---

**Article:**
Gardner, P. orcid.org/0000-0002-1882-9728, Bull, L.A. orcid.org/0000-0002-0225-5010, Dervilis, N. orcid.org/0000-0002-5712-7323 et al. (1 more author) (2021) Overcoming the problem of repair in structural health monitoring: Metric-informed transfer learning. Journal of Sound and Vibration, 510. 116245. ISSN 0022-460X

https://doi.org/10.1016/j.jsv.2021.116245

---

Journal of Sound and Vibration 510 (2021) 116245

Contents lists available at ScienceDirect

Journal of Sound and Vibration

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

# Overcoming the problem of repair in structural health monitoring: Metric-informed transfer learning

P. Gardner *, L.A. Bull, N. Dervilis, K. Worden

*Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Sheffield S1 3JD, UK*

ARTICLE INFO

*Keywords:*
Transfer learning
Domain adaptation
Population-based structural health monitoring

ABSTRACT

Structural repairs alter the physical properties of a structure, changing its responses, both in terms of its normal condition and of its different damage states. This difference in responses manifests itself as a shift between the pre- and post-repair data distributions, which can be problematic for conventional data-driven approaches to structural health monitoring (SHM), and limits their effectiveness in industrial applications. This limitation occurs typically because approaches assume that the data distribution is the same in training as appears in testing; with an algorithm failing to generalise when this assumption is not true; that is, pre-repair labels no longer apply to the post-repair data. Transfer learning, in the form of domain adaptation, proposes a solution to this issue, by mapping the pre- and post-repair data distributions onto a shared latent space where their distributions are approximately equal, allowing pre-repair label knowledge to be used to classify the post-repair data. This paper demonstrates the applicability of domain adaptation as a method for overcoming the problem of repair on a dataset from a Gnat trainer aircraft. In addition, a novel modification to an existing domain adaptation technique – joint distribution adaptation – is proposed, which seeks to improve the semi-supervised learning phase of the algorithm by considering a metric-informed procedure. The metric-informed joint distribution adaptation algorithm is benchmarked against, and shown to outperform, both conventional data-based approaches and other domain adaptation techniques.

## 1. Introduction

Repairing a structure is a crucial option in any asset management process, and vital for the structure to remain operating safely. However, structural repairs and modifications cause changes to the physical properties of a system that will affect its structural responses. This issue is problematic for data-based approaches to structural health monitoring (SHM) [1], which typically assume that the overall structural response of the system under normal operation and for various health states will stay the same over time, i.e. that the data distributions pre- and post-repair are the same for all damage states of interest. As a result, when a structure is repaired, a new data-based model must be constructed that captures the behaviour of the repaired structure. This requirement means that all previously-acquired data relating to various damage states will no longer be applicable (because of the shift in data distributions). Consequently, a new labelling campaign would be required every time the structure is repaired, which will often be infeasible for most industrial applications. One solution to this issue is to utilise transfer learning, transferring label knowledge from the pre-repair dataset to the post-repair unlabelled data via unsupervised domain adaptation. This technology means any knowledge obtained pre-repair can be utilised for the post-repair structure, creating more optimal and efficient data-based algorithms.

* Corresponding author.
*E-mail address:* p.gardner@sheffield.ac.uk (P. Gardner).

https://doi.org/10.1016/j.jsv.2021.116245
Received 4 September 2020; Received in revised form 29 April 2021; Accepted 25 May 2021
Available online 5 June 2021

Domain adaptation is one branch of transfer learning that seeks to map feature data from a labelled source dataset (here the pre-repair dataset) to an unlabelled target dataset (the post-repair dataset). By learning a mapping on the datasets, typically into a shared latent subspace, label information can be transferred, meaning knowledge is not lost after a structural repair. Domain adaptation techniques have recently been investigated in several contexts. Gardner et al. utilised domain adaptation for performing population-based SHM [2,3] – the process of transferring inferences between different structures [3–5] – and Fink et al. discuss how domain adaptation can be applied to fleet prognostics and health management (PHM) between critical systems [6]. In addition, domain adaptation has been utilised in rotating machinery contexts, transferring fault diagnostics between different loading scenarios [7–9] and between different machines [10]. Moreover, Chakraborty et al. developed a technique for transferring knowledge between nodes in a sensor network, overcoming issues associated with a faulty sensor in the network. The current paper explores the novel application of utilising domain adaptation in overcoming system changes caused by structural repairs.

Several transfer learning and domain adaptation algorithms exist within the literature [7,8,11–22], with the focus here being on methods that consider manipulations of the feature space, particularly in reducing the distance between the source and target distributions in a shared latent space [14,15,17–19,21]. In addressing the issue of repair problems in SHM, a novel modification to joint distribution adaptation (originally introduced by Long et al.) [17], is proposed. Joint distribution adaptation is an unsupervised domain adaptation algorithm that seeks to find a mapping into a shared latent space where the distance between the joint distributions of the source and target datasets are minimised. The contribution in this paper introduces a new approach for obtaining initial pseudo-labels for the unlabelled target data using a metric-informed approach, rather than naïve self-labelling as in [17]. In addition, a code implementation accompanies this paper — https://github.com/pagard/EngineeringTransferLearning.

The outline of this paper is as follows. Section 2 discusses the problems structural repairs cause for data-based SHM methods and introduces the Gnat aircraft dataset. Existing domain adaptation techniques are applied in Section 3 to features extracted from previous analysis on the Gnat dataset in the literature [23,24], benchmarking their classification performance. Given that these extracted features required damage-state knowledge from the pre- and post-repair states in the analysis [23,24], Section 4 explores using domain adaptation as a feature extraction and transfer learning tool for 'repair' scenarios on the Gnat aircraft. In the light of these analyses, a novel modification to joint distribution adaptation is introduced, namely *metric-informed* joint distribution adaptation, in Section 5. The method is applied to both feature sets in Section 6, where metric-informed joint distribution adaptation is shown to outperform existing techniques. Finally, Section 7 states the conclusions from the paper.

## 2. Challenges introduced by structural repairs

Data-based approaches to structural health monitoring (SHM) are generally constructed based on the assumption that both training and test data are drawn from the same underlying distribution. This assumption allows the engineer to train a machine learner on data describing a set of damage states, knowing that the learner will generalise, meaning any new examples of those damage states will be correctly classified. However, the assumption that both training and test data are from the same underlying distribution is too strong an assumption for many real-world applications. Many different phenomena can affect the structure or system of interest, resulting in a different underlying distribution, such as structural repairs, mounting equipment or changing the configuration of the structure, new environmental or operational conditions, etc. It should be noted that changes due to environmental and operational variations (EoVs) are not investigated in this paper, as these typically involve understanding changes to the data distribution with relation to a dependent variable [25], e.g. the loading condition, therefore requiring a different approach to that proposed in this paper.

Fig. 1 illustrates the effect of changing data distributions, known as *domain shift*, where the original training data – here denoted as the *source* data – has an overall different distribution to new observations — here called the *target* data. It is clear from the figure that a machine learner trained on the source data will not generalise to the target data, due to differences in the joint distribution of the data. Conventional machine learning may show reasonable classification results for some classes, such as Class One, where the conditional distributions are only slightly shifted. However, this behaviour would be concerning, masking the invalidation of the assumption that the training and test data are drawn from the same distribution. The problem is most clear for Classes Two and Three, where the changes in conditional distribution mean that part of Class Two (target) is now overlapping with Class Three (source) training data, which would lead to misclassification.

Structural repair is a phenomenon that causes changes in the underlying distribution of the data. For example, replacing components or applying repair materials, alters the mass, stiffness and damping properties of the system, meaning that the physical structure is different to its pre-repair state. In a structural health monitoring context, repairs will be an accepted part of asset management, with the structure being repaired, when necessary, throughout the lifespan of the structure. Any data-based strategy must therefore account for the fact that repairs will take place, otherwise the structural health monitoring procedure will be invalid each time a repair is performed. This fact will lead to a costly retraining campaign each time a structural modification is made, as well as making all previously-labelled data points redundant, as they refer to the previous data distribution. The following section focusses on the problem of repairing a structure in the specific context of a wing on a Gnat trainer aeroplane.

### 2.1. Gnat aircraft dataset

The Gnat aircraft dataset comes from an experimental campaign in which a network of uni-axial accelerometers were used to obtain transmissibilities from the structure under different pseudo-damage scenarios (as it was not possible to truly damage the structure), relating to the removal of nine inspection panels [23,26,27]. During the experiments, a broadband white-noise

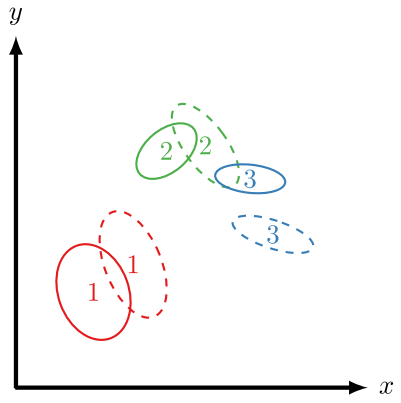


**Fig. 1.** An example of changes in the underlying distribution of a system; source (–) and target (––).

**Table 1**
Accelerometers used to form transmissibilities and their associated inspection panel [23].

| Panel | Associated transmissibility | Reference accelerometer | Response accelerometer |
|---|---|---|---|
| P1 | T1 | AR | A1 |
| P2 | T2 | AR | A2 |
| P3 | T3 | AR | A3 |
| P4 | T4 | BR | B1 |
| P5 | T5 | BR | B2 |
| P6 | T6 | BR | B3 |
| P7 | T7 | CR | C1 |
| P8 | T8 | CR | C2 |
| P9 | T9 | CR | C3 |

**Table 2**
Test configurations in experimental campaign indicating which panels were removed and their associated class label $Y$. Class label $Y = 0$ refers to the normal condition, i.e. no panels were removed.

| Sequence | Repeat | Configurations | | | |
|---|---|---|---|---|---|
| 1 | 1 | Normal; $Y = 0$ | P1; $Y = 1$ | P2; $Y = 2$ | P3; $Y = 3$ |
| 2 | 2 | Normal; $Y = 0$ | P1; $Y = 1$ | P2; $Y = 2$ | P3; $Y = 3$ |
| 3 | 1 | Normal; $Y = 0$ | P4; $Y = 4$ | P5; $Y = 5$ | P6; $Y = 6$ |
| 4 | 2 | Normal; $Y = 0$ | P4; $Y = 4$ | P5; $Y = 5$ | P6; $Y = 6$ |
| 5 | 1 | Normal; $Y = 0$ | P7; $Y = 7$ | P8; $Y = 8$ | P9; $Y = 9$ |
| 6 | 2 | Normal; $Y = 0$ | P7; $Y = 7$ | P8; $Y = 8$ | P9; $Y = 9$ |
| 7 | 1 | Normal; $Y = 0$ | | | |

excitation was applied to the structure via an electrodynamic shaker below inspection panel four, on the bottom surface of the wing. A schematic of the aircraft wing, depicting the locations of inspection panels, accelerometers and transmissibility paths is shown in Fig. 2. Each transmissibility path was designed to target a particular inspection panel, i.e. the reference (denoted R) and target accelerometers define a path that crosses one particular inspection panel; the panels and their associated transmissibilities are presented in Table 1.

