resulting in a total of 113,580 nodal degrees of freedom. For this mesh density all eigenvalues considered are converged. The material properties included in this numerical model are illustrated in Table 3 below.



(a)



(b)

Fig. 2. Experimental setup and schematic diagram of the Brake–Reuß beam. (a) The setup includes a shaker and seven piezoelectric accelerometers of PCB type. (b) A schematic diagram showing the positions of the shaker and accelerometers.

Table 3
Material properties of Brake–Reuß beam (Stainless Steel 304).

Property	Value
Young's Modulus (E)	193 GPa
Poisson's Ratio (ν)	0.29
Density (ρ)	8000 kg/m^3

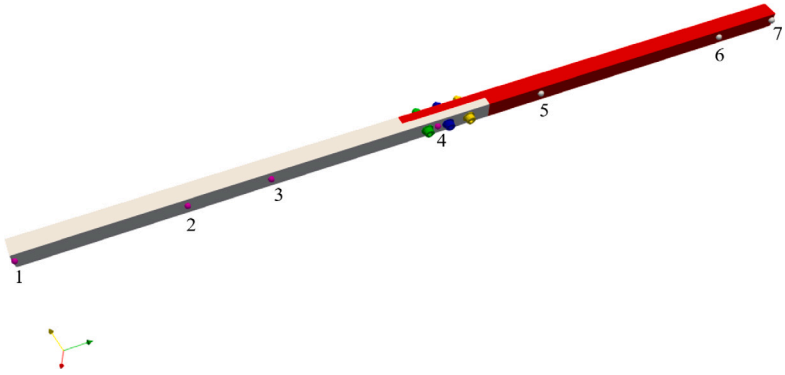


Fig. 3. FE model of the Brake–Reuß beam.

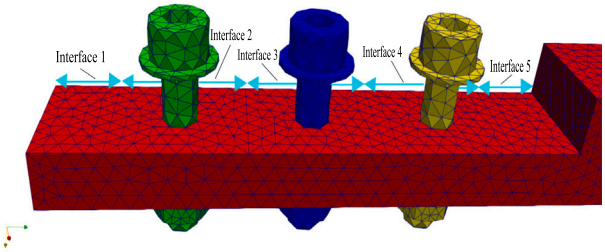


Fig. 4. Spring interfaces between the faces to build the joint of FE model of Brake–Reuß beam.

Table 4
Natural frequencies of the first five bending modes.

Mode	Updated numerical (Hz)	Experimental (Hz)	Abs. Difference (%)
1	81.27	80.99	0.34
2	284.44	281.28	1.12
3	504.35	508.27	0.77
4	839.97	831.92	0.96
5	1288.24	1307.11	1.44

Table 5
Brake–Reuß beam: stiffness reduction parameters for damage modelling in FE model.

Damage class	Percentage	Reduction value
All bolts torque 10 Nm	50%	0.5
All bolts torque 5 Nm	25%	0.25

5.2.1. Joint modelling

One of the approaches to simulate joints is by representing bolt pretensions as directional springs in the FE model. In the present case, a similar strategy has been applied by adding spring interfaces next to the bolts on the joint, identical to [26]. These interfaces serve as the contacts to the FE-model lap-joint. Spring interfaces and bolts are also connected with mesh connections (see Fig. 4). In order to simulate the damage classes listed in Table 2, one can alter the spring stiffness of interfaces and the material stiffness of the bolts. The final combination of parameters required to be altered is eight, including five interface stiffnesses and three material stiffnesses of the bolts.

The FE model was updated using the sensitivity technique proposed by Mottershead et al. [24] by comparing the experimentally-acquired first five bending modes. Furthermore, the parameters selected to update were the material stiffness of all FE model parts as well as the stiffness parameters of the spring interfaces. For more details see [28]. Proportional damping was added to the updated model by computing a damping matrix with the support of experimental natural frequencies and appropriate damping ratios of the third and fourth bending modes [29]. Following the model updating, the material stiffness of the main parts of the beam was adjusted to 183.29 GPa. The three bolts of the joint had stiffness values of 332.9 GPa, 97.5 GPa and 318 GPa respectively, while the five spring interfaces varied between 99.6 MPa and 99.9 MPa. The updated model underwent experimental validation, utilising natural frequencies and modeshapes (Modal Assurance Criterion). Table 4 summarises the comparison of updated numerical and experimental natural frequencies. The visualisation of updated numerical and experimental FRFs can be seen in Fig. 5.

In this case study, FRFs from the Brake–Reuß beam FE model are extracted for application of the proposed framework. To simulate damage in the numerical model, the torque setting for the experimental damage cases presented in Table 2 is adopted as a reference. As the 10 Nm and 5 Nm bolt pretension corresponds to 50% and 25% of the nominal torque setting, damage was introduced to the FE model by reducing eight parameters which control the joint stiffness to 50% and 25% of their initial values (see Table 5). The following step was feature selection on the FE data which correspond to the seven sensor locations. As in [7], a visual inspection of FRFs was performed, and one feature per sensor from Sensors 2,3,4 and 7 was selected. Fig. 6 displays the four features (one peak, three troughs) for all three selected damage classes (Normal, 10 Nm, 5 Nm).

5.3. TCA procedure

The procedure for domain adaptation starts by creating the data for the source and target domains. In this context, the FE domain is configured as the source domain, while the Experimental domain is the target domain. The healthy feature datasets (see Section 4) of the source and target domains (500×45) are merged to create a combined dataset of 1000×45. The input features and hyperparameters in this machine-learning algorithm play a vital role in domain adaptation. In addition, the choice of TCA input features or variables is influenced by a number of factors, such as the source and target domains, the type of problem, and the experimental structure used. For this problem, the TCA (Transfer Component Analysis) hyperparameters have been selected

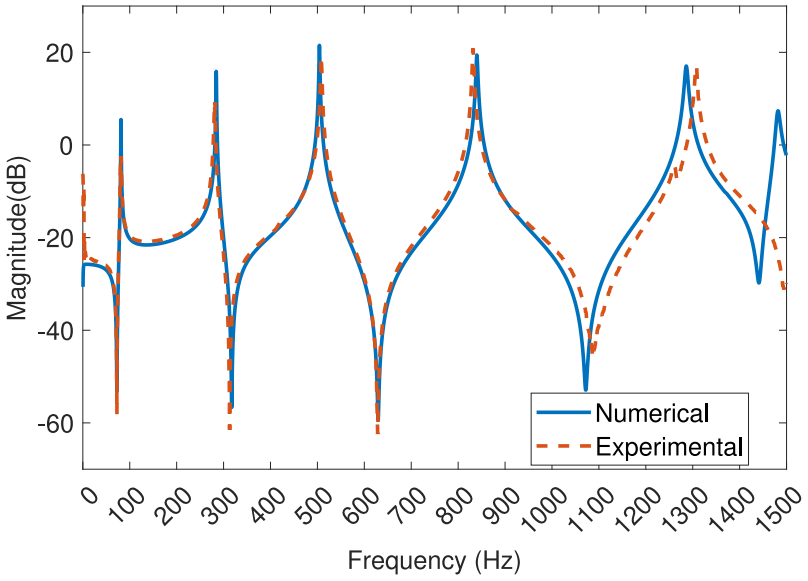


Fig. 5. Comparison between updated numerical and experimental FRFs of Brake–Reuß beam.

