Data-Centric Engineering (2024), 5: e4
doi:10.1017/dce.2024.1

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Multitask feature selection within structural datasets

Sarah Bee[1], Jack Poole[1], Keith Worden[1], Nikolaos Dervilis[1] and Lawrence Bull[2]

[1]Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Sheffield, UK
[2]Department of Engineering, University of Cambridge, Cambridge, UK
Corresponding author: S.C. Bee; Email: scbee1@sheffield.ac.uk

Received: 28 July 2023; **Revised:** 19 December 2023; **Accepted:** 06 January 2024

Keywords: feature selection; generalization; multitask learning; sparsity; structural health monitoring

Abstract

Population-based structural health monitoring (PBSHM) systems use data from multiple structures to make inferences of health states. An area of PBSHM that has recently been recognized for potential development is the use of multitask learning (MTL) algorithms that differ from traditional single-task learning. This study presents an application of the MTL approach, *Joint Feature Selection with LASSO*, to provide automatic feature selection. The algorithm is applied to two structural datasets. The first dataset covers a binary classification between the port and starboard side of an aircraft tailplane, for samples from two aircraft of the same model. The second dataset covers normal and damaged conditions for pre- and postrepair of the same aircraft wing. Both case studies demonstrate that the MTL results are interpretable, highlighting features that relate to structural differences by considering the patterns shared *between* tasks. This is opposed to single-task learning, which improved accuracy at the cost of interpretability and selected features, which failed to generalize in previously unobserved experiments.

Impact Statement

Multitask learning (MTL) is known to be beneficial for population-based structural health monitoring (PBSHM). Joint feature selection with LASSO is the beginning of a thread of research into MTL methods that can be applied to PBSHM to select physically meaningful features and to develop techniques for improving damage diagnosis and prognosis.

1. Introduction

A core challenge within the field of data-based structural health monitoring (SHM) is that of selecting meaningful features for damage detection. For example, features that are significantly different can be selected even if their differences are only a result of operational and environmental effects rather than structural differences (Rohrmann et al., 2000; Sohn, 2006). One data-driven approach for analyzing structures is *population-based structural health monitoring* (PBSHM). PBSHM considers data from multiple structures holistically, with the aim of improving performance, in comparison to data from individual structures with their own respective models.

Multitask learning (MTL) is a suite of methods that considers multiple tasks simultaneously, as shown in Figure 1, and this learning approach is relevant for PBSHM. Caruana (1997) developed an early form of multitask learning using a neural network with backpropagation to train tasks simultaneously and improve

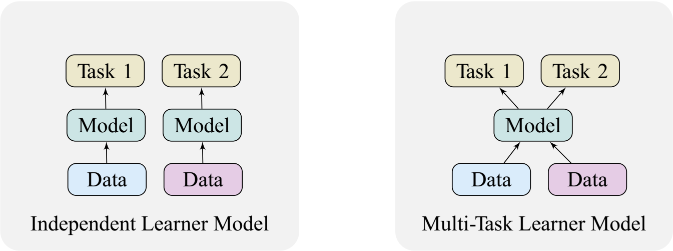


Figure 1. *Comparison of independent and multitask machine learning.*

generalization. One of the principles of MTL algorithms is that, by sharing training data, the performance of the model will improve for all tasks in comparison to each task performed individually. By training multiple tasks together, in the form of multiple structures, the size of the dataset is effectively increased and hence meaningful *feature selection* can occur by considering the patterns that are shared *between* tasks. From the available data, not all feature variations will contain meaningful information about the structure, therefore feature selection is critical. On top of improved generalization, selecting appropriate features can reduce the signal processing requirements on the measured data (Staszewski, 2002) by using a subset of the full measurements. MTL also has the potential to benefit PBSHM, where datasets are often incomplete, because of the cost and safety implications of obtaining the data (Gardner et al., 2022).

This work utilizes MTL for both *automatic feature selection* and classification, while encouraging sparsity in the resulting model. Two case studies demonstrate a sparse solution for both independent learning and MTL, where MTL improves the selection of meaningful features by considering patterns that are consistent between the related tasks. The paper is structured as follows: Section 2 provides an overview of existing research and the contribution of this research to the field; Section 3 discusses the background of the algorithm; Section 4 presents the first case study, which applies the algorithm to different structures from the same aircraft model; Section 5 presents the second case study, which applies the algorithm to a structure pre and postrepair of the same aircraft wing; and Section 6 concludes this paper.