There were two categories of inspection panel, a set of small panels {P3, P6} – both with an area of 0.00825 $m^2$ [23] – and a set of larger panels {P1, P2, P4, P5, P7, P8, P9} — with areas greater than 0.0176$m^2$ [23]. Each transmissibility covered a frequency range of 1024–2048 Hz containing 1024 spectral lines. For more details about the experimental campaign, the interested reader is referred to [23,26,27].

The test strategy involved applying damage sequentially, achieved by removing inspection panels covered by each transducer group, A, B and C respectively. The 25 configurations investigated are presented in Table 2, grouped into seven sequences of measurements, each containing data from the normal condition, with sequences one to six containing damage states. For each of the 25 configurations, 100 repeat measurements were obtained, totalling 2500 measurements (700 of the normal condition and 1800 measurements of damage states) over the complete experimental campaign. The real and imaginary parts of the transmissibilities were converted to magnitudes and the phase was discarded [23].

The effects of removing and reattaching inspection panels are similar to those from structural changes introduced by performing repairs. In fact, removing and reattaching inspection panels is typically a step required in performing maintenance to the aircraft wing. For this reason, the experiential dataset is used to demonstrate the typical changes in data distributions encountered during an SHM campaign involving structural repairs.

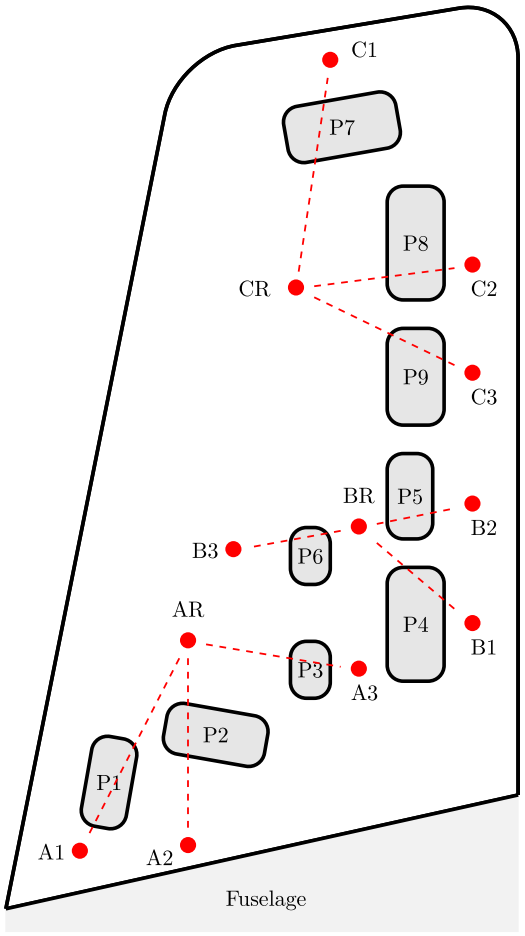


**Fig. 2.** A representative schematic of the Gnat aircraft starboard wing (not to scale), indicating the locations of inspection panels, accelerometers and transmissibility paths.
*Source:* Recreated from [23].

The main source of variability between configuration repeats, e.g. Sequence One and Two, arises from uncertainties in the boundary conditions of the inspection panels, despite the torque on the fastenings being controlled for the configurations [23]. Fig. 3 presents a comparison of transmissibility $T_1$, for Sequences One and Two in Table 2, for the normal condition $Y = 0$ (top panel) and the removal of panel one $Y = 1$ (bottom panel). It is clear that there are significant changes in the response between the repeat configurations in Sequences One and Two for both the normal and damage conditions. In fact, the changes due to the 'repair' are *larger* than those associated with damage. This domain shift, which appears for all transmissibilities and damage conditions, causes difficulties in applying a machine learner trained from one repeat, to another. It is clear that just removing and replacing the same inspection panels causes larger variations than those caused by damage, motivating the need for a novel approach to data-based SHM in these repair scenarios.

### 2.2. The potential of transfer learning

Transfer learning is a branch of machine learning that seeks to overcome issues associated with domain shift, which can occur when a structure is repaired. The approaches leverage information from a group of datasets, in aiding a task on a different dataset [11–13]. One subcategory of approach within transfer learning is *domain adaptation*, seeking to utilise a source dataset $\{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{N_s}$ in aiding the predictive function $f_t(\cdot)$ on a target dataset $\{\boldsymbol{x}_i^t, y_i^t\}_{i=1}^{N_t}$, where the marginal distributions $p(X_s) \neq p(X_t)$, and potentially the conditional distributions $p(\boldsymbol{y}_s \mid X_s) \neq p(\boldsymbol{y}_t \mid X_t)$, are different (as seen in Fig. 1) [11]. In addition, domain adaptation assumes that the feature and label spaces are equal, $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$, respectively; where $X$ and $\boldsymbol{y}$ contain finite samples from $\mathcal{X}$ and $\mathcal{Y}$. Typically, domain adaptation techniques are interested in the scenario where the target domain is completely unlabelled, i.e. the target dataset is $\{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$, which occurs in a repair scenario.

It is clear that the scenario introduced in Section 2.1 leads to a problem suited to domain adaptation, namely that the joint distribution is different between repair conditions, as demonstrated in Fig. 3. For this reason, domain adaptation algorithms are

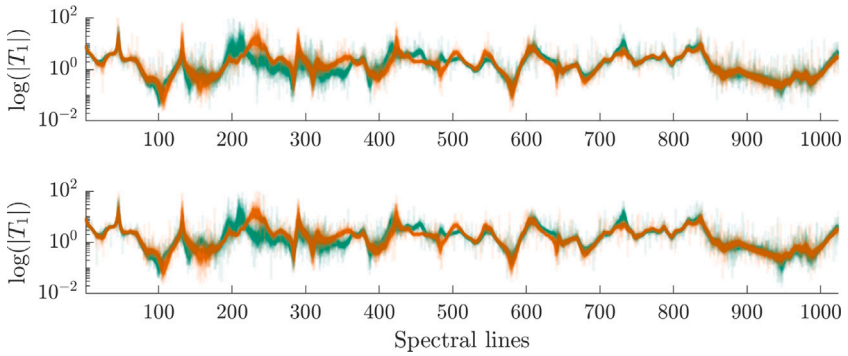


**Fig. 3.** Comparison of transmissibility $T_1$ for before (—) and after (—) repeat configuration measurements (sequence one and two in Table 2 respectively). Top and bottom panels depict the normal condition and class one respectively.

**Table 3**
Nine features selected from a genetic algorithm in [24].

| Feat. no. | 1 | 2 | 3 | 7 | 9 | 28 | 31 | 33 | 44 |
|---|---|---|---|---|---|---|---|---|---|
| Trans. | T2* | T2* | T1* | T4* | T5 | T7* | T7* | T8* | T9* |
| Spectral lines | 342–351 | 667–676 | 845–874 | 249–258 | 195–214 | 757–766 | 331–340 | 327–336 | 935–944 |

*Denotes the inverse transmissibility.

explored here, mapping source and target datasets from each repeat configuration onto a new feature space where both datasets lie on top of each other, i.e. the distance between data distributions is minimised. This process creates a new feature space where label information can be transferred from the source to the target domain, therefore overcoming issues associated with structural repairs in SHM campaigns.

## 3. Overcoming a repair problem on a Gnat aircraft wing: features derived from engineering-informed knowledge

The Gnat aircraft wing dataset has been explored several times in the literature [23,24,27–29]. In these previous works, the problem of repeat measurements within the sequence of data (mimicking a repair scenario) were overcome by creating a feature space from the interleaved dataset [23,24,28,29]. However, this strategy is often unsatisfactory in many real-world applications, as examples of all repair configurations are not normally available at the start of the SHM campaign. For the sake of continuity, and to benchmark domain adaptation against previous work, the feature space from past analysis is initially considered.

The feature space is based on work in [23], where engineering judgement was used to select features for each panel that were sensitive to that panel alone. The features were based on discordancy measures, in this case Mahalanobis squared-distances (MSDs), where the aim of each feature was to be sensitive to outliers from a particular panel removal. The Gaussian model for each MSD was comprised of the complete set of 700 normal condition observations covering a particular range of spectral lines for a chosen transmissibility (given that novelty detection was successfully performed in [27]). For example, a feature that was sensitive to the removal of P2 was composed from the inverse of Transmissibility One, considering the spectral line range 845–874 [23]. In total, 44 damage sensitive features were identified, each selected considering the interleaved dataset and engineering judgement — the interested reader is referred to [23] for an in-depth outline of how the features were determined and created. The authors state that a semi-objective criteria, based on desired outlier statistics, was used to select nine features (one feature for each panel), that would produce a separable feature space for the localisation problem. A multilayer perceptron (MLP) classifier was subsequently applied to the feature space — where the complete dataset was divided into three equal sections: training, validation and testing. The MLP produced a localisation accuracy of 86.5% on the independent test set [23].

A second analysis was performed in [24] with the aim of producing a more objective criteria for feature selection. This analysis used a genetic algorithm (GA) to select the nine most ‘optimal’ features from the 44, where the cost function was constructed from the accuracy of an MLP classifier trained on the candidate features. The GA-selected features, shown in Table 3, for the complete set of sequences, were again separated into three equal datasets: training, validation and testing. An MLP produced a localisation accuracy of 98.1% on the interleaved independent test set [24] (combining both pre- and post-repair data), showing an improvement on the semi-objective feature set in [23].

The first two principal components of the feature space are presented in Fig. 4, where the dataset has been divided into pre- and post-repair datasets (rather than the training, validation and test partitions in [24]). The pre-repair dataset comprises of sequences $\{1,3,5\}$, i.e. the first repeat of each damage scenario, with the post-repair dataset composed from sequences $\{2,4,6\}$, i.e. the second repeat for each damage scenario (where all 700 normal condition observations have been used in constructing the features).

This feature space was extracted, given knowledge of the damage condition for both the pre- and post-repair conditions, which is often not obtainable *a priori*, hence the need for transfer learning. However, given this engineered feature space, it can be seen

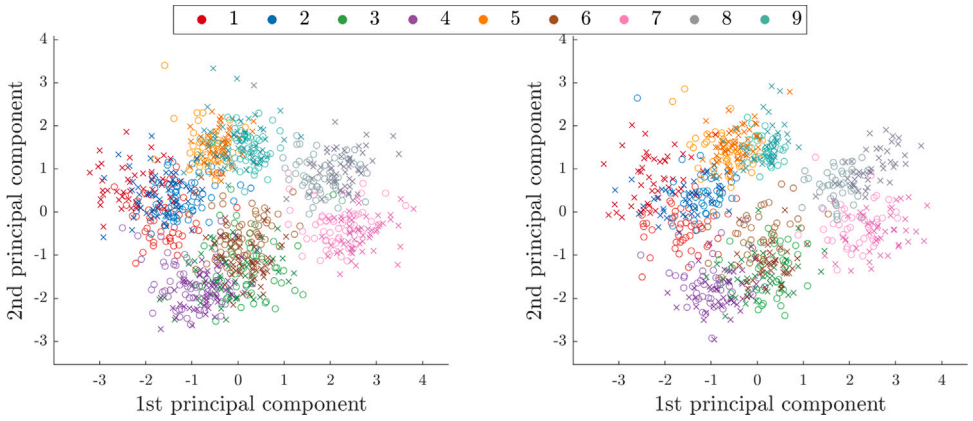


**Fig. 4.** The first two principal components for nine GA-selected features. Pre-repair (source) data (×) and post-repair (target) data (∘) are depicted where the left and right panels show the mapping training and testing data.

that both the pre- and post-repair observations have similar distributions, meaning that a conventional machine learning strategy is able to generalise well from the pre-repair source data to the post-repair target data, hence the results in [24,27]. However, even for this highly-engineered feature space, where both pre- and post-repair data have been used, there are slight discrepancies in the distributions between pre- and post-repair states, meaning that domain adaptation can be used to enhance classification performance when compared to a conventional machine learning approach; this is explored by applying domain adaptation techniques, mainly via joint distribution adaptation [17], which is outlined next.