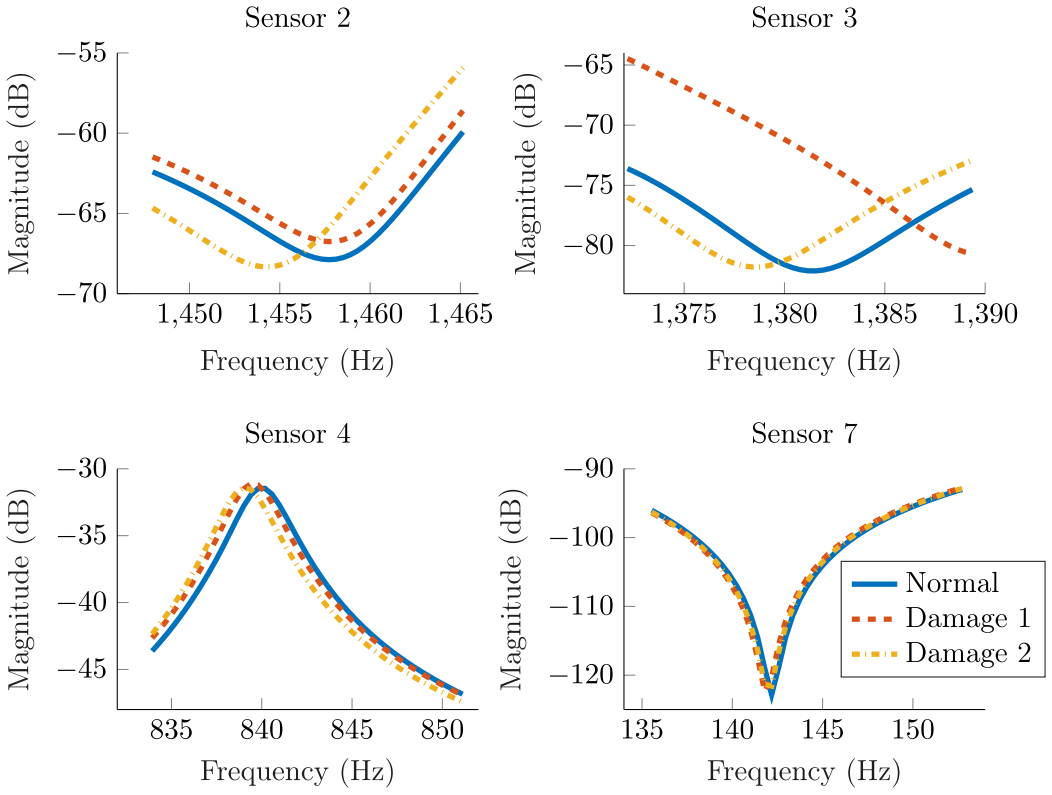


Fig. 6. Frequency range for the manually-selected FE features of the beam from sensors 2,3,4 and 7: one peak and three troughs of FRFs.

from [11], using the input data (combined domain data) with a linear kernel having a sigma of 10, regularisation parameter (μ) is 1, and a latent subspace dimension of 25. The subspace dimension is maintained lower than the original feature space on purpose to achieve dimensionality reduction. The TCA algorithm takes the combined dataset of dimensions 1000 × 45 along with the specified parameters as input. Via the TCA algorithm, an output of a new feature set is generated, with dimensions 1000 × 25 considering the

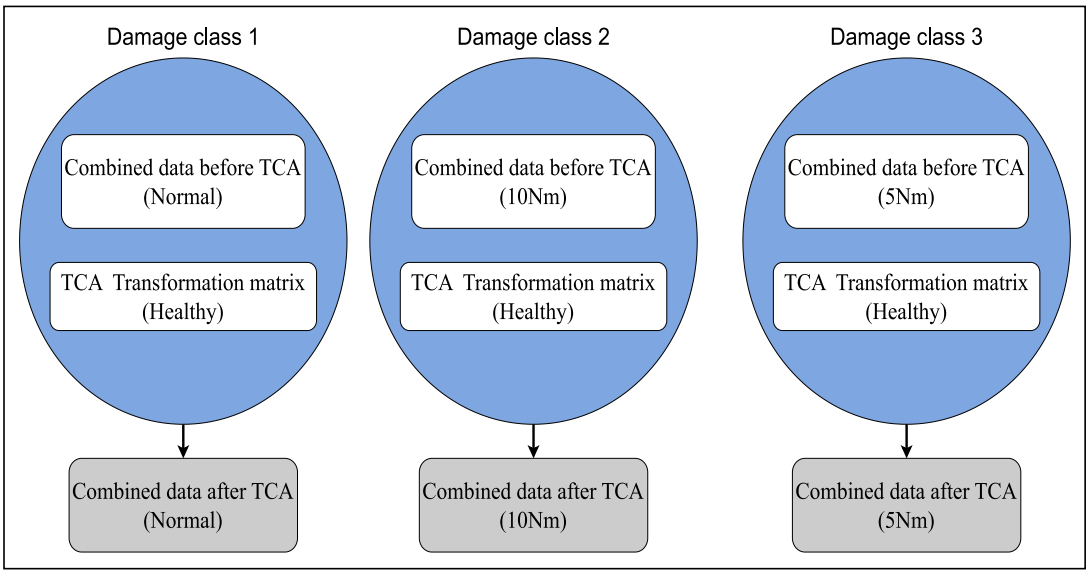


Fig. 7. Procedure to apply TCA (domain adaptation) with baseline (healthy) transformation matrix on a three-class problem. This procedure was frequently used throughout this article for dual, and four-class problems.

provided latent feature space dimension $m = 25$. Additionally, the output includes a transformation matrix of dimensions 1000×25 for the chosen feature, which is also a combined matrix of both the source and target domains.

Fig. 7 illustrates the transformation of the damage classes (feature set) into the latent feature space with a block diagram for a three-class problem. For each chosen feature, the combined feature sets related to damage undergo matrix multiplication using the transformation matrix obtained from the previous step, specifically derived from the normal (healthy) class. The resulting output feature set (1000×25) is then divided into FE and experimental domain features. In the final phase, the feature sets of the normal and damage classes are utilised to compute the Mahalanobis distance using Eq. (11) between the baseline data and the damage classes. The same procedure is repeated for all chosen feature datasets to acquire the Mahalanobis distance feature set of dimensions 4×1500 (input data (Mahalnobis distances) $\times$ (Normal, 10 Nm, 5 Nm)) for a three-class problem. Additionally, the above-stated procedure (quantifying distances) is applied to both FE and experimental features, both before and after the implementation of TCA. Training of classifiers is carried out using FE data, while testing is conducted on experimental features. These final data (input data for classifiers based on distance measures (training and testing)) are compared before and after TCA to visualise the clusters and outliers of both datasets.

5.4. JDA procedure

In order to compare the results of this SHM application with another domain adaptation algorithm beyond TCA, JDA is employed. Typically, conventional JDA trains a classifier on the source data and utilises pseudo labels from the target data iteratively to optimise output feature sets. However, the proposed methodology in this article depends solely on the normal (source and target) data for generating a transformation matrix which is in turn used to project data in new latent feature space. The sole deviation in parameters compared to TCA is the kernel type. In line with a domain adaptation application [13,30], the kernels implemented can be either '*linear*' or '*radial basis function*'. However, in this study, the '*linear*' kernel underperforms in comparison to the '*radial basis function*' in the JDA application. Therefore, the '*radial basis function*' is chosen as the kernel type which is a non-linear kernel. According to Long et al. [23] the recommended range of optimal regularisation parameter for JDA lies within $\lambda \in [0.001, 1]$. Thus, the hyperparameter (regularisation parameter = 1) adopted from TCA was reiterated while using JDA, which yielded favourable outcomes.

5.5. Dual-class problem

As a simple problem to start with, the dual-class problem with classes normal and all three bolts with pretension torque 10 Nm (see Table 2) was chosen. Domain adaptation was carried out for both classes using the corresponding feature sets with the transformation matrix from the baseline (healthy) data.

For the 800 neural networks used, the input data set of the FE features was divided into a 70% training, 15% validation, and a 15% testing set. This FE testing set (15%) aids in acquiring the initial performance of neural networks in FE data. The test dataset from experimental features provides the overall effectiveness of the approach. Any neural network (NN) that has a classification efficiency of more than 90% is typically regarded as an excellent network. There were 677 out of 800 neural networks that performed

Table 6
Dual-class problem: Neural networks with performance > 90% on experimental data of Brake–Reuß beam before domain adaptation (DA), after application of TCA, and after JDA.

Total neural networks	Performance of neural networks		
	Before DA	After TCA	After JDA
800	677	800	796

Table 7
Dual-class problem of Brake–Reuß beam: training and testing results of KNN classifiers (with optimised 'K' values) before Domain Adaptation. Training accuracy is assessed using the validation set (15%) derived from FE data, whereas testing accuracy is measured using purely experimental data.

K	Distance metric	Training accuracy FE Data	Testing accuracy Experimental data
12	Euclidian	100%	99.3%
50	City block	100%	100%
1	Chebyshev	100%	94.8%
8	Minkowski (Cubic)	100%	97.1%
9	Cosine	100%	99.3%

Table 8
Dual-class problem of Brake–Reuß beam: training and testing results of KNN classifiers (with optimised 'K' values) after TCA, JDA. Training accuracy is assessed using the validation set (15%) derived from FE data, whereas testing accuracy is measured using purely experimental data.

K	Distance metric	Training accuracy FE Data	Testing accuracy Experimental data (TCA)	Testing accuracy Experimental data (JDA)
12	Euclidian	100%	100%	99.5%
50	City block	100%	100%	100%
1	Chebyshev	100%	100%	98.6%
8	Minkowski (Cubic)	100%	100%	99.1%
9	Cosine	100%	100%	99.5%

well on the experimental data prior to the application of TCA. The application of TCA improved the performance of NNs, as all 800 networks performed excellently on experimental data. JDA showed equivalent results. The comparison of the neural networks performing the best classification on experimental features before and after the application of domain adaptation is provided in Table 6.