2. Related work

MTL has been utilized in SHM: Wan and Ni (2019), Li et al. (2020, 2021) use MTL models to reconstruct data; Liu et al. (2019) simultaneously detect the location and magnitude of damage on bridges; Dhada et al. (2020) detect anomalies in asset fleets. In the context of modal analysis, Huang et al. (2019) use hierarchical Bayesian models to learn multiple, correlated regression models. Di Francesco et al. (2021) also use hierarchical models to build corrosion models from evidence at multiple locations, and Papadimas and Dodwell (2021) infer model parameters of material constitutive models. This paper focuses on the use of MTL to automate *feature selection* in SHM systems, which is often a manual process. For example, Manson et al. (2003b) and Worden et al. (2003) manually selected features by reviewing transmissibility features against a set of criteria (strong, fair, or weak) and used these as inputs to pattern recognition models. While time-consuming, this process of manually selecting windows from the frequency domain is common throughout the vibration-based monitoring literature. Worden et al. (2008) present work toward automating feature selection, where a genetic algorithm (GA) was used to maximize task prediction during selection. The GA improved classification when compared to earlier studies, but the approach does not necessarily provide features that are *interpretable*, with clear meaning in an engineering context. Mitra et al. (2002) measured the similarity between features, discarding features that provide similar information to the existing set. This approach can provide a sparse set of features; which is also the target of the method present here, to aid the selection of more structurally meaningful features. However, these previous solutions are informed by a *single* task only. As such, this work proposes that MTL can used to help select interpretable features. This is achieved by utilizing *shrinkage* effects, and, most importantly, considering the patterns that are shared between multiple related tasks.

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

Another related and popular method for generating features in SHM is principal component analysis (PCA) (Wang and Ong, 2009; Dackermann et al., 2014; Gordan et al., 2017; Bolourani et al., 2021). PCA is typically used in a single-task setting for dimension reduction by maximizing the variance of a dataset through a linear projection, which is then used to define a subspace. It is widely known that PCA is effective, especially for visualization, though it does not always find a representation that is sensitive to damage. Additionally, PCA is a form feature *extraction* rather than *selection*: it transforms the original features (to generate new ones) rather than selecting a subset of those originally available.

In terms of *multitask* feature extraction, joint domain adaptation (JDA) can be used (Long et al., 2013; Gardner et al., 2021). JDA is a projection technique that minimizes the distance between a source and target distribution in a projected space (rather than maximizing variance, as with PCA). In turn, the datasets can be assumed to have been generated from the same underlying distribution, and a shared model can be learned. JDA is effective and utilized as a benchmark in this work; however, since it is used for feature extraction it transforms the original features, whereas the proposed LASSO approach selects from the original set (while also considering multiple tasks).

2.1. Novelty of work

The model used in this research provides both feature selection and classification in one step, as opposed to separate feature selection and classification models. The model also provides shrinkage of the existing features, which is utilized to improve the interpretability of the results. Unlike other feature selection/extraction techniques, this model uses the original features within the model, rather than transformed features – for example, the features generated in PCA or JDA. Therefore, the work provides a novel implementation of a one-step feature selection and classification model, with the benefit of interpretable results. Most interestingly, the resultant features can be used to generalize to new, previously unobserved experiments.

3. The algorithm

MTL has the benefit of providing improved generalization; in the context of SHM, this means that commonalities between structures can be identified. The assumption is that commonalities identified within an MTL model will relate to similarities between structures, and therefore utilizing MTL as a means of feature selection should provide more interpretable features. The model used here is for binary classification, where all of the original features are presented to the model, and only a subset of relevant features are activated. Hence, this algorithm combines automatic feature selection and classification. The sections below will define the model, how it is solved and how to measure the success of the algorithm.

3.1. The LASSO loss function

For both the independent learner and MTL, logistic regression is used for classification. To mathematically model binary classifications of "True" or "False" a linear regression is used in the form of $\boldsymbol{W}^T\boldsymbol{x}^{(i)}$, followed by an activation function. The Sigmoid function is used in this case, to generate a predicted value between zero and unity:

$$\hat{y}^{(i)} = \frac{1}{1+e^{-\boldsymbol{W}^T\boldsymbol{x}^{(i)}}} \tag{3.1}$$

where $\boldsymbol{x}^{(i)} \in \mathbb{R}^M$ is an observed set of readings of M features, $\boldsymbol{W} \in \mathbb{R}^M$ is the weight vector with a corresponding weight for each of the M features, and the superscript i refers to one of the N observed sets of readings, $i \in \{1,2,\ldots,N\}$.