### 3.1. Joint distribution adaptation

Joint distribution adaptation is an algorithm designed to map labelled source and unlabelled target datasets onto a space where the distance between their joint distributions is minimised [17]. By minimising the distance between the source and target joint distributions in an unsupervised manner, JDA is able to find a mapping where the source and target distributions are approximately equal, meaning that the source and target datasets will lie on top of each other in the latent space, allowing label information to be transferred from the source to the target domains. The approach infers a nonlinear transformation $\phi$ from the original feature space onto a *reproducing kernel Hilbert space* (RKHS), i.e. $\phi : \mathcal{X} \to \mathcal{H}$, where $p(\boldsymbol{y}_s, \phi(X_s)) \approx p(\boldsymbol{y}_t, \phi(X_t))$. However, the target conditional distribution $p(\boldsymbol{y}_t \mid X_t)$ is not obtainable given an unlabelled target domain. Target pseudo-labels $\hat{\boldsymbol{y}}_t$ are therefore estimated and used in approximating the joint distribution. Furthermore, the posterior probabilities, $p(\boldsymbol{y}_s \mid X_s)$ and $p(\boldsymbol{y}_t \mid X_t)$, are difficult and computationally expensive to obtain. As a result, JDA utilises the class conditional distributions, $p(X_s \mid \boldsymbol{y}_s = c)$ and $p(X_t \mid \hat{\boldsymbol{y}}_t = c)$, where $c = \{1, \ldots, C\}$ in the class label set $\mathcal{Y}$, as a substitute for the conditional distribution. Consequently, JDA minimises the distance between the marginal distributions, $p(\phi(X_s))$ and $p(\phi(X_t))$, and the class conditional distributions, $p(\phi(X_s) \mid \boldsymbol{y}_s = c)$ and $p(\phi(X_t) \mid \hat{\boldsymbol{y}}_t = c)$, simultaneously in order to find an optimal mapping $\phi$ that approximately minimises the distance between the source and target joint distributions.

The distance criterion utilised in JDA is the (squared) *maximum mean discrepancy* (MMD) distance. This distance between distributions is calculated by assessing the difference between two empirical means, via a nonlinear mapping into an (RKHS), specifically via a kernel, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathsf{T}}\phi(\boldsymbol{x}_j)$ [30]. For the marginal distributions, the distance can be formed as,

$$D(p(X_s), p(X_t)) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \phi(\boldsymbol{x}_{s,i}) - \frac{1}{N_t} \sum_{i=1}^{N_t} \phi(\boldsymbol{x}_{t,i}) \right\|_{\mathcal{H}}^2 = \mathrm{tr}(KM) \tag{1}$$

where $K = \phi(X)^{\mathsf{T}}\phi(X) \in \mathbb{R}^{(N_s+N_t)\times(N_s+N_t)}$ given $X = X_s \cup X_t \in \mathbb{R}^{(N_s+N_t)\times d}$ where $d$ is the dimension of the features. The matrix $M$ describes the empirical mean defined as,

$$M(i,j) = \begin{cases} \frac{1}{N_s^2}, & x_i, x_j \in X_s \\ \frac{1}{N_t^2}, & x_i, x_j \in X_t \\ \frac{-1}{N_s N_t}, & \text{otherwise.} \end{cases} \tag{2}$$

JDA seeks to learn an empirical approximation of the optimal kernel by utilising a low-rank empirical kernel embedding $\tilde{K} = KWW^{\mathsf{T}}K$ [31]. This allows the distance between the marginal distribution to be rewritten as,

$$D(p(X_s), p(X_t)) = \mathrm{tr}\left(W^{\mathsf{T}}KMKW\right) \tag{3}$$

where $W \in \mathbb{R}^{(N_s+N_t)\times k}$ are the weights which perform a reduction and transformation. Solving an optimisation problem for Eq. (3) is the same as performing transfer component analysis (TCA), which seeks to minimise the marginal distance only [14]. Modifying Eq. (3), the class conditional, along with the marginal distributions, can be considered, leading to an approximation of the distance between the joint distributions formed as,

$$D(p(\boldsymbol{y}_s, X_s), p(\boldsymbol{y}_t, X_t)) \approx \sum_{c=0}^{C} \mathrm{tr}\left(W^{\top} K M_c K W\right) \tag{4}$$

where $c = \{0, 1, \ldots, C\}$, which is the set of class labels in $\mathcal{Y}$, with the addition of zero such that when $c = 0$ the distance is equivalent to Eq. (3). The matrix $M_c$ defines the empirical mean, but now considers the means of the marginal and class conditionals for each class $c$,

$$M_c(i,j) = \begin{cases} \frac{1}{N_s^{(c)} N_s^{(c)}}, & x_i, x_j \in D_s^{(c)} \\ \frac{1}{N_t^{(c)} N_t^{(c)}}, & x_i, x_j \in D_t^{(c)} \\ \frac{-1}{N_s^{(c)} N_t^{(c)}}, & \begin{cases} x_i \in D_s^{(c)} x_j \in D_t^{(c)} \\ x_j \in D_s^{(c)} x_i \in D_t^{(c)} \end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $D_s^{(c)} = \{\boldsymbol{x}_i : \boldsymbol{x}_i \in D_s \wedge y(\boldsymbol{x}_i) = c\}$ are the instances that belong in class $c$ given the true source label $y(\boldsymbol{x}_i)$ of $\boldsymbol{x}_i$ and $D_t^{(c)} = \{\boldsymbol{x}_i : \boldsymbol{x}_i \in D_t \wedge \hat{y}(\boldsymbol{x}_i) = c\}$ are the instances that belong in class $c$ given the pseudo-target label $\hat{y}(\boldsymbol{x}_i)$ of $\boldsymbol{x}_i$ (where $\wedge$ is the logical AND symbol). The numbers of data points in each class for the source and target domains are therefore $N_s^{(c)} = |D_s^{(c)}|$ and $N_t^{(c)} = |D_t^{(c)}|$.

Minimising the approximate distance between the joint distributions can subsequently be formed as the following regularised optimisation problem,

$$\min_{W^{\top} K H K W = \mathbb{I}} = \sum_{c=0}^{C} \mathrm{tr}\left(W^{\top} K M_c K W\right) + \mu \mathrm{tr}\left(W^{\top} W\right) \tag{6}$$

which is subject to the constraint, $W^{\top} K H K W = \mathbb{I}$ (kernel principal component analysis), removing the trivial solution $W = 0$; where $H = \mathbb{I} - 1/(N_s + N_t)\mathbf{1}$ is a centring matrix, $\mathbb{I}$ is an identify matrix and $\mathbf{1}$ a matrix of ones. The regularisation parameter $\mu$ controls a complexity penalty on the weights. A Lagrangian approach can be used to convert equation (6) to an eigenvalue problem for $W$, where the eigenvectors required for the mapping correspond to the $k$ smallest eigenvalues ($\phi$) of,

$$\left(K \sum_{c=0}^{C} M_c K + \mu \mathbb{I}\right) W = K H K W \phi \tag{7}$$

where the $k$-dimensional transformed feature space is calculated by $Z = KW \in \mathbb{R}^{(N_s+N_t)\times k}$. Once the data have been mapped onto this new transfer space, a classifier can be constructed on the source domain data and applied to the target domain data, where the joint distributions are now 'close' (given the minimum MMD distance).

The main complexity in JDA is that the algorithm requires pseudo-labelling of the target features in order to obtain the target domain class conditionals. In the original algorithm, Long et al. recommend using a naïve self-labelling technique and iterating in an EM-like approach to refine the pseudo-label estimates [17]. The method begins by initialising $W$ when $M_c$ is evaluated only for $c = 0$, i.e. the distance between marginal distributions alone is minimised, as this requires no knowledge of the target labels. In this initial transformed space a classifier is trained using the source data and applied to the target data, providing estimates of $\hat{\boldsymbol{y}}_t$. These estimates are used to update $M_c$ for $c = \{0, 1, \ldots, C\}$, and the process of re-estimating $W$ and then $\hat{\boldsymbol{y}}_t$ is iterated. The challenge with this approach is that, if the class conditionals are *very* different, then the initial mapping, based on the marginal distributions, will result in poor estimates of the target labels $\hat{\boldsymbol{y}}_t$. This problem can be difficult to overcome, even with repeat iterations, as the incorrect pseudo-labels will lead to an inappropriate set of weights, compounding the error. As a result, a modification to JDA is introduced later on in Section 5; however, for now, vanilla JDA is applied in the following section in order to demonstrate the applicability of domain adaptation to the problem of repair in SHM.

### 3.2. Domain adaptation on genetic algorithm-selected feature space

In order to overcome the repair problem on the Gnat aircraft wing, domain adaptation methods are applied to the GA-selected feature space. Several domain adaptation methods are compared, those involving minimising statistical distances such as TCA, JDA, and weighted-balanced distribution adaptation (WBDA) [21], – a modification to JDA that seeks to overcome class imbalance – along with a manifold-based approach, the geodesic flow kernel (GFK) [16], as well as adaptation regularisation-based transfer learning (ARTL) which incorporates the adaptation mapping into a least-squares discriminative classifier [18]. These approaches are benchmarked against conventional approaches to data-based SHM, where a classifier is trained on the pre-repair (source) data and applied to the post-repair (target) data. In this case study a $k$-Nearest Neighbour (KNN) classifier is implemented. Although any

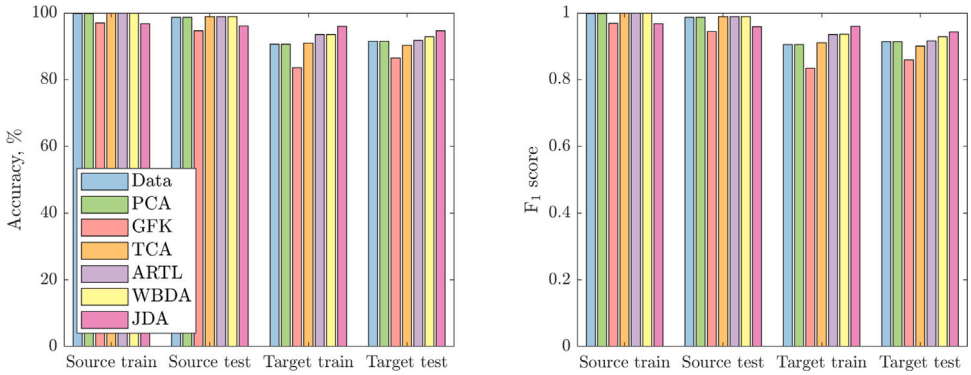


**Fig. 5.** Comparison of classification performance given each feature space considering the nine GA-selected features; left panel, accuracy of predictions, right panel, $F_1$ scores.

classifier can be used in combination with TCA, JDA, WBDA and GFK, a KNN is used here as successful domain adaptation will result in a feature space where the source and target dataset are close in terms of Euclidean distance.