At the same time, KNN classifiers with the different distance metrics listed in Table 1 were used for training and testing in order to check and compare damage detection against various classifiers. FE features were partitioned into two subsets: an 85% training set and a 15% validation set, both of which were supplied as input to the classifier. During the training process, Bayesian optimisation was used to fine-tune the 'K' value of each classifier as outlined in Section 4.1. A testing set of experimental data was provided to all trained KNN classifiers. Following the application of TCA, every classifier performed flawlessly on the experimental data. Tables 7 and 8 show how various KNN classifiers performed before and after the application of TCA and the comparison with JDA.

5.6. Three-class problem

In order to demonstrate a more complex problem, all three damage classes in Table 2 are chosen for analysis in the present sub-section. In a manner similar to that described in Section 5.3, the damage features are obtained in the new latent space using a transformation matrix from the healthy or normal case of the features. Once again, but now for three classes, the Mahalanobis distance features are computed. The input features to the classifier are of dimensions 4×1500, with the first 500 representing the normal class, the next 500 representing damage class 1 (10 Nm), and the final 500 representing damage class 2 (5 Nm). In order to visualise the effect of TCA on the feature-data used as an input to the classifiers, a 3-D scatter plot is used to illustrate their first three principal components (using PCA). Figs. 8 and 9 and Fig. 10 show that TCA effectively aligns the corresponding classes (normal FE with normal experimental, 10 Nm FE with 10 Nm experimental etc.). Notably, before the application of TCA the 10 Nm FE data are closer to the 5 Nm experimental data (see Fig. 8).

The same data division employed for the dual-class problem was carried out for both the training and testing datasets here. In trained neural networks, only 118 of the 800 neural networks were performing well on the experimental data prior to the application of TCA. Following the application, 764 of 800 networks on experimental data behaved flawlessly. As a result, the fraction of networks that perform well on experimental data rose from 14.75% to 95.5%.

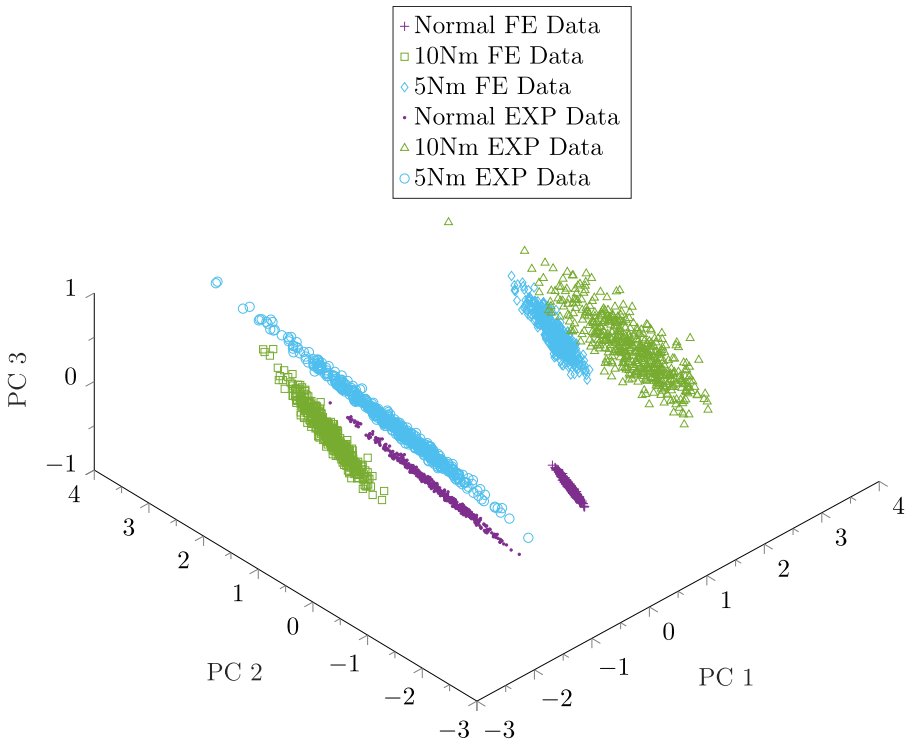


Fig. 8. Brake–Reuß beam: comparison of first three principal components (using PCA) of input data of the three-class problem before application of TCA. Notably, the '5 Nm FE Data' and '10 Nm Experimental Data' exhibit closer proximity. Colour denotes the same class (Normal, 10 Nm or 5 Nm).

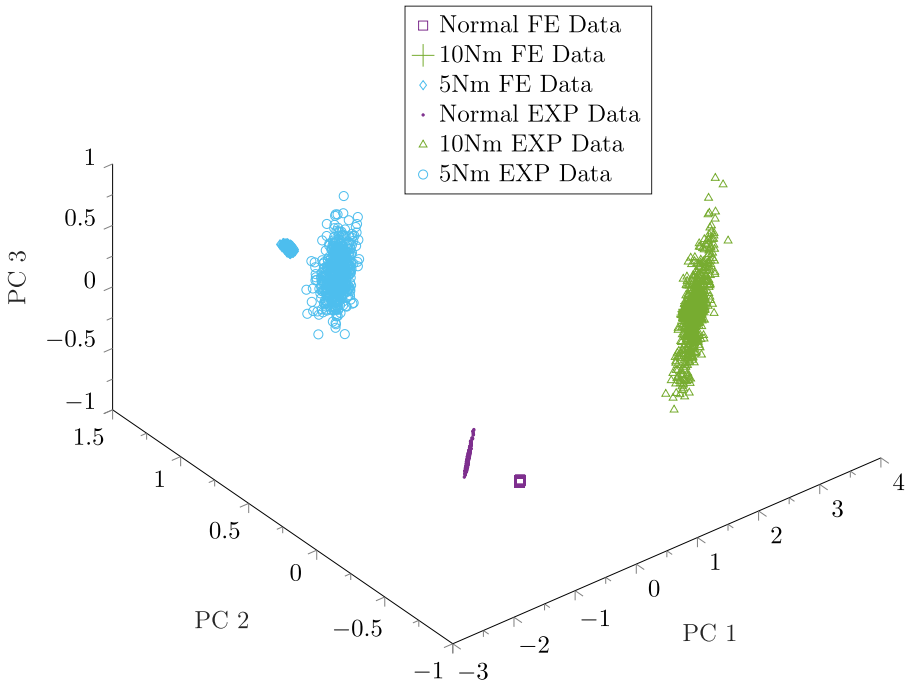


Fig. 9. Brake–Reuß beam: comparison of first three principal components (using PCA) of input data of the three-class problem after application of TCA. Implementing TCA highly reduced the separation between FE and EXP classes, positioning '10 Nm FE data' behind '10 Nm EXP data'. Colour denotes the same class (Normal, 10 Nm or 5 Nm).

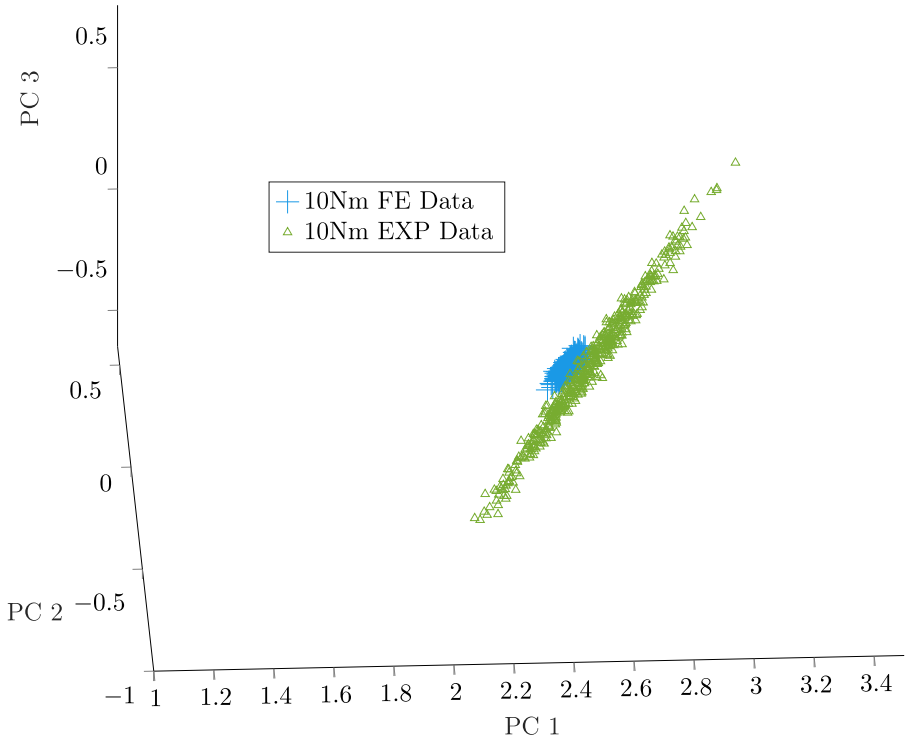


Fig. 10. Brake–Reuß beam: zoomed-in view of the first three principal components, showcasing both 10 Nm finite element (FE) and experimental (EXP) data from Fig. 9.