When the Sigmoid function is used, the resultant value from equation (3.1) will be a value between 0 and 1. The value can be interpreted as a probability that $\hat{y}^{(i)}$ is "True". To set the classification output, a threshold value must be selected; in this work, the typical value of 0.5 is used. The cross-entropy loss function for learning is dependent on the predicted value $\hat{y}$ and the observed value y:

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

$$J\left(y^{(i)},\hat{y}^{(i)};\boldsymbol{W}\right) = -\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)}\,log\left(\hat{y}^{(i)}\right)+\left(1-y^{(i)}\right)log\left(1-\hat{y}^{(i)}\right)\right) \tag{3.2}$$

Equation (3.2) represents the *empirical loss function* for a single task. The empirical loss function is a measurement of error per sample, which is averaged over all N measurement sets. Without regularization, the model is likely to overfit; that is, it will perform well for training data but will not perform well when new data are tested. To reduce the likelihood of overfitting, a further term is added to the loss function. The *Least Absolute Shrinkage and Selection Operator* (LASSO) algorithm (Tibshirani, 1996) adds a regularization term to the empirical loss (3.2) in the form of an ℓ^1 norm. To understand the impact of the ℓ^1 norm on the loss function, it is useful to understand the general form of ℓ^p norms. If $\boldsymbol{W}$ is a vector, $[w_1, w_2, \ldots w_M]$, then the ℓ^P norm is given by,

$$\|\boldsymbol{W}\|_P = \left(\sum_{j=1}^{M} w_j^P\right)^{\frac{1}{P}} \tag{3.3}$$

To visualize the effects of LASSO, consider a simple two-feature optimization; Figure 2 shows three examples of such an optimization. $\hat{W}$ refers to the optimal solution of the weights without any penalty. If implemented, this solution would fit the training data well but is likely to have high variance. Hence, a penalty is included. The ‘constraint surface’ in the three examples is represented by the pink shape centered on the origin. The penalties, for some $c > 0$, are visualized here for comparison as follows:

- ℓ^1 (or LASSO) penalty: $\sum_{j=1}^{M}|w_j| < c$,
- ℓ^2 penalty: $\left(\sum_{j=1}^{M} w_j^2\right)^{\frac{1}{2}} < c$,
- ℓ^∞ penalty: $\left(\sum_{j=1}^{M} w_j^\infty\right)^{\frac{1}{\infty}} < c$.

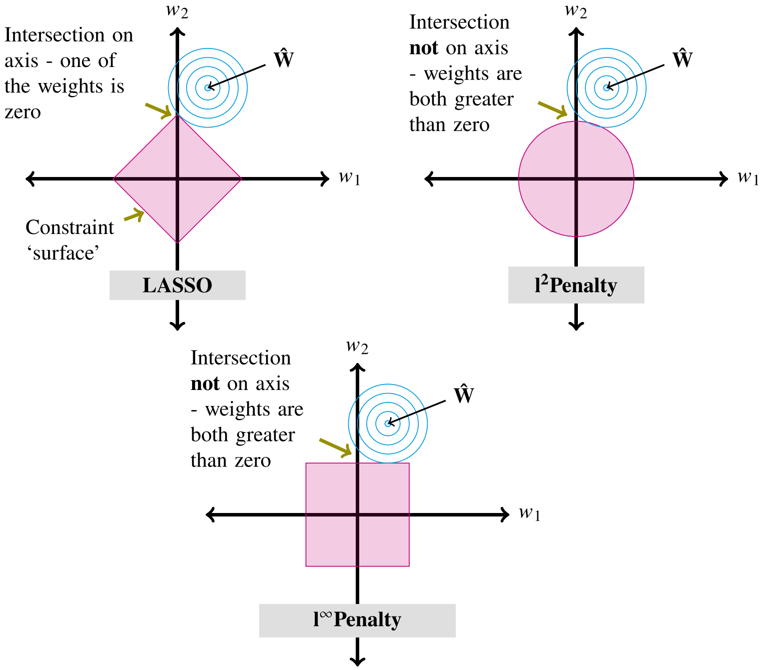

Intersection on axis - one of the weights is zero
w_2
Ŵ
w_1
Constraint ‘surface’
LASSO
Intersection **not** on axis - weights are both greater than zero
w_2
Ŵ
w_1
l²Penalty
Intersection **not** on axis - weights are both greater than zero
w_2
Ŵ
w_1
l^∞Penalty

Figure 2. *A visualization of the shrinkage gained from the LASSO compared to other regularization methods.*

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

The solution to the optimization is the point where the contour of the optimal solution meets the edge of the constraint surface.