The Gnat dataset is divided into a pre-repair source domain and a post-repair target domain. As the domain adaptation approaches need some unlabelled target domain data, the source and target datasets are divided into mapping training and testing data, i.e. the source and target training data are used in inferring the domain adaptation mapping. The KNN classifiers are only trained on the source training data and are applied to the source testing, target training and target testing datasets as three independent datasets. Each KNN classifier uses a Euclidean distance metric where the number of neighbours have been selected via ten-fold cross-validation on the source training data.

For TCA, WBDA, ARTL and JDA, a radial basis function (RBF) kernel was implemented, moving the data into an RKHS, i.e. $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/2L^2)$ where $L$ is the scale parameter. This kernel function is chosen as it provides a smooth and flexible nonlinear mapping, where the hyperparameter $L$ was determined using a median heuristic on the training data, according to [30]. The regularisation parameter for TCA, WBDA and JDA was $\mu = 0.1$ (the other regularisation parameters in ARTL were both set to 0.1), and as TCA, WBDA, ARTL and JDA all map to a space which is at least $(d-1)$-dimensional, eight components were selected (with the GA-selected features having been designed to be a low-dimensional representation). The domain adaptation techniques are trained using the source and target training data, where the target training data are unlabelled. WBDA, ARTl and JDA were implemented with 10 iterations to allow for convergence. Fig. 5 presents results from applying a KNN to the following feature spaces: the original feature space, a nine-dimensional principal component space, the GFK kernel space, the eight-dimensional TCA transfer component space, the eight-dimensional WBDA transfer component space and the eight-dimensional JDA transfer component space, along with the results from ARTL.

In this scenario, the KNN trained on the original feature space performs well on the source test data (98.7%) which is comparable to the 98.1% obtained on the complete interleaved dataset by the MLP in [24]. However, the performance drops on the target datasets (90.6% in training and 91.6% in testing). This drop in performance is caused by the changes in the data distributions from the source to target data. In comparison, JDA improves performance on the target training and testing datasets to 96.0% and 94.6% respectively. It is noted that JDA aims to find a mapping that reduces the distance between the source and target datasets, and does not have a mechanism for maintaining class separability. As a result, JDA shows a slight reduction in performance on the source datasets, when compared to the KNN trained on the data and PCA feature spaces. What JDA shows in this case study is a more consistent classification performance across each dataset, showing that the feature space is more suitable for learning generalisable behaviours when compared to the data and PCA feature spaces. In addition, TCA, WBDA and ARTl all provide similar performances, maintaining relatively good classification scores in the source domain, with WBDA and ARTL improving classification in the target domain, and TCA providing target classification scores similar to using the original feature space. It is noted that none of these approaches outperform JDA in the target domain. In this case study, GFK has failed to identify a suitable mapping, mainly due to the fact that the subspace embeddings provide little benefit for this low-dimensional original feature space.

These results show the potential for domain adaptation to overcome differences in data distributions caused by structural repairs. The next section considers domain adaptation on the complete transmissibility set, meaning that domain adaptation is more heavily relied upon in performing feature extraction as well as transfer learning. In addition, considering the full transmissibility set removes the issues associated with selecting features given an understanding of damage from both the pre- and post-repair states; something often not obtainable in practical scenarios.

## 4. Overcoming a repair problem on a Gnat aircraft wing: feature extraction and transfer learning

The full transmissibility dataset from the Gnat aircraft is a high-dimensional feature space ($\mathbb{R}^{2500\times9216}$) that experiences domain shift via repeated sequences of measurements of the same damage scenarios. This feature space represents a more typical scenario for repair problems in SHM, and is more challenging for domain adaptation techniques because of greater differences in the joint

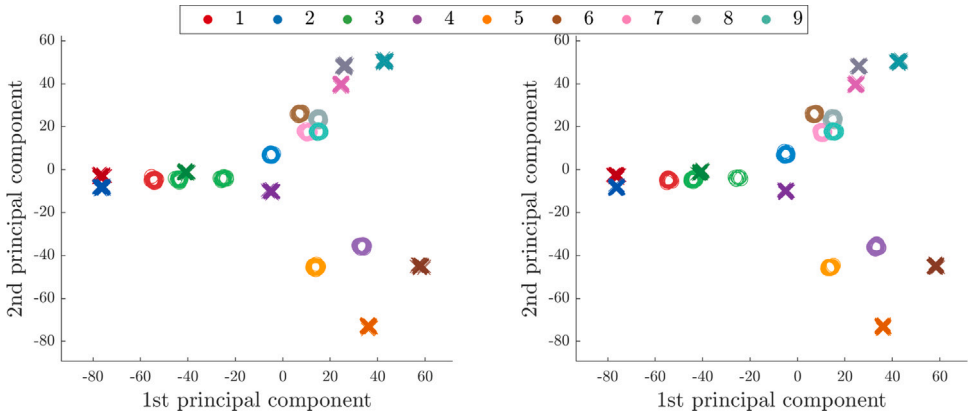


**Fig. 6.** The first two principal components for the complete dataset feature set $\mathcal{X} = \log(h + \{T_i\}_{i=1}^{9})$. Source data (×) and target data (∘) are depicted where the left and right panels show the mapping training and testing data.

distributions between the pre- and post-repair states. This section explores the repair problem when domain adaptation techniques are utilised in performing both a feature extraction role as well as knowledge transfer.

TCA, WBDA, ARTL and JDA are domain adaptation techniques that perform dimensionality reduction as well as knowledge transfer. This ability to find a reduced-dimension shared feature space means that the approaches can be used as a feature extraction technique for the Gnat dataset, removing the problems associated with hindsight in Section 3. The complete dataset is composed from the feature space $\mathcal{X} = \log(h + \{T_i\}_{i=1}^{9})$; here a small offset $h$ is added to the transmissibilities in order to remove the logarithm of zero elements [32]. Once more, the SHM problem is localisation of panel removals, meaning the label space is $\mathcal{Y} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The repair scenario is formed by considering the first repeat from each transducer group as the pre-repair source data, $X_s \in \mathbb{R}^{900\times 9216}$ formed from sequences $\{1, 3, 5\}$ in Table 2, and the post-repair target data, $X_t \in \mathbb{R}^{900\times 9216}$ to be formed from sequences $\{2, 4, 6\}$. The source and target datasets were divided into mapping training and testing datasets composed of 500 and 400 randomly-selected observations respectively.

The first two principal components of the feature data are shown in Fig. 6. It is clear from the PCA space that the complete dataset is particularly challenging for domain adaptation when the target domain is completely unlabelled. Many of the classes switch positions between the source and target datasets; for example Class Six, which relates to the removal of a small panel, not only has a large degree of separation between the source and target datasets, but the target data for Class Six is very similar to those of Classes Seven, Eight and Nine. This picture shows that the conditional distributions are *very* different, and will invalidate the initial assumption in JDA, that matching the marginal distributions will result in good initial pseudo-label estimates.

TCA, WBDA, ARTL and JDA were implemented with $\mu = 0.1$ (the other regularisation parameters in ARTL were both set to 0.1) and an RBF kernel (with the hyperparameters obtained from the median heuristic [30]). The number of transfer components selected was nine. For both domain adaptation techniques, the source and target training datasets were used in learning the mapping, where the target training dataset is unlabelled. For each feature space (apart from ARTL which is itself a classifier) a KNN classifier was trained on the source training data, where the number of neighbours was selected through ten-fold cross validation. Classification performance metrics are presented in Fig. 7. As expected, all the domain adaptation methods have not rectified the problems with the feature space because of the significant differences in the conditional distributions, and in fact, has led to negative transfer [22,33] in certain parts of the feature space. An example of this negative transfer is shown in the JDA mapping in Fig. 8. It is also noted that the data and PCA feature spaces have produced equivalent classification scores, showing that PCA has not improved separability when compared to the original data. Motivated by these difficulties, the next section introduces metric-informed JDA as a method for relaxing the initial pseudo-labelling assumption in JDA.

## 5. Metric-informed joint distribution adaptation

Domain adaptation methods typically assume that the target dataset is unlabelled – hence the need for transfer learning – and this assumption is true of the repair problem (i.e. an unlabelled post-repair dataset). This issue poses challenges in matching the joint distributions between the source and target datasets, i.e. $p(\boldsymbol{y}_s, X_s) \approx p(\boldsymbol{y}_t, X_t)$, as information relating to the target conditional distribution $p(\boldsymbol{y}_t \mid X_t)$ is unknown. To overcome this challenge, some form of semi-supervised learning is required, leveraging both labelled source and unlabelled target data points in training to estimate the predictive function [34]. In this scenario, semi-supervised learning is used to provide *target pseudo-labels* $\hat{\boldsymbol{y}}_t$, which can be used to approximate the target joint distribution, i.e. $p(\hat{\boldsymbol{y}}_t, X_t) \approx p(\boldsymbol{y}_t, X_t)$ if $\hat{y}_i^t \approx y_i^t \;\; \forall i = 1, \ldots, N_t$.

Metric-informed JDA seeks to obtain target pseudo-label estimates utilising a method informed by a normalised distance metric, specifically the Mahalanobis squared distance (MSD). The proposed algorithm builds on joint distribution adaptation (JDA) [17], with an additional metric-assisted step, offering better initial estimates of the target pseudo-labels than naïve self-labelling alone. The following subsections introduce the metric-assisted methodology before outlined the complete algorithm.

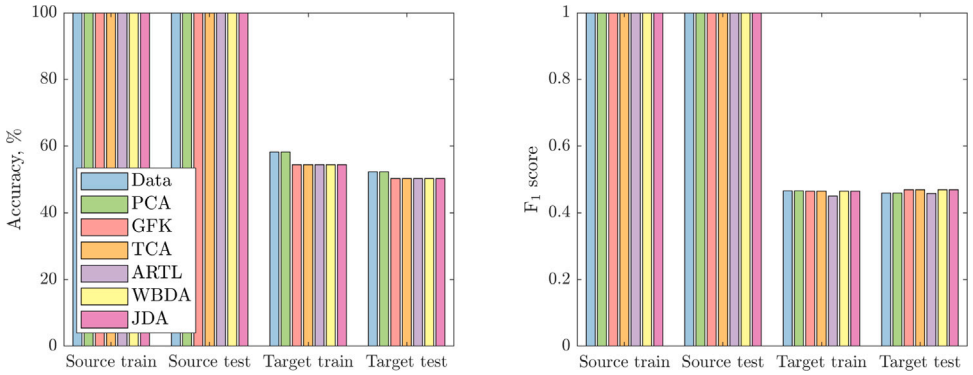


**Fig. 7.** Comparison of classification performance given each feature space, considering the complete dataset; left panel, accuracy of predictions, right panel, $F_1$ scores.