Table 9
Three-class problem: Neural networks with performance > 90% on experimental data of Brake–Reuß beam before domain adaptation (DA), after application of TCA, and after JDA.

Total neural networks	Performance of neural networks		
	Before DA	After TCA	After JDA
800	118	764	740

Table 10
Three-class problem of Brake–Reuß beam: training and testing results of KNN classifiers (with optimised 'K' values) before Domain Adaptation. Training accuracy is assessed using the validation set (15%) derived from FE data, whereas testing accuracy is measured using purely experimental data.

K	Distance metric	Training accuracy FE Data	Testing accuracy Experimental data
19	Euclidian	100%	88.8%
16	City block	100%	92%
2	Chebyshev	100%	89.2%
38	Minkowski (Cubic)	100%	87.4%
1	Cosine	100%	89.5%

Table 9 illustrates the comparison of neural networks exhibiting high classification accuracy on experimental features before and after the application of TCA and JDA. The hyperparameters chosen for the TCA, as outlined in [11], kernel sigma as 10 and regularisation parameter (μ) set to 1. Subsequent investigation of the TCA parameters by varying the kernel sigma between 10, 20 and 30 and μ by 1 to 10 did not show any effect on the overall number of well-performing neural networks.

In addition, KNN classifiers responded in a similar manner to neural networks when dealing with the three-class problem. Once again, the 'K' value for each selected KNN classifier (see Table 1) was refined by Bayesian optimisation. Tables 10 and 11 provide a demonstration of the performance of KNN classifiers before and after the implementation of TCA.

Table 11
Three-class problem of Brake–Reuß beam: training and testing results of KNN classifiers after TCA, JDA. Training accuracy is assessed using the validation set (15%) derived from FE data, whereas testing accuracy is measured using purely experimental data.

K	Distance metric	Training accuracy FE Data	Testing accuracy Experimental data (TCA)	Testing accuracy Experimental data (JDA)
19	Euclidian	100%	100%	98.7%
16	City block	100%	100%	98.8%
2	Chebyshev	100%	100%	98.5%
38	Minkowski (Cubic)	100%	100%	98.5%
1	Cosine	100%	100%	97.3%

Table 12
Order of experimental tests and bolt configurations of the beam with reduction in torque values.

Test	Bolt configuration	Reduction		
		B1	B2	B3
Normal	All bolts Torque (B1, B2, B3) = 20 Nm	1	1	1
Damage class 1	All bolts Torque (B1, B2, B3) = 10 Nm	0.5	0.5	0.5
Damage class 2	All bolts Torque (B1, B2, B3) = 5 Nm	0.25	0.25	0.25
Damage class 3	Bolt B1 = 5 Nm, Bolts (B2, B3) = 20 Nm	0.25	1	1
Damage class 4	Bolt B2 = 20 Nm, Bolts (B1, B3) = 10 Nm	0.5	1	0.5

Table 13
Five-class problem: Neural networks with performance > 90% on experimental data of Brake–Reuß beam before domain adaptation (DA), after application of TCA, and after JDA.

Total neural networks	Performance of neural networks		
	Before DA	After TCA	After JDA
800	0	755	730

5.7. Five-class problem

This section explores the methodology's application to additional classes with varied bolt configurations. Building upon the three existing classes presented in Table 9, two additional classes are introduced, resulting in a total of five classes. In the fourth class, the torque of bolt B1 is decreased to 5 Nm, while bolts B2 and B3 remain at the standard normal condition torque of 20 Nm. In the fifth class, the torques of bolts B1 and B3 are both reduced to 10 Nm, with bolt B2 maintained at the normal condition torque of 20 Nm. Detailed configurations and the corresponding torque reduction values are outlined in Table 12.

The established procedure for feature selection, parameters for domain adaptation, and training classifiers is adhered to consistently, as outlined in the previous section for the three-class problem. Similarly, the classifiers trained on FE data from domain adaptation (DA) include 800 neural networks with varying hidden units and random initialisations, as well as five KNN classifiers employing different distance metrics. All trained classifiers are stored and subsequently tested with experimental data. The results demonstrate consistent performance, despite the increase in the number of classes and the variation in bolt configurations. The performance of the classifiers on experimental data after domain adaptation is presented in Tables 13 and 14.

6. Case study II: Wingbox structure (Four-class problem)

6.1. Experimental setup

In order to increase the complexity of the problem for the application of the aforementioned methodology, the second structure chosen was a wingbox laboratory structure. The experimental data from [6] were employed to further conduct an experimental modal analysis. The test setup used in [6] was a stiffened aluminium panel intended to replicate an aircraft wingbox. The top sheet of the Wingbox is a 750 × 500 × 3 mm aluminium sheet. This sheet is reinforced by two C-channel parts riveted to the shorter edges and two stiffening stringers made of an angle section attached down the length of the sheet, as shown in Fig. 11. Experimentally, free-free boundary conditions were approximated by suspending the wingbox from a large frame with springs and nylon lines attached at the corners of the top sheet.

Table 14
Five-class problem of Brake–Reuß beam: training and testing results of KNN classifiers after TCA, JDA. Training accuracy is assessed using the validation set (15%) derived from FE data, whereas testing accuracy is measured using purely experimental data.

K	Distance metric	Training accuracy FE Data	Testing accuracy Experimental data (TCA)	Testing accuracy Experimental data (JDA)
3	Euclidian	100%	100%	96.6%
5	City block	100%	100%	100%
2	Chebyshev	100%	100%	98%
28	Minkowski (Cubic)	100%	100%	99%
1	Cosine	100%	100%	99.1%

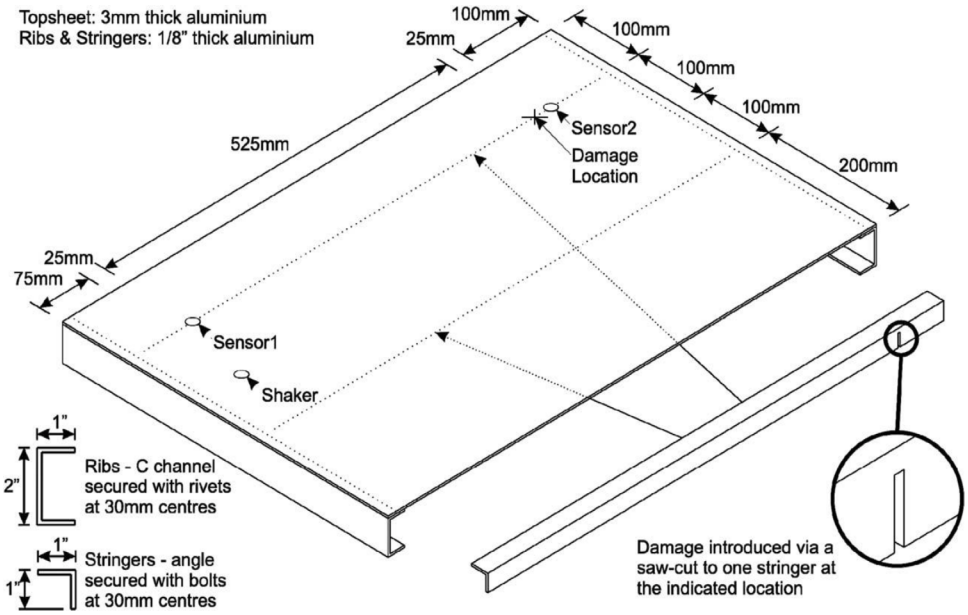


Fig. 11. Schematic diagram of experimental Wingbox structure [6].

Table 15
Order of damage scenarios of the wingbox structure.