For the three examples in Figure 2, the optimal weight is in the same position; the only difference is the shape representing the penalty term. For LASSO, the intersection is on the axis, that is, w_1 is 0, whereas for the other two penalties, the solution is not. To summarize, the visualization demonstrates how LASSO can shrink some of the features to 0, whereas this is less likely with other ℓ^n penalty terms. By implementing ℓ^1 regularization the resulting model is more likely to be representative when given test datasets. However, it should be noted that if the regularization dominates, then under-fitting will occur, and the model will perform poorly for both the training and the test datasets. The total loss function given by the standard independent LASSO algorithm is,

$$\Gamma(y^{(i)},\hat{y}^{(i)};\boldsymbol{W}) = J(y^{(i)},\hat{y}^{(i)};\boldsymbol{W}) + \lambda\|\boldsymbol{W}\|_1 \tag{3.4}$$

where λ is a scalar known as the regularization parameter, and $\|\boldsymbol{W}\|_1 = \sum_{j=1}^{M}|w_j|$ is the ℓ^1 norm of the weight vector.

This total loss function can increase sparsity as it applies the ℓ^1 norm. A large feature set is often less interpretable; therefore, sparsity is a desirable characteristic as it not only offers a reduction in processing requirements but has the additional benefit of better interpretability. In addition, if fewer features are required in a model, this helps to combat the curse of dimensionality (Bellman and Kalaba, 1959), which is often an issue for vibration-based SHM.

3.2. Multitask LASSO

To use the loss function across multiple tasks within one model, the loss function must consider all related tasks. Joint regularization with LASSO was introduced in MTL by Obozinski et al. (2006) to encourage features to share the same sparsity pattern among similar tasks by adding an $\ell^{2,1}$ constraint. The constraint is ℓ^2 across the different tasks, which is then combined into an ℓ^1-norm across the features,

$$\begin{aligned} \|\boldsymbol{W}_L\|_2 &= \left(\sum_{l=1}^{L} w_{j,l}^2\right)^{1/2} && \left(\ell^2\text{across the tasks}\right) \\ \|\boldsymbol{W}_{\boldsymbol{M}}\|_{2,1} &= \sum_{j=1}^{M}\left|\left(\sum_{l=1}^{L} w_{j,l}^2\right)^{1/2}\right| && \left(\ell^1\text{across the features}\right) \end{aligned} \tag{3.5}$$

where L is the number of tasks, and $\boldsymbol{W}_L \in \mathbb{R}^M$.

For each feature, there is an ℓ^2 norm constraint between the tasks. The next layer of constraint is the ℓ^1 norm. For this research, if a feature has a zero weight with this constraint, then all of the tasks will have a zero weight for the given feature; that is, the weight matrix is shared and all of the tasks will share the same sparsity pattern. For multiple tasks the empirical loss function (3.2), becomes,

$$J\left(y_l^{(i)},\hat{y}_l^{(i)};\boldsymbol{W}_l\right) = -\frac{1}{L}\sum_{l=1}^{L}\frac{1}{N_l}\sum_{i=1}^{N_l}\left(y_l^{(i)}\,log\left(\hat{y}_l^{(i)}\right) + \left(1-y_l^{(i)}\right)log\left(1-\hat{y}_l^{(i)}\right)\right) \tag{3.6}$$

where $\boldsymbol{W}_l = [w_{1,l}, w_{2,l}, \ldots w_{M,l}]$ refers to the weight vector, and N_l refers to the number of samples for a given task l. Using the empirical loss function defined above (3.6), the total loss function (3.4), becomes,

$$\Gamma\left(y_l^{(i)},\hat{y}_l^{(i)};\boldsymbol{W}_l\right) = J\left(y^{(i)},\hat{y}^{(i)};\boldsymbol{W}_{\mathbf{1}}\right) + \lambda\|\boldsymbol{W}_M\|_{2,1} \tag{3.7}$$

where $\|\boldsymbol{W}_M\|_{2,1} = \sum_{j=1}^{M}\left|\left(\sum_{l=1}^{L} w_{j,l}^2\right)^{1/2}\right|$.

3.3. Solving with gradient boosting

Further sparsity can be achieved in both independent learning and multitask learning by using *gradient boosting* to solve the algorithm (Obozinski et al., 2006). *Gradient boosting* uses coordinate descent as opposed to gradient descent. Rather than updating all weights simultaneously in one iteration, gradient boosting only updates the weight that provides the largest reduction in the loss. Algorithm 1 details the methodology of gradient boosting for the multitask learner algorithm. As all of the weights are initially zero and only one weight is adjusted per iteration, if a feature has a low influence on the loss function, then it will remain zero, hence encouraging sparse solutions.

Algorithm 1. Multitask boosted LASSO algorithm with a shared weight matrix

```
Set initial parameters: step size, $\varepsilon$ and tolerance, $\xi$
ENSURE: $\varepsilon \geq \xi$.
Determine