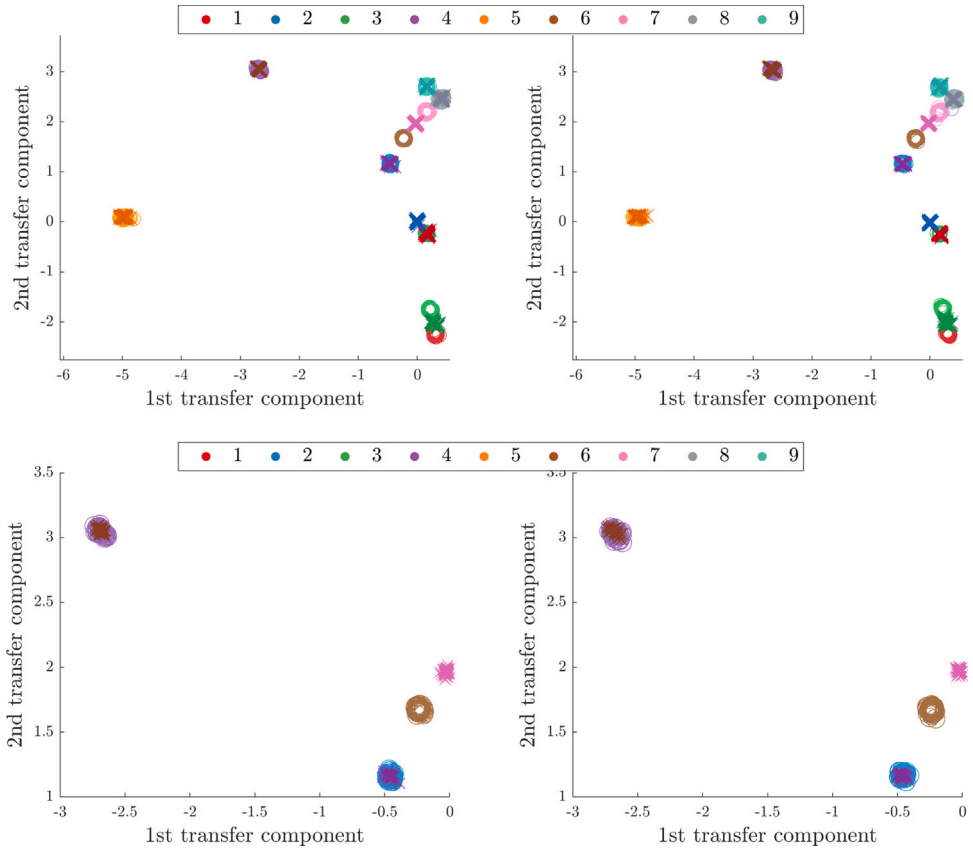


**Fig. 8.** Joint distribution adaptation transfer components for the complete transmissibility set $\mathcal{X} = \log(h + \{T_i\}_{i=1}^{9})$. Source data (×) and target data (○) are depicted where the left and right panels show the mapping training and testing data. Top panels present the complete data space with the bottom panels showing a zoomed region where negative transfer has occurred.

### 5.1. *Metric-informed pseudo-labelling*

As aforementioned, a significant challenge in implementing JDA is obtaining accurate pseudo-labels for the target dataset $\hat{\mathbf{y}}_t$ (such that a mapping can be found that minimises the distance between joint distributions). The method proposed by Long et al. relies on the transformed space, based on the initial marginal distribution matching, to offer an adequate first guess at the pseudo-labels [17] — akin to assuming that initially the conditional distributions between the source and target datasets are equal. This may be limiting in some scenarios, and hence a procedure is introduced here that utilises a metric-informed approach using Mahalanobis squared

distance (MSD) metrics in providing improved initial pseudo-labels to JDA. This metric-informed approach is akin to assuming that for each class the source cluster will be closest to the corresponding target cluster.

In order to encode the assumption that data from the same class will be most similar for the source and target data (i.e. the same classes in the source and target are 'closest'), a distance metric, specifically the Mahalanobis squared distance (MSD) is utilised. The reason for this choice of metric is that the MSD metric has been well-studied within the field of outlier analysis, and is a natural distance metric for performing novelty detection [1,26,27,35]. Parallels can be made between the process of determining if data from one class in the target domain are most similar to their corresponding class in the source domain – the goal of the metric-informed psuedo-labelling step – and novelty detection, where the aim is to determine if new data points are similar to some previously-defined set of normal condition data points. Formally, the Mahalanobis squared distance (MSD) is defined as,

$$D_i^2 = (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^{\mathsf{T}} \hat{\Sigma}^{-1} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}) \tag{8}$$

where $\boldsymbol{x}_i$ is the observation being considered, given a sample mean $\hat{\boldsymbol{\mu}}$ and covariance $\hat{\Sigma}$ determined from $X = \{\boldsymbol{x}_i\}_{i=1}^N$, therefore assessing the squared-Euclidean distance from the sample mean weighted by the sample covariance. In the field of outlier analysis, the MSD is a measure of novelty and assumes that the rest of the data are distributed given some normal distribution, i.e. $X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, which is therefore approximated through the sample mean $\hat{\boldsymbol{\mu}}$ and covariance $\hat{\Sigma}$. A threshold $T$ can be set for an MSD discordancy test such that, given a confidence bound, observations can be determined as inlying or not. A Monte Carlo approach can be used in order to obtain the threshold given the expected statistics of the MSD samples [35] (and is the method used to obtain the threshold for each class in this paper). It is noted that in the proposed metric-informed psuedo-labelling step, although an MSD distance is used (and a threshold considering its expected statistics), this process is not strictly performing outlier analysis in a formal sense, and instead is being used as a guiding metric to find an initialisation on the psuedo-labels for domain adaptation (hence is named metric-informed).

Metric-informed pseudo-labelling seeks to estimate target pseudo-labels by considering MSDs based on each class in the source domain. The central idea is that labelled source data are used to calculate MSDs for each class $D_c^2(\boldsymbol{x}_i^s)\ \forall\ c \in \{1, 2, \dots, C\}$; formed from a sample mean $\hat{\boldsymbol{\mu}}(D_s^{(c)})$ and covariance $\hat{\Sigma}(D_s^{(c)})$ per class. The assumption is that this MSD, when applied to the unlabelled target data $D_c^2(\boldsymbol{x}_i^t)$, will provide the lowest MSD values for target data from the same corresponding class, i.e. it assumes that data from the same class will be most similar for the source and target data. Each class MSD is subsequently normalised by its threshold $\bar{D}_c^2(\boldsymbol{x}_i) = D_c^2(\boldsymbol{x}_i)/T_c$, where all MSD values below $T_c$ are set to zero. This normalisation allows for objective comparison between each class, given class imbalance. A normalised MSD feature space is formed $D_C^2 \in \mathbb{R}^{(N_s+N_t)\times C} = \{\bar{D}_c^2(\boldsymbol{x}_i)\}\ \forall\ c \in \{1, 2, \dots, C\}$ — where there is a normalised MSD vector for each class. The target pseudo-label can then be obtained by setting the label for the $i$th target instance to be the class $c$ corresponding to the minimum normalised MSD, i.e. the $c$ where $D_c^2 = \min(\bar{D}_C^2)$. An overview of the procedure is outlined in the first section of Algorithm 1.

Metric-informed JDA can be divided into two actions: obtain the initial pseudo-labels from the outlier procedure, then perform JDA given these pseudo-label estimates. It is noted that although there are parallels with transfer metric learning [36], the use of the Mahalanobis distance in the proposed method is to create a constraint on the initial target pseudo-label estimates, and not formally used as part of the cost function for learning the feature mapping (as would be the case in transfer metric learning). Algorithm 1 outlines the approach, where the main difference to conventional JDA is that initially the conditional distributions between the source and target data can be more different, as long as the majority of source and target data for each class are more similar to each other than any other class.

### 5.2. Metric-informed JDA and the curse of dimensionality

One problem with performing metric-assisted pseudo-labelling via MSD measures, is that the MSD suffers from the *curse of dimensionality*, meaning if the number of observations in $X$ is small, i.e. $N < (d + 1)$ the sample covariance will be singular [37]. This is the case for the Gnat dataset, where each transmissibility has a size $1 \times 1024$ (number of observations by dimension), and the complete feature space, i.e. all nine transmissibilities $X = \{T1, T2, T3, T4, T5, T6, T7, T8, T9\}$, has a size $1 \times 9216$ (i.e. $d = 9216$), but there are only 100 observations for each class, for each repeat (i.e. $N = 100$). This issue makes applying standard Mahalanobis squared distances challenging, and has led to the idea of outlier ensembles in [28] (i.e. an ensemble of MSD distances). It is proposed that in the case where $N < (d + 1)$ for any particular class, that an ensemble approach is taken (as in [28]) where MSD ensembles are used to construct the MSD feature space $D_C^2$. A brief outline of MSD ensembles approach is provided below, where the steps can be substituted into Algorithm 1.

MSD ensembles use a committee of $M$ models, each modelling the sample mean $\hat{\boldsymbol{\mu}}_m$ and covariance $\hat{\Sigma}_m$, creating a weighted average MSD measure $D_E^2(\boldsymbol{x}_i)$,

$$D_E^2(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} w_m D_m^2(\boldsymbol{x}_i) \tag{9}$$

where $D_m^2(\boldsymbol{x}_i)$ is the $m$th committee member's MSD with respect to the $i$th observation, formed from $\hat{\boldsymbol{\mu}}_m$ and $\hat{\Sigma}_m$, with $w_m$ being its associated weight — set in this paper to unity. Ensembles are formed from a set of $M$ members, with each member being formed from a number of features $N_f$ (where a guiding heuristic is that $N_f < \sqrt{N}$ [28]). For each member, a random set of $N_f$ features, from the $d$-dimensional set, are obtained and used to form the $m$th model's sample mean $\hat{\boldsymbol{\mu}}_m$ and covariance $\hat{\Sigma}_m$, which in turn are used to calculate $D_m^2(\boldsymbol{x}_i)$ for all observations. Once each member's MSD has been obtained, the average is estimated via equation (9). This average MSD can be used in lieu of the conventional MSD of the complete feature in Algorithm 1.

**Algorithm 1** Metric-informed joint distribution adaptation

**Initialise pseudo-labels**

Form dataset $X = X_s \cup X_t$
Initialise $D_C^2 \leftarrow \{N \times C\}$
**for** $c = 1, 2, \ldots, C$ **do**
  $D_s^{(c)} = \{\boldsymbol{x}_i : \boldsymbol{x}_i \in D_s \wedge y(\boldsymbol{x}_i) = c\}\ \forall i \in 1, 2, \ldots, N$ ▷ Get source data for class $c$
  Calculate sample mean $\hat{\boldsymbol{\mu}}_s^{(c)}$ and covariance $\hat{\Sigma}_s^{(c)}$ ▷ Obtain model parameters
  Calculate MSD $D_c^2(X)$ ▷ Calculate MSD for all observations
  Obtain the threshold $T_c$ for a given confidence bound ▷ Using a Monte Carlo approach [35]
  Normalise the MSD $\bar{D}_c^2(X)$
  Store normalised MSD as column in $\bar{D}_C^2(X)$
**end for**
Obtain target pseudo-labels $\hat{y}(X_t) = c$ corresponding to $\min(\bar{D}_C^2(X_t))$

**Domain adaptation**

Set number of iterations
Construct $M_c\ \forall c \in 0 : C$ from equation (5) given $\{y(X_s), \hat{y}(X_t)\}$
**for** $i = 1, 2, \ldots,$ iterations **do**
  Solve eigen-decomposition in equation (7) and select $k$ smallest eigenvectors to construct $W$
  Calculate projection $Z = KW$
  Train classifier $f$ on $Z_s$ and $y(\boldsymbol{x}_i^s)$
  Update $\hat{y}(\boldsymbol{x}_i^t)$ from classifier $f$
  Update $M_c\ \forall c \in 1, 2, \ldots, C$ from equation (5) given $\{y(X_s), \hat{y}(X_t)\}$
**end for**

## 6. Overcoming the repair problem on a Gnat aircraft wing: feature extraction and transfer learning using metric-informed joint distribution adaptation

Metric-informed joint distribution adaptation can be used to relax the initial pseudo-labelling assumptions in JDA. The following sections apply the technique to the transmissibility features for the large panels only, then the complete transmissibility dataset, and finally follow up with the approach on the GA-selected features.