Test	Type
Normal	Normal
Damage Class 1	25% saw-cut
Damage Class 2	50% saw-cut
Damage Class 3	75% saw-cut

In this experimental setup, a damage scenario was introduced in the form of a saw-cut. The location of the defect, in all cases, was at the outside stringer 125 mm from the edge of the panel. The recorded measurements were within a frequency range of 0–1600 Hz with a resolution of 0.5 Hz. In this case study, four damage cases were considered, which included normal and three different depths of saw-cuts (see Table 15). The saw-cut depths used were 5.5 mm, 11 mm, and 16.5 mm, respectively. Moreover, these saw-cut depths correspond to 25%, 50%, and 75% of the damaged stringer's width.

6.2. FE model: Wingbox structure

The FE model of the wingbox structure, also created using in-house MATLAB code, and was meshed with 11,254 MITC4 [31] shell elements of size h = 7.5 mm and 70,890 nodal degrees of freedom. In a similar fashion to joint modelling in the beam model (see Section 5.2.1), riveted and bolted joints of the wingbox are connected with spring interfaces. The material properties incorporated into this numerical model are outlined in Table 16. Because of the incorporation of shell elements, the complexity and the computational time for modal analysis were significantly reduced. Additionally, the experimental setup included 14 sensors distributed at various locations on the top aluminium plate. These sensor positions were employed in the FE model for recording

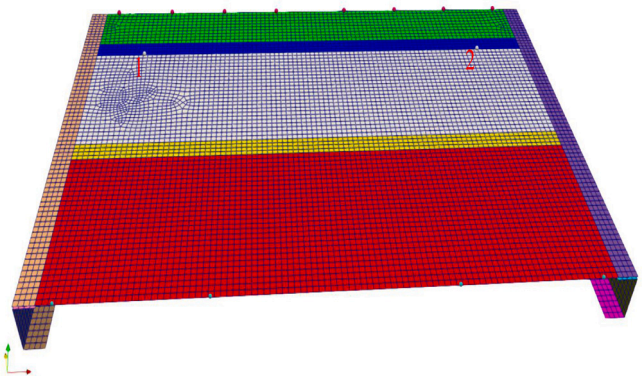


Fig. 12. FE model of the Wingbox structure with two sensor locations used for transmissibility features. Different element colours correspond to different groups used in the updating process.

Table 16
Material Properties of wingbox structure (Aluminium).

Property	Value
Young's Modulus (E)	70 GPa
Poisson's Ratio (ν)	0.3
Density (ρ)	2700 kg/m^3

Table 17
Material stiffness values for the main components of the wingbox structure before and after model updating.

Component	Young's Modulus (Before)	Young's Modulus (After)
Top plate parts	70 GPa	70–84 GPa
Stringer 1	70 GPa	76.09 GPa
Stringer 2	70 GPa	76.09 GPa
C-channel part 1	70 GPa	70.15 GPa
C-channel part 2	70 GPa	70.15 GPa

Table 18
Natural frequencies of the first 7 modes of the wingbox structure.

Mode	Updated numerical (Hz)	Experimental (Hz)	Abs. Difference (%)
1	23.87	24.70	3.36
2	68.39	69.99	2.28
3	108.34	104.31	3.86
4	116.29	117.56	1.08
5	117.51	123.99	5.22
6	181.42	185.86	2.38
7	199.28	197.39	0.95

the FRFs at their respective positions. The FE model of the wingbox, which geometrically coincides with the test setup, including quad mesh and sensor positions utilised in the analysis is displayed in Fig. 12.

In this case, a similar strategy of model updating to the beam model has been implemented [24,28]. The parameters selected for updating were the material stiffness of 11 element groups, illustrated in Fig. 12. Post model updating, the material stiffness for all element groups varied between 70 GPa and 84 GPa. During updating, stiffness values were constrained within a range of ± 20% relative to their original values using the method proposed by Zhang et al. [32]. The updated model was validated with the experimental data as can be seen in Table 18 with a comparison of the first seven natural frequencies. Since frequency domain features are mainly used here for the training of the classifiers and the frequency comparison was acceptable, the model was also considered acceptable. The updated material stiffness values for the main components of the wingbox structure are presented in Table 17. Damage modelling of the saw-cuts in the FE model can be acquired by stiffness reduction of elements at the position of damage which is on the outside stringer 125 mm from the edge of the panel (see Fig. 11).

There are four elements (on the FE model) present at the location of the saw-cut at the outside stringer. A 25% saw-cut on the stringer was simulated by modifying the stiffness of the bottom element, while a 50% saw-cut was emulated by adjusting the stiffness of the bottom two elements. To model the damage for a 75% saw-cut on the stringer, the stiffness of the bottom three elements was altered. The stiffness of elements in the respective cases was lowered by 99%. The combination of elements used for stiffness reduction is displayed in Fig. 13.

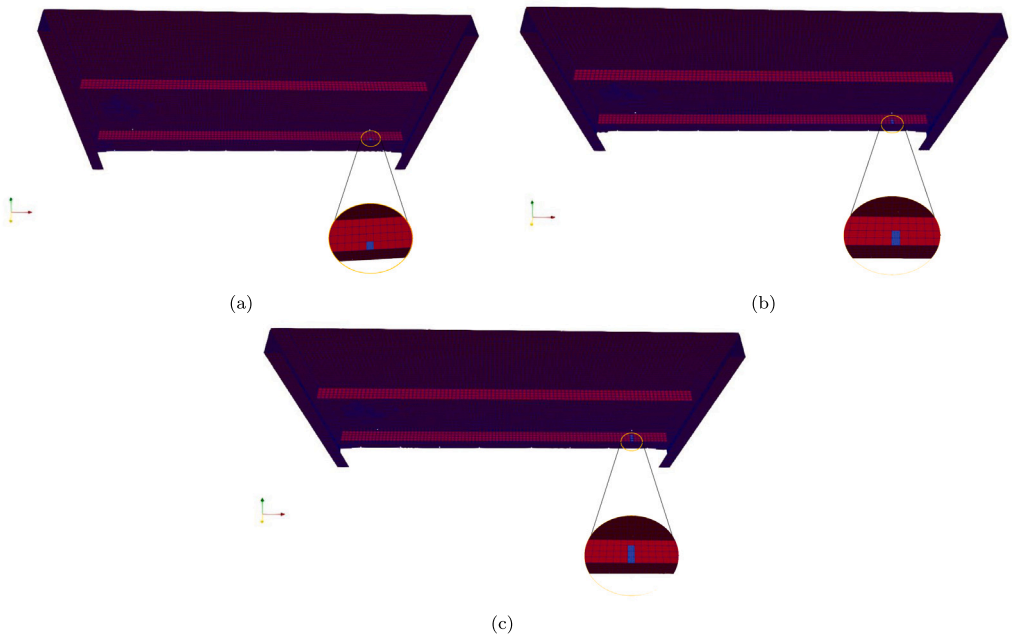


Fig. 13. Element selection for damage modelling in the Finite Element Wingbox Structure: (a) Elements chosen for stiffness reduction represent 25% saw-cut (b) Elements chosen for stiffness reduction represent 50% saw-cut (c) Elements chosen for stiffness reduction represent 75% saw-cut.

Table 19
Four-class problem: Neural networks with performance > 90% on experimental data of Wingbox structure before domain adaptation (DA), after application of TCA, and after JDA.

Total neural networks	Performance of neural networks		
	Before DA	After TCA	After JDA
800	0	776	768

During this phase, the Frequency Response Functions (FRFs) of the wingbox FE model were extracted at sensor positions 1 and 2, and transmissibilities are computed based on these FRFs. Repeating this procedure for data from all damage scenarios results in transmissibilities for respective damage classes. The features were then manually chosen using the FE transmissibilities, considering at least one feature per damage class (saw-cut depth). In total, four features are manually selected for evaluating Mahalanobis distances. The procedure depicted in Fig. 7 was repeated for this problem with each feature set of dimension 1000×50 (each source and target domain comprises 500 observations). Once the Mahalanobis distance features (input feature-vector data provided to classifiers) were obtained, both the training (FE) and testing (EXP) features for the classifiers have dimensions of 4×2000, with 500 features per class as indicated in Table 15. With the support of a 3-D scatter plot, a comparison of normalised features is performed. The first three principal components of features (classifier inputs) before the application of Domain Adaptation are shown in Figs. 14, 15 while features (first three principal components of classifier inputs) after the TCA application are illustrated in Fig. 16. More detailed views of the compared principal components after application of TCA are displayed in Figs. 17 and 18.

Data splitting for classification was carried out in a similar manner as described in Section 5. Subsequently, these feature data are used to train the KNN and neural network classifiers. The accuracy of the trained classifiers is then evaluated using the experimental feature data. Prior to the implementation of TCA, no networks in this four-class problem performed well on experimental data. After the application, 766 networks of 800 showed a high classification rate (> 90%) on experimental data. The fraction of networks performing well on experimental data increased from 0% to 97%. Table 19 displays the comparison of neural networks performing with good classification accuracy on experimental features, before and after the application of TCA and JDA.