### 6.1. Analysis considering only the large inspection panels

Previous analysis in the literature demonstrated that the smaller panels in the Gnat dataset cause confusion during localisation [23]. In addition, recent analysis in [29] has shown that creating a hierarchy of classifiers that detect whether the class belongs to the large or small panel subsets, and then localising damage within these sets, provides boosted classification performance when compared to the complete classification problem. As a result, the large inspection panels were considered in isolation, i.e. $\mathcal{Y} \in \{1, 2, 4, 5, 7, 8, 9\}$. The feature space was constructed from the log transmissibilities that covered the large panels, i.e. $\mathcal{X} = \log(h + \{T1, T2, T4, T5, T7, T8, T9\})$. The repair scenario considered the first repeat of each transducer group to form the source dataset, i.e. $X_s \in \mathbb{R}^{700\times7168}$ was formed from sequences $\{1, 3, 5\}$ in Table 2, and the target dataset from the second repeat for each transducer group, i.e. $X_t \in \mathbb{R}^{700\times7168}$, formed from sequences $\{2, 4, 6\}$ in Table 2. The source and target datasets are further divided into mapping training and testing sets, with 500 and 200 randomly-selected data points respectively.

A visualisation of the first two principal components is shown in Fig. 9, which highlights the differences between the pre- (source) and post- (target) repair scenarios. Some classes have remained relatively close together, such as Class Four, while others, e.g. Class Two, have moved far enough away that they now lie closer to source data from other classes. It is clear from this figure that a classifier trained on the pre-repair source data will not generalise to the target domain. However, unlike the complete dataset, very few of the classes switch places between the source and target data, and instead show a slight shift in cluster positions.

TCA, WBDA, ARTL, JDA and M-JDA were implemented with $\mu = 0.1$ (the other ARTL regularisation parameters was set to 0.1) and an RBF kernel (with the hyperparameters estimated from the median heuristic [30]). The minimum number of transfer components that provided good classification performance for each domain adaptation method was two, demonstrating a high dimensionality reduction of the space. The domain adaptation techniques were trained using the source and target training datasets, where the target training dataset was unlabelled. Once more, KNN classifiers were trained on each feature space (apart from ARTL which is itself a classifier) using the source training data where the number of neighbours was selected via ten-fold cross validation.

Because of the high dimensionality of the feature space, M-JDA was implemented using an MSD ensemble approach. An ensemble of 1000 members ($M = 1000$) was used for each class, where the number of features used for each member was eight, i.e. $N_f = 8$, selected by the heuristic $N_f \approx \sqrt{N}$ [28], where $N$ is the number of observations in the source training dataset for each class. A

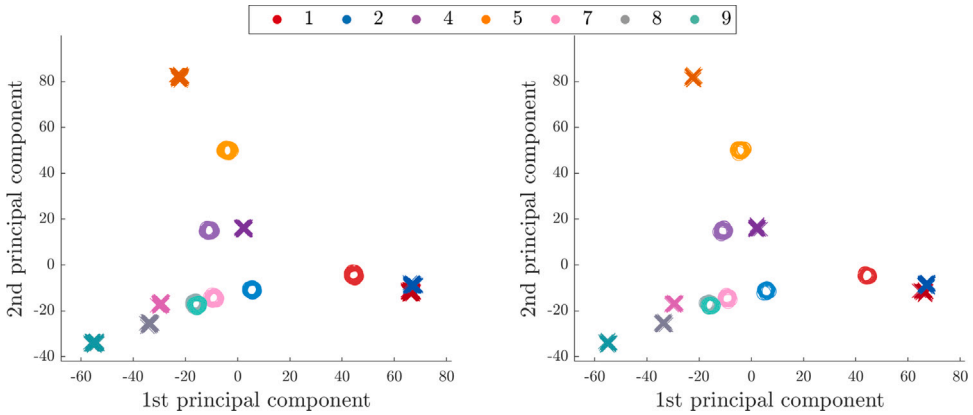


**Fig. 9.** The first two principal components for the large panel feature set $\mathcal{X} = \log\{T_1, T_2, T_4, T_5, T_7, T_8, T_9\}$. Source data (×) and target data (∘) are depicted where the left and right panels show the mapping training and testing data.

threshold for each class was set based on a 99% confidence bound, and was determined using a Monte Carlo approach with 10 000 samples [35]. Examples of the MSDs against their thresholds are presented in Fig. 10 for Classes One, Two, Four and Five. Each ensemble flags the source testing data related to their respective class as below the threshold (and in an outlier analysis setting can be seen as inlying), with the remaining classes having data above the threshold (and in the language of outlier analysis would be identified as outlying). Given the significant changes in the dataset distributions between the source and target datasets, all the target data are above the threshold (and if using the language of outlier analysis would be flagged as outlying), again highlighting the need for transfer. However, for each class ensemble, the spread of classes for the source data are generally similar to the class spread for the target data, i.e. the ordering of classes by 'closeness' would result in a similar order for the source and target datasets. Importantly, for the proposed approach, the closest class for the target dataset in each ensemble corresponds to the class in which the ensemble was constructed, e.g. the ensemble for Class One produces an MSD that means that Class One is the closest class to the threshold for the target dataset. Classes Two and Four are the only classes where this behaviour is not distinct.

In order to obtain pseudo-labels from the unsupervised MSD-ensemble predictions, the normalised MSD feature space is formed as defined in Section 5.1. The pseudo-labels are obtained as the class that corresponds to the minimum normalised MSD for each observation. Two confusion matrices are presented in Fig. 11, showing the accuracy of these pseudo-labels obtained from the MSD procedure on the target datasets. It can be seen that very little confusion has occurred between classes. The metric-informed pseudo-labels were used to initialise M-JDA, where the identified feature space is shown in Fig. 12, and all classes are successfully matched.

The performance of each KNN classifier, and ARTL, are demonstrated in Fig. 13, along with the classification performance of the metric-informed pseudo-labels (MI) — the initial pseudo-labels used in M-JDA. It can be seen that M-JDA has successfully produced 100% accuracy, and significantly outperforms JDA, where negative transfer has occurred due to poor pseudo-label estimates. It is interesting to note that the metric-informed pseudo-labels perform better than TCA, WBDA, GFK and JDA. Vanilla JDA initialises the pseudo-labels by matching the marginal distribution; essentially the pseudo-labels are those from TCA, which in this case provide poor classification accuracy. Because of this poor initial guess for the target labels, JDA performs much worse than M-JDA. Finally, it noted that ARTL has achieved high classification accuracies of 99.6% and 98.5% on the target training and testing datasets. This high performance is because ARTL has a manifold regularisation penality, based on the graph Laplacian, in the mapping (along with minimising the MMD distance). This analysis further evidences that methods like M-JDA and ARTL that consider a nearest neighbour style of approach improve the algorithm's ability to produce positive transfer when class conditionals are very different.

### 6.2. Analysis considering the complete dataset

M-JDA was applied to the complete transmissibility dataset in the same manner as outlined in Section 4, i.e. $\mathcal{X} = \log(h + \{T_i\}_{i=1}^{9})$ and $\mathcal{Y} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Keeping the analysis consistent, M-JDA was applied using an RBF kernel (where the hyperparameters were selected using the median heuristic [30]) and $\mu = 0.1$ with nine components being selected. The source and target training datasets were used to infer the M-JDA mapping, where the target training dataset is unlabelled. A KNN classifier was trained on the source training dataset, where the number of neighbours was selected via ten-fold cross validation. The dimensionality of the complete dataset means that an MSD ensemble approach is required; the ensemble for each class was composed of 1000 members ($M = 1000$), where the number of features used for each member was eight, i.e. $N_f = 8$. The threshold corresponded to a 99% confidence interval was determined using a Monte Carlo approach with 10 000 samples [35].

The classification performance metrics are presented in Fig. 14, where M-JDA and the metric-informed pseudo-labels (MI) have both outperformed the other domain adaptation methods. In this scenario, the data and PCA spaces both produce the same classification scores, showing that PCA has again failed to simplify the classification problem. Another interesting observation is