Intuitively, to investigate the varying kernel sigma length (a TCA parameter), a range of values between 10–30 was chosen [11] keeping the regularisation parameter (μ) constant. This modification makes no significant change in the overall count of neural networks that demonstrated strong classification performance on experimental data. Interestingly, in this particular problem when using a *linear* kernel, the results demonstrate that varying lengths of sigma have negligible impact on the performance of the classifiers. Furthermore, by keeping kernel sigma constant and altering the regularisation parameter within the range of 1–10, there is no significant change in the number of neural networks achieving strong classification accuracy on experimental data.

For KNN classifiers, the 'K' value is fine-tuned in the training phase using Bayesian optimisation, considering various distance metrics (see Table 1). When presented with experimental features of the Wingbox structure, chosen KNN classifiers exhibited optimal

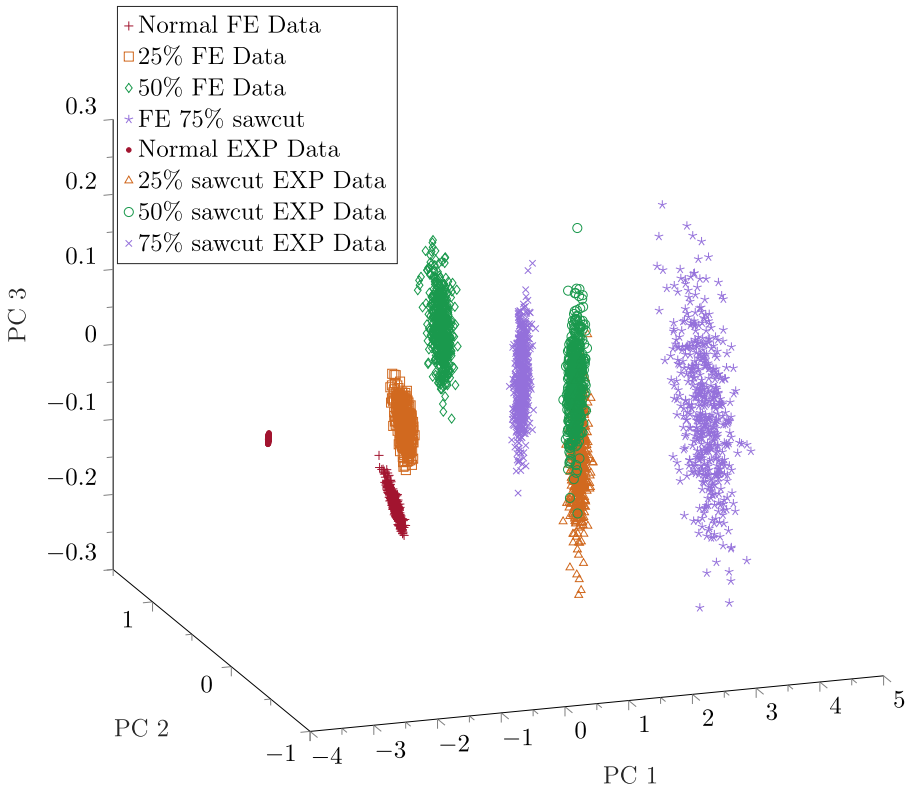


Fig. 14. Wingbox structure: comparison of first three principal components (using PCA) derived from input data of the four-class problem before application of TCA. Notably, 25% experimental saw-cut data and 50% experimental saw-cut data are in close proximity.

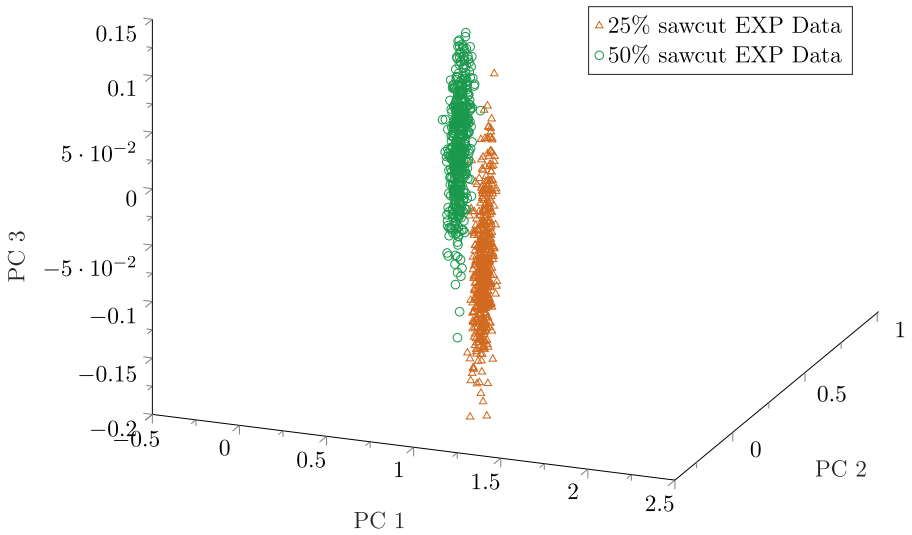


Fig. 15. Wingbox structure: zoomed-in view of first three principal components of classes 25% experimental saw-cut data and 50% experimental saw-cut data from Fig. 14 (before TCA).

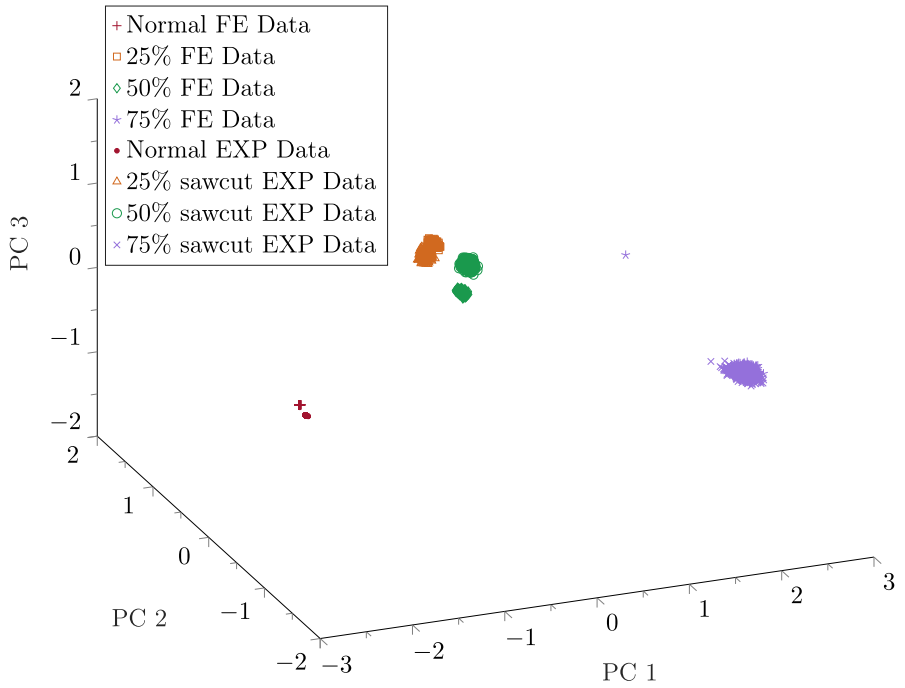


Fig. 16. Wingbox structure: comparison of first three principal components (using PCA) of input data of the four-class problem after application of TCA. Colour denotes a similar class (Normal, 25%, 50%, 75%).

Table 20
Four-class problem of Wingbox structure: training and testing results of KNN classifiers (with optimised 'K' values) before Domain Adaptation. Training accuracy is assessed using the validation set (15%) derived from FE data, whereas testing accuracy is measured using purely experimental data.

K	Distance metric	Training accuracy FE Data	Testing accuracy Experimental data
13	Euclidian	100%	48%
1	City block	100%	50%
22	Chebyshev	100%	37.5%
319	Minkowski (Cubic)	100%	48.4%
2	Cosine	100%	25%

performance (after Domain Adaptation). Tables 20 and 21 show how KNN classifiers performed on experimental data before and after the incorporation of domain adaptation (TCA and JDA).

Here, all KNN classifiers demonstrate exceptional effectiveness when TCA is applied. Classifiers prior to the application of domain adaptation do not exhibit better performance. The reduced performance of KNN classifiers can be possibly attributed to the misclassification of data because of confusion between the 25% experimental saw-cut and 50% experimental saw-cut classes. This can be seen in Figs. 14 and 15 where the classes 25% and 50% experimental saw-cut data are closely clustered.