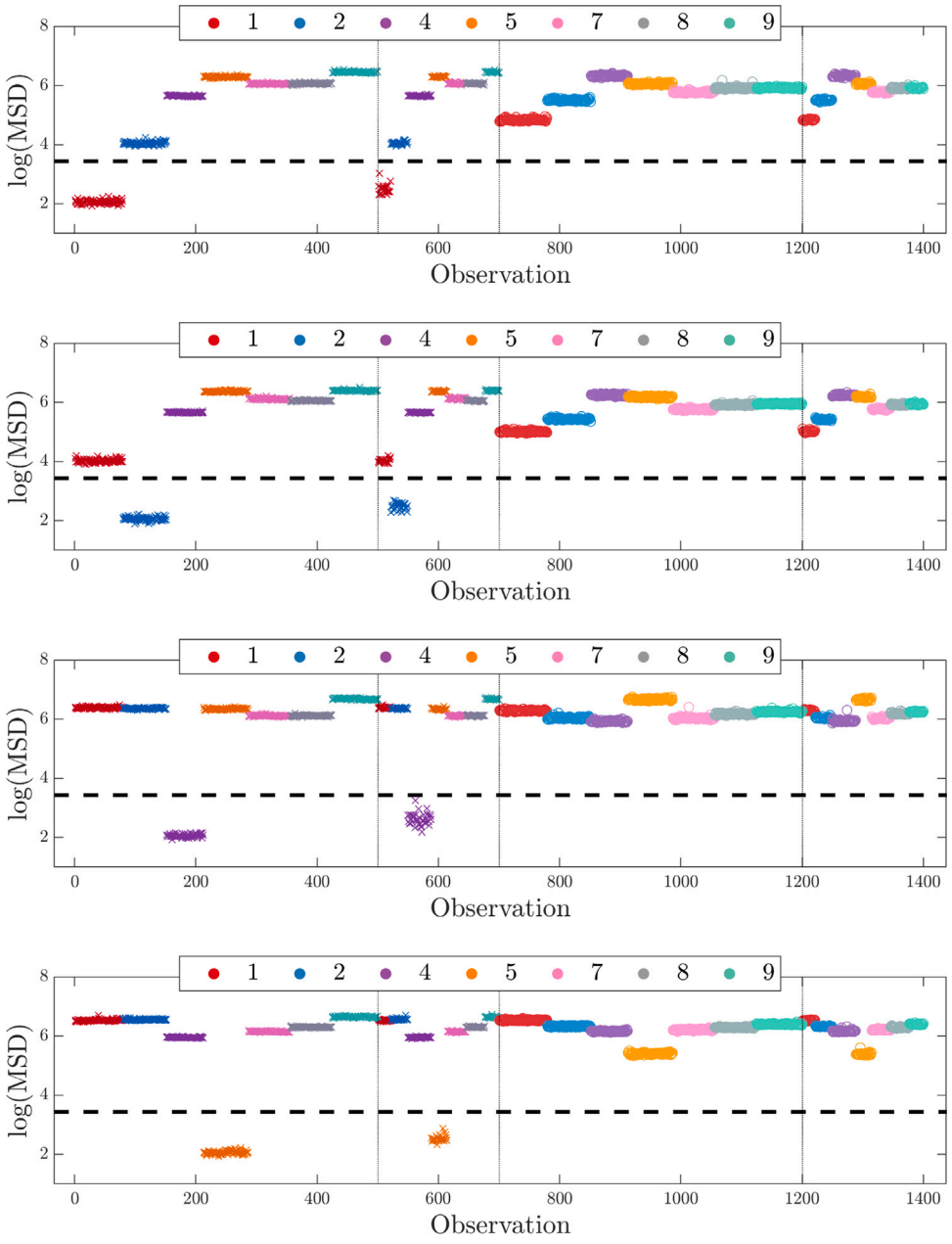


**Fig. 10.** Ensemble Mahalanobis squared distances and their thresholds for the large panel dataset when considering class labels one (top panel), two (top middle panel), four (bottom middle panel) and five (bottom panel). Vertical lines denote the source training, source testing, target training and target testing datasets.

that once more, the metric-informed pseudo-labels outperform TCA, explaining why M-JDA is able to provide higher classification accuracy when compared to JDA. Even though M-JDA has improved classification performance compared to all other methods, negative transfer has occurred, meaning some classes were incorrectly paired between the source and target datasets. The reason for negative transfer can be seen from the PCA feature space in Fig. 6, where the conditional distributions are *very* different, meaning that the source classes are not always most similar to their corresponding target classes, invalidating the M-JDA initial psuedo-labelling assumption. This example is important as it highlights the importance in understanding expected changes in the joint distribution between the source and target datasets, and the need for physics-based knowledge to be inserted into transfer learning methods to prevent negative transfer.

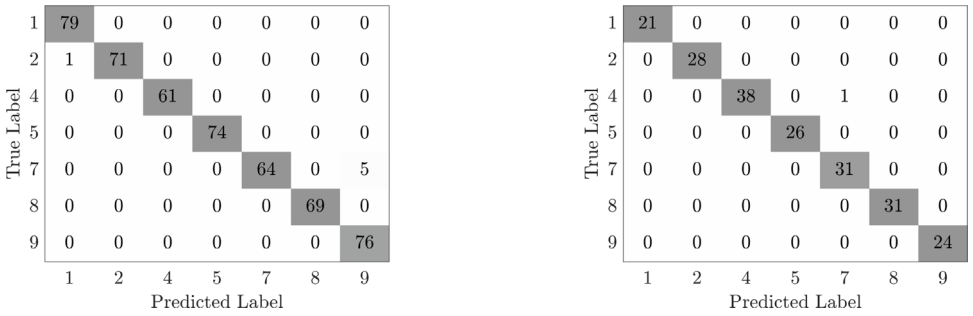


**Fig. 11.** Confusion matrices for the target datasets when considering pseudo-labels from MSD-based analysis for the large panel dataset; mapping training, left panel and mapping testing, right panel.

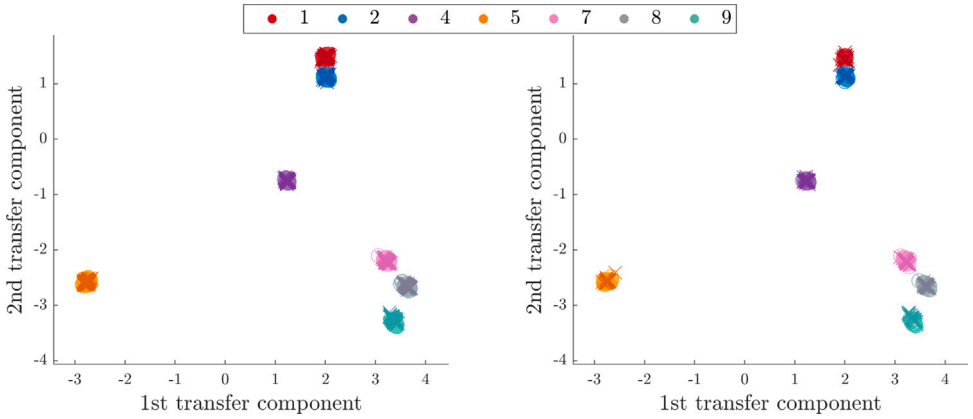


**Fig. 12.** Metric-informed joint distribution adaptation transfer components for the large panel feature set $\mathcal{X} = \log\{T_1, T_2, T_4, T_5, T_7, T_8, T_9\}$. Source data (•) and target data (+) are depicted where the left and right panels show the mapping training and testing data.

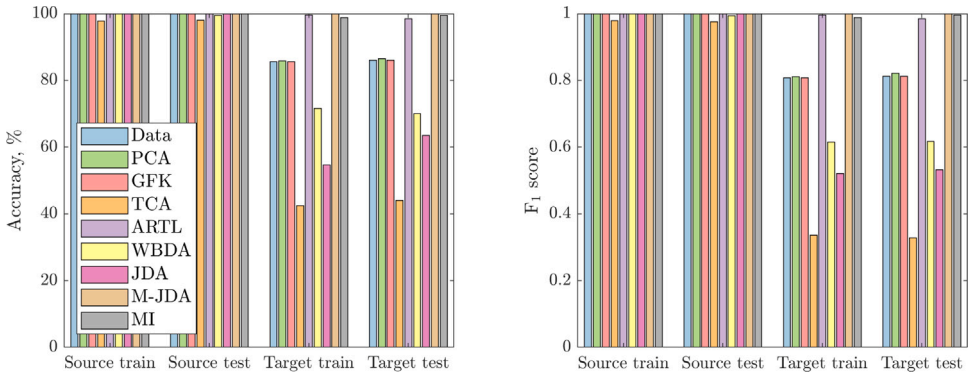


**Fig. 13.** Comparison of classification performance given each feature space, considering the large panel dataset; left panel, accuracy of predictions, right panel, $F_1$ scores. MI refers to the metric-informed pseudo-labels.

### 6.3. Analysis considering the genetic algorithm-selected features

Finally, M-JDA was applied to the engineering knowledge-based features from Section 3, where nine features had been selected from using a genetic algorithm; the label space covered all nine panels. For consistency, M-JDA was applied with an RBF kernel, $\mu = 0.1$, and eight transfer components. The source and target training datasets were used in inferring the mapping, where the target training dataset was unlabelled. A KNN classifier was trained on the source training dataset, where the number of neighbours was selected via ten-fold cross validation. Again, it is noted that any classifier can be used in combination with these domain adaptation techniques. Because of the dimension of the feature space, Mahalanobis squared distance metrics can be performed in order to obtain the metric-informed pseudo labels.

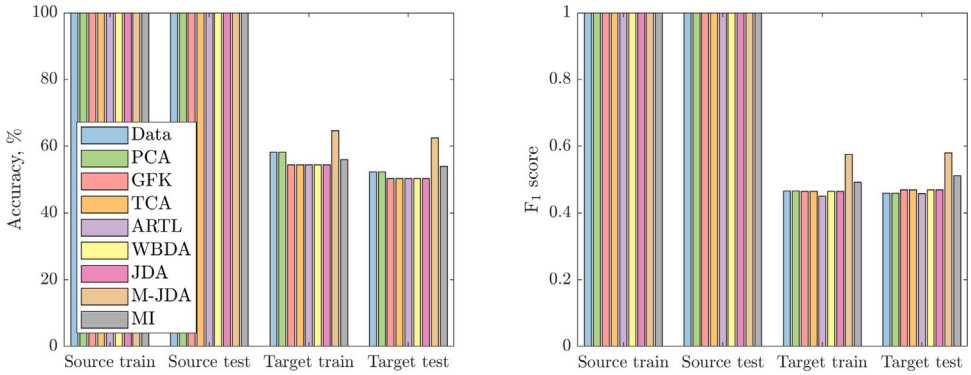


**Fig. 14.** Comparison of classification performance of M-JDA to the other methods, given each feature space, considering the complete dataset; left panel, accuracy of predictions, right panel, $F_1$ scores. MI refers to the metric-informed pseudo-labels.

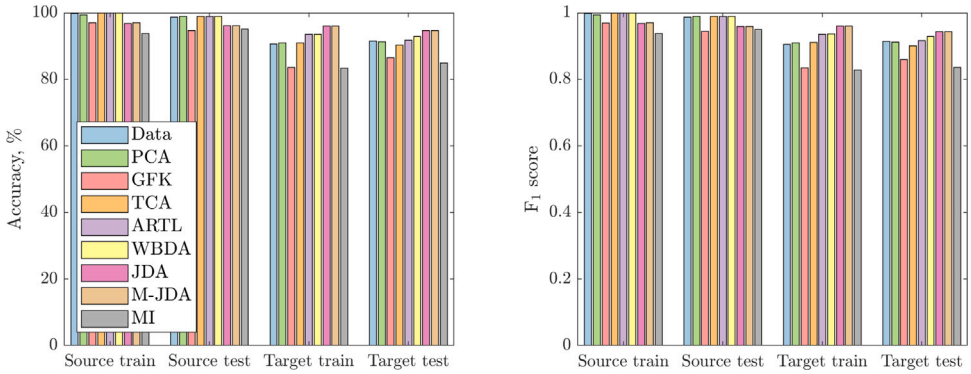


**Fig. 15.** Comparison of classification performance of M-JDA to the other methods, given each feature space, considering the nine GA-selected features; left panel, accuracy of predictions, right panel, $F_1$ scores. MI refers to the metric-informed pseudo-labels.

Classification performance is compared in Fig. 15, along with the metric-informed pseudo-labels (MI) that initialise M-JDA. Both M-JDA and JDA produce similar accuracy and $F_1$ scores. This identical performance occurs due to the feature engineering in the GA-selected nine-dimensional feature space. Each feature in the GA-selected feature space is composed from a novelty detector, meaning that the conditionals between the source and target datasets are similar, reducing the affect of the metric-informed pseudo-labels. Both JDA and M-JDA outperform both the other domain adaptation and non-domain adaptation approaches, showing the benefits of either approach in repair problems in data-based SHM.