7. Conclusion

The primary goal of this study was to address the issue of data availability for high-value structures in the context of supervised machine learning. FE modelling, combined with transfer component analysis (TCA) and joint domain adaptation (JDA) was applied here. Two case studies were employed, a nonlinear Brake–Reuß beam, where damage scenarios correspond to different torque settings on a lap joint, and a wingbox laboratory structure where damage was introduced as saw-cuts. In all cases in total, there were distinct scenarios of dual, three, four and five classes. TCA or JDA was always applied on normal data (both on the source domain – the FE data, and the target domain – the experimental data) meaning that no experimental data from damaged states were used for initial training. Any new data from a potentially-unknown class were projected into the latent space of the normal data. The classifiers used were standard Multi-Layer Perceptrons with one hidden layer and K-nearest neighbours (KNN).

All results showed that TCA was able to significantly improve the rate of classifiers which performed equally well in the FE and the experimental data. Visualisation of the input feature-data provided into the classifiers confirmed that TCA aligned the FE and experimental-data classes increasing the probability of having decision boundaries which work for both cases well. In the dual-class

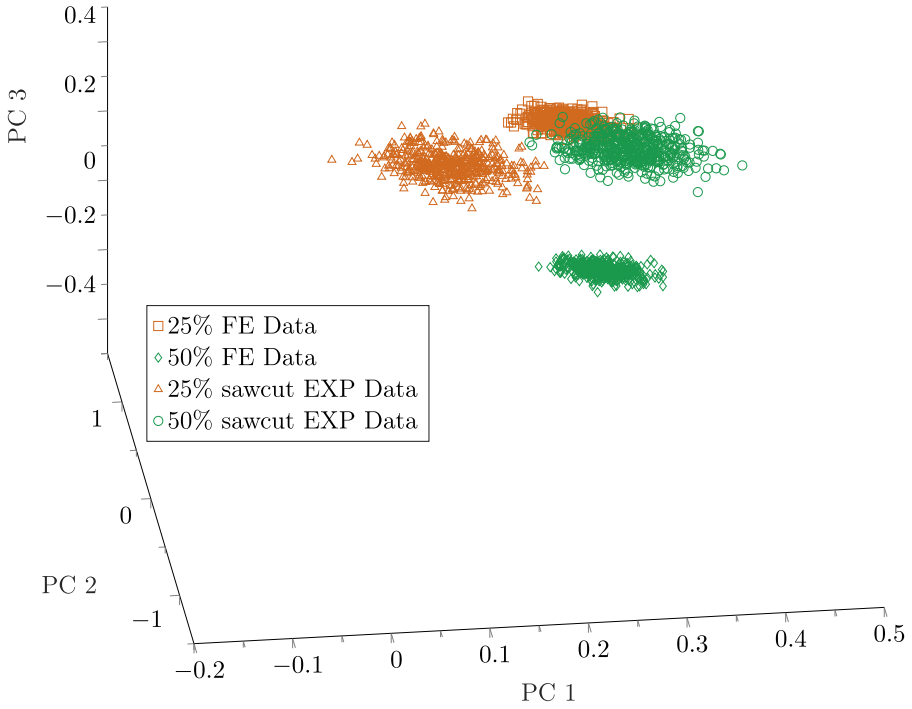


Fig. 17. Wingbox structure: zoomed-in view of Fig. 16, highlighting the first three principal components of input data (distance feature vectors) comparisons among the two classes (25% and 50% saw-cut; FE and experimental data) post-TCA application. Colour denotes a similar class (25% and 50%). This figure implies that the 25% FE Data and 50% sawcut data are near each other, yet a considerable distance actually separates these classes.

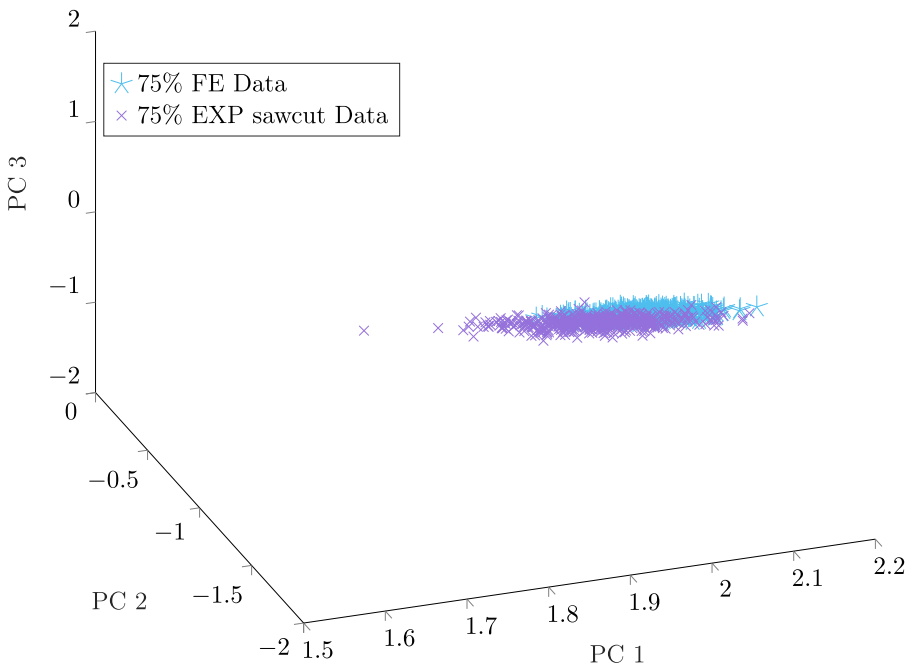


Fig. 18. Wingbox structure: zoomed-In view of Fig. 16, highlighting the first three principal components of input data (distance feature vectors) comparisons among 75% damage classes post-TCA application. Classes 75% FE Data and 75% experimental (EXP) saw-cut Data with different colours for discrimination.

Table 21
Four-class problem of Wingbox structure: training and testing results of KNN classifiers (with optimised 'K' values) after TCA, JDA. Training accuracy is assessed using the validation set (15%) derived from FE data, whereas testing accuracy is measured using purely experimental data.

K	Distance metric	Training accuracy FE Data	Testing accuracy Experimental data (TCA)	Testing accuracy Experimental data (JDA)
13	Euclidian	100%	100%	100%
1	City block	100%	100%	100%
22	Chebyshev	100%	99.9%	100%
319	Minkowski (Cubic)	100%	99.9%	100%
2	Cosine	100%	99.6%	100%

case (i.e. normal or damaged), good classification accuracy was shown on the experimental data both before and after applying TCA (out of 800 neural networks, 677 performed well before Domain Adaptation and all 800 afterwards). In the three-class problem (normal, damage scenario 1, damage scenario 2), TCA increased the ratio of neural networks working well on experimental data from 14.75% to 95.5%. KNN classifiers were also excellent. In the second case study, a wingbox laboratory structure was chosen to increase the structural complexity. In the four-class problem with different saw-cut lengths, TCA improved the ratio of neural networks which worked well on experimental data from 0% to 97%. All the deployed KNN classifiers demonstrated consistent performance on the experimental data, there was a negligible drop compared to the Brake–Reuß beam case (in TCA).

In summary, the results of these two case studies emphasise the advantages of employing domain adaptation to train supervised learning algorithms to identify experimental damage scenarios solely based on FE models and the data from normal (healthy) cases. The proposed methodology, when applied with JDA, has shown a similar capacity to yield outcomes compared to TCA. However, the kernel type used is non-linear '*radial basis function*'. The inclusion of more classes in the damage scenarios will likely complicate the feature selection and will require more complex decision boundaries from the classifiers. As long as discriminating features can be extracted and domain adaptation is possible the approach will work; however, this can be investigated in future work which can also involve more advanced feature selection methods and explore the sensitivity of the approach in terms of the damage severity cases.

CRediT authorship contribution statement

Raja Sekhar Battu: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Konstantinos Agathos:** Writing – review & editing, Software, Methodology. **Julián Mauricio Londoño Monsalve:** Writing – review & editing, Data curation. **Keith Worden:** Writing – review & editing, Funding acquisition. **Evangelos Papatheou:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

Data availability

Data will be available upon request from repository 10.6084/m9.figshare.26955016.

Acknowledgements

The support of the UK Engineering and Physical Sciences Research Council (EPSRC), United Kingdom from grant reference EP/W005816/1 is gratefully acknowledged. For the purposes of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

References

[1] S.W. Doebling, C.R. Farrar, M.B. Prime, D.W. Shevitz, Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in Their Vibration Characteristics: A Literature Review, Tech. Rep., Los Alamos National Laboratory, Los Alamos, New Mexico, 1996, pp. 1–134, http://dx.doi.org/10.2172/249299.