In order to demonstrate that any classifier can be combined with these domain adaptation techniques (apart from ARTL, as it is a classifier itself), the methods (including the non-domain adaptation approaches) were rerun on the engineering knowledge-based features from Section 3, with a relevance vector machine (RVM) classifier with an RBF kernel and a Bernoulli likelihood [38]. This nonlinear discriminative classifier is suitable for datasets with a low number of observations, and provides different insights into the classification problem from a KNN, allowing data points that are not necessarily close in Euclidean space to be classified correctly using a more flexible boundary function. The results from this analysis are shown in Fig. 16. Differences in performance, compared to a KNN, are marginal, with the same trends in performance occurring for all methods. For example, M-JDA's predictive performance on the target training and test data increases from 96.0% and 94.6% with a KNN classifier, to 97.4% and 96.4% with an RVM classifier.

These final examples shows that the repair problem can be made significantly less challenging by intelligent feature engineering, especially if the pre- and post-repair observations are engineered such that the joint distributions remain approximately equivalent. However, this will often be challenging to perform for most SHM campaigns without knowledge of damage in the post-repair structure, and as such, domain adaptation methods will be required to perform, not only knowledge transfer, but potentially significant feature extraction as well.

## 7. Conclusions

Structural repairs cause changes to data distributions, leading to domain shift. This shift in the data distribution can lead to a classifier trained on pre-repair data failing to generalise to post-repair data. This shift represents a significant problem in practically applying most machine-learning approaches to SHM. This paper has proposed that domain adaptation techniques can be used to

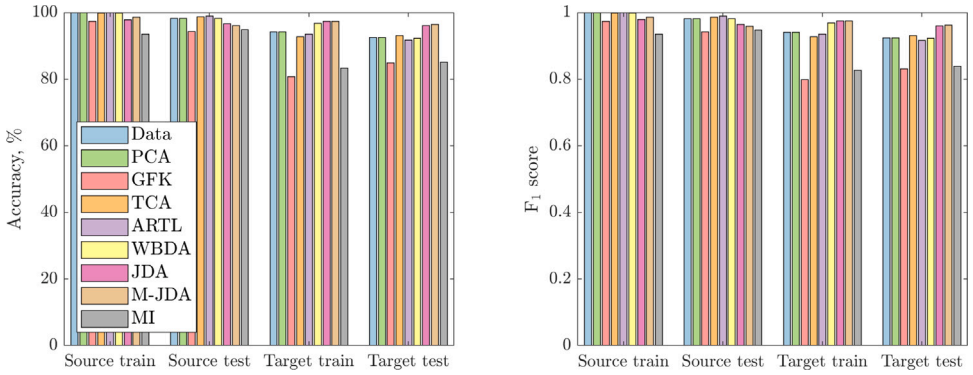


**Fig. 16.** Comparison of classification performance of M-JDA to the other methods, given each feature space, considering the nine GA-selected features and an RVM classifier; left panel, accuracy of predictions, right panel, $F_1$ scores. MI refers to the metric-informed pseudo-labels.

overcome domain shift problems caused by structural repairs, finding a shared latent space where the (source) pre-repair data are mapped onto the (target) post-repair data.

The domain adaptation techniques investigated in this paper were geodesic flow kernel, transfer component analysis, adaptation-regularisation based transfer learning, weighted balanced distribution adaptation, joint distribution adaptation and a novel modification of JDA: *metric-informed joint distribution adaptation*. These methods were benchmarked on feature spaces created from an experimental dataset in which a Gnat trainer aircraft was artificially damaged via the removal of inspection panels. For each case study M-JDA outperformed the other domain adaptation methods as well as conventional approaches to data-based SHM.

Metric-informed joint distribution adaptation utilises Mahalanobis squared distances to create initial pseudo-labels for joint distribution adaptation. The approach assumes that the source domain classes, on the whole, will be closest to their corresponding target domain class. By introducing this assumption, M-JDA has been shown to improve upon conventional JDA in the case studies provided in this paper. However, M-JDA did produce some level of negative transfer when attempting to perform feature extraction and transfer learning on the complete transmissibility dataset. This effect occurred as classes switch relative positions from the source to target domain, meaning that the conditional distributions were *very* different. This is a particularly difficult problem for domain adaptation techniques as the target domain is unlabelled and therefore there is no prior knowledge that classes will switch between training and testing. In the analysis that considered only the large panels, negative transfer did not occur and M-JDA produced 100% classification accuracy, showing the effectiveness of the approach when the metric-informed pseudo-labelling assumption is met. Further research will be required to create generally-applicable domain adaptation approaches for all repair scenarios. These algorithms will likely require physics-based guidance, informing the algorithm which classes are likely to switch locations from the source to target datasets.

Finally, M-JDA (and domain adaptation more generally) has been demonstrated as applicable for overcoming changes in data distributions caused by structural repairs. These results indicate that data-based SHM should have improved industrial applicability, with the potential for label data to be transferred from pre- to post-repair states.

## CRediT authorship contribution statement

**P. Gardner:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Software. **L.A. Bull:** Conceptualization, Writing - review & editing. **N. Dervilis:** Conceptualization, Methodology, Writing - review & editing. **K. Worden:** Conceptualization, Methodology, Funding acquisition, Writing - review & editing, Supervision.

## 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 paper.

## Acknowledgements

The authors would like to acknowledge the support of the UK Engineering and Physical Sciences Research Council via grants EP/R006768/1, EP/R003645/1 and EP/R004900/1.

## References


[1] C.R. Farrar, K. Worden, Structural Health Monitoring: A Machine Learning Perspective, John Wiley & Sons, Ltd, Chichester, UK, 2012.
[2] P. Gardner, X. Liu, K. Worden, On the application of domain adaptation in structural health monitoring, Mech. Syst. Signal Process. 138 (2020) 106550.
[3] P. Gardner, L.A. Bull, J. Gosliga, N. Dervilis, K. Worden, Foundations of population-based structural health monitoring, Part III: Heterogeneous populations, transfer and mapping, Mech. Syst. Signal Process. 149 (2021) 107142.
[4] L.A. Bull, P. Gardner, J. Gosliga, N. Dervilis, E. Papatheou, A.E. Maguire, C. Campos, T.J. Rogers, E.J. Cross, K. Worden, Foundations of population-based structural health monitoring, Part I: Homogeneous populations and forms, Mech. Syst. Signal Process. 148 (2021) 107141.
[5] J. Gosliga, P. Gardner, L.A. Bull, N. Dervilis, K. Worden, Foundations of population-based structural health monitoring, Part II: Heterogeneous populations and structures as graphs, networks, and communities, Mech. Syst. Signal Process. 148 (2021) 107144.
[6] O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, M. Ducoffe, Potential, challenges and future directions for deep learning in prognostics and health management applications, Eng. Appl. Artif. Intell. 92 (2020) 103678.
[7] W. Zhang, G. Peng, C. Li, Y. Chen, Z. Zhang, A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals, Sensors 17 (2) (2017).
[8] X. Li, W. Zhang, Q. Ding, J.-Q. Sun, Multi-layer domain adaptation method for rolling bearing fault diagnosis, Signal Process. 157 (2019) 180–197.
[9] Q. Wang, G. Michau, O. Fink, Domain adaptive transfer learning for fault diagnosis, in: 2019 Prognostics and System Health Management Conference (PHM-Paris), 2019, pp. 279–285.
[10] X. Li, X.-D. Jia, W. Zhang, H. Ma, Z. Luo, X. Li, Intelligent cross-machine fault diagnosis approach with deep auto-encoder and domain adaptation, Neurocomputing 383 (2020) 235–247.
[11] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359.
[12] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (2017) 29.
[13] O. Day, T.M. Khoshgoftaar, A survey on heterogeneous transfer learning, J. Big Data 4 (2017) 29.
[14] 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.
[15] L. Duan, D. Xu, I.W. Tsang, Learning with augmented features for heterogeneous domain adaptation, in: Proceedings of the 29th International Conference on Machine Learning, ICML 2012, 2012.
[16] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2066–2073.
[17] M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, Transfer feature learning with joint distribution adaptation, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 2200–2207.
[18] M. Long, J. Wang, G. Ding, S.J. Pan, P.S. Yu, Adaptation regularization: a general framework for transfer learning, IEEE Trans. Knowl. Data Eng. 26 (2014) 1076–1089.
[19] W. Li, L. Duan, D. Xu, I.W. Tsang, Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 1134–1148.
[20] E. Kodirov, T. Xiang, Z. Fu, S. Gong, Unsupervised domain adaptation for zero-shot learning, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2452–2460.
[21] J. Wang, Y. Chen, S. Hao, W. Feng, Z. Shen, Balanced distribution adaptation for transfer learning, in: 2017 IEEE International Conference on Data Mining (ICDM), 2017, pp. 1129–1134.
[22] Z. Wang, Z. Dai, B. Póczos, J. Carbonell, Characterizing and avoiding negative transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11293–11302.
[23] G. Manson, K. Worden, D. Allman, Experimental validation of a structural health monitoring methodology: Part III. Damage location on an aircraft wing, J. Sound Vib. 259 (2) (2003) 365–385.
[24] K. Worden, G. Manson, G. Hilson, S.G. Pierce, Genetic optimisation of a neural damage locator, J. Sound Vib. 309 (3) (2008) 529–544.
[25] H. Sohn, Effects of environmental and operational variability on structural health monitoring, Phil. Trans. R. Soc. A 365 (2007) 539–560.
[26] K. Worden, G. Manson, D. Allman, Experimental validation of a structural health monitoring methodology: Part I. Novelty detection on a laboratory structure, J. Sound Vib. 259 (2) (2003) 323–343.
[27] G. Manson, K. Worden, D. Allman, Experimental validation of a structural health monitoring methodology: Part II. Novelty detection on a gnat aircraft, J. Sound Vib. 259 (2) (2003) 345–363.
[28] L.A. Bull, K. Worden, R. Fuentes, G. Manson, E.J. Cross, N. Dervilis, Outlier ensembles: A robust method for damage detection and unsupervised feature extraction from high-dimensional data, J. Sound Vib. 453 (2019) 126–150.
[29] G. Tsialiamanis, D.J. Wagg, P. Gardner, N. Dervilis, K. Worden, On partitioning of an SHM problem and parallels with transfer learning, in: Proceedings of IMAC XXXVIII International Conference on Modal Analysis, Houston, USA, 2020.
[30] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schöolkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (2012) 723–773.
[31] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.
[32] S.R. Johnson, G.C. Rausser, Effects of misspecifications of log-linear functions when sample values are zero or negative, Amer. J. Agric. Econ. 53 (1) (1971) 120–124.
[33] M.T. Rosenstein, Z. Marx, L.P. Kaelbling, T.G. Dietterich, To transfer or not to transfer, in: NIPS 2005 Workshop on Transfer Learning, 2005.
[34] O. Chapelle, B. Schölkopf, A. Zien, Semi-Supervised Learning, MIT press, 2006.
[35] K. Worden, G. Manson, N.R.J. Fieller, Damage detection using outlier analysis, J. Sound Vib. 229 (3) (2000) 647–667.
[36] Y. Luo, Y. Wen, L.-Y. Duan, D. Tao, Transfer metric learning: Algorithms, applications and outlooks, 2018, arXiv:1810.03944.
[37] P.J. Rousseeuw, K.V. Driessen, A fast algorithm for the minimum covariance determinant estimator, Technometrics 41 (3) (1999) 212–223.
[38] M.E. Tipping, The relevance vector machine, in: Advances in Neural Information Processing Systems, MIT Press, 2000, pp. 652–658.