[2] H. Sohn, C.R. Farrar, F.M. Hemez, D.D. Shunk, D.W. Stinemates, B.R. Nadler, J.J. Czarnecki, A Review of Structural Health Monitoring Literature: 1996 – 2001, Tech. Rep. LA-13976-M, Los Alamos National Laboratory, Los Alamos, New Mexico, 2004, pp. 1–311.

[3] C.R. Farrar, K. Worden, Structural Health Monitoring: A Machine Learning Perspective, first ed., John Wiley and Sons, 2012, http://dx.doi.org/10.1002/9781118443118.

[4] K. Worden, G. Manson, The application of machine learning to structural health monitoring, Phil. Trans. R. Soc. A 365 (1851) (2007) 515–537, http://dx.doi.org/10.1098/rsta.2006.1938.

[5] A. Malekloo, E. Ozer, M. AlHamaydeh, M. Girolami, Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights, Struct. Health Monit. 21 (4) (2022) 1906–1955, http://dx.doi.org/10.1177/14759217211036880.
[6] E. Papatheou, G. Manson, R.J. Barthorpe, K. Worden, The use of pseudo-faults for novelty detection in SHM, J. Sound Vib. 329 (2010) 2349–2366, http://dx.doi.org/10.1016/j.jsv.2009.07.020.
[7] E. Papatheou, G. Manson, R.J. Barthorpe, K. Worden, The use of pseudo-faults for damage location in SHM: An experimental investigation on a Piper Tomahawk aircraft wing, J. Sound Vib. 333 (2014) 971–990, http://dx.doi.org/10.1016/j.jsv.2013.10.013.
[8] A. Pagani, M. Enea, E. Carrera, Component-wise damage detection by neural networks and refined FEs training, J. Sound Vib. 509 (2021) 116255, http://dx.doi.org/10.1016/j.jsv.2021.116255.
[9] A. Pagani, M. Enea, Displacement and strain data-driven damage detection in multi-component and heterogeneous composite structures, Mech. Adv. Mater. Struct. 29 (2022) 1–16, http://dx.doi.org/10.1080/15376494.2022.2149907.
[10] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359, http://dx.doi.org/10.1109/tkde.2009.191.
[11] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2011) 199–210, http://dx.doi.org/10.1109/tnn.2010.2091281.
[12] D. Chakraborty, N. Kovvali, B. Chakraborty, A. Papandreou-Suppappola, A. Chattopadhyay, Structural damage detection with insufficient data using transfer learning techniques, in: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, Vol. 7981, Proc. of SPIE, 2011, pp. 1175–1183, http://dx.doi.org/10.1117/12.882025.
[13] P. Gardner, X. Liu, K. Worden, On the application of domain adaptation in structural health monitoring, Mech. Syst. Signal Process. 138 (2020) 106550, http://dx.doi.org/10.1016/j.ymssp.2019.106550.
[14] J. Poole, P. Gardner, N. Dervilis, L. Bull, K. Worden, On statistic alignment for domain adaptation in structural health monitoring, Struct. Health Monit. 22 (3) (2023) 1581–1600, http://dx.doi.org/10.1177/14759217221110441.
[15] K. Worden, L.A. Bull, P. Gardner, J. Gosliga, T.J. Rogers, E.J. Cross, E. Papatheou, W. Lin, N. Dervilis, A brief introduction to recent developments in population-based structural health monitoring, Front. Built Environ. 6 (146) (2020) 547125, http://dx.doi.org/10.3389/fbuil.2020.00146.
[16] L.A. Bull, P.A. Gardner, N. Dervilis, E. Papatheou, M. Haywood-Alexander, R.S. Mills, K. Worden, On the transfer of damage detectors between structures: An experimental case study, J. Sound Vib. 501 (2021) 116072, http://dx.doi.org/10.1016/j.jsv.2021.116072.
[17] L.A. Bull, D. Di Francesco, M. Dhada, O. Steinert, T. Lindgren, A.K. Parlikad, A.B. Duncan, M. Girolami, Hierarchical Bayesian modeling for knowledge transfer across engineering fleets via multitask learning, Comput. Aided Civ. Infrastruct. Eng. 38 (7) (2023) 821–848, http://dx.doi.org/10.1111/mice.12901.
[18] A.I. Ozdagli, X. Koutsoukos, Domain adaptation for structural health monitoring, in: Annual Conference of the PHM Society, 2020, pp. 1–9, http://dx.doi.org/10.36001/phmconf.2020.v12i1.1184.
[19] A.I. Ozdagli, X. Koutsoukos, Domain adaptation for structural fault detection under model uncertainty, Int. J. Progn. Health Manag. 12 (2) (2021) 1–13, http://dx.doi.org/10.36001/ijphm.2021.v12i2.2948.
[20] P. Martakis, Y. Reuland, A. Stavridis, E. Chatzi, Fusing damage-sensitive features and domain adaptation towards robust damage classification in real buildings, Soil Dyn. Earthq. Eng. 166 (2023) 107739, http://dx.doi.org/10.1016/j.soildyn.2022.107739.
[21] S.d. Silva, M.O. Yano, C.G. Gonsalez-Bueno, Transfer component analysis for compensation of temperature effects on the impedance-based structural health monitoring, J. Nondestruct. Eval. 40 (3) (2021) 1–17, http://dx.doi.org/10.1007/s10921-021-00794-6.
[22] H. Xiao, H. Ogai, W. Wang, A new deep transfer learning method for intelligent bridge damage diagnosis based on muti-channel sub-domain adaptation, Struct. Infrastruct. Eng. - Maint., Manag., Life-Cycle Des. Perform. (2023) 1–16, http://dx.doi.org/10.1080/15732479.2023.2167214.
[23] M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, Transfer feature learning with joint distribution adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, Institute of Electrical and Electronics Engineers Inc., 2013, pp. 2200–2207, http://dx.doi.org/10.1109/iccv.2013.274.
[24] J.E. Mottershead, M. Link, M.I. Friswell, The sensitivity method in finite element model updating: A tutorial, Mech. Syst. Signal Process. 25 (2011) 2275–2296, http://dx.doi.org/10.1016/j.ymssp.2010.10.012.
[25] K. Worden, G. Manson, N.R. Fieller, Damage detection using outlier analysis, J. Sound Vib. 229 (2000) 647–667, http://dx.doi.org/10.1006/jsvi.1999.2514.
[26] R. Lacayo, L. Pesaresi, J. Groß, D. Fochler, J. Armand, L. Salles, C. Schwingshackl, M. Allen, M. Brake, Nonlinear modeling of structures with bolted joints: A comparison of two approaches based on a time-domain and frequency-domain solver, Mech. Syst. Signal Process. 114 (2019) 413–438, http://dx.doi.org/10.1016/j.ymssp.2018.05.033.
[27] SDTools – Product and company, URL https://www.sdtools.com/.
[28] R. Battu, K. Agathos, C. Smith, E. Papatheou, Robust training databases for supervised learning algorithms in structural health monitoring applications, in: Conference Proceedings of ISMA2022 - USD2022, K U Leuven, Leuven, 2022, pp. 3671–3681.
[29] E. Papatheou, C. Lord, N. Dervilis, D. Wagg, K. Worden, An exploratory study of the suitability of a wind turbine blade as a nonlinear demonstrator, in: Conference Proceedings of ISMA2016, Leuven, Belgium, 2016, pp. 4069–4080.
[30] L.A. Bull, P.A. Gardner, N. Dervilis, E. Papatheou, M. Haywood-Alexander, R.S. Mills, K. Worden, Transferring damage detectors between tailplane experiments, in: Conference Proceedings of the Society for Experimental Mechanics Series, Vol. 9, Springer, 2022, pp. 199–211, http://dx.doi.org/10.1007/978-3-030-76004-5{_}22.
[31] K.-J. Bathe, E.N. Dvorkin, A four-node plate bending element based on Mindlin/Reissner plate theory and a mixed interpolation, Internat. J. Numer. Methods Engrg. 21 (2) (1985) 367–383, http://dx.doi.org/10.1002/nme.1620210213.
[32] Q. Zhang, C. Chang, T. Chang, Finite element model updating for structures with parametric constraints, Earthq. Eng. Struct. Dyn. 29 (7) (2000) 927–944, http://dx.doi.org/10.1002/1096-9845(200007)29:7<927::AID-EQE955>3.0.CO;2-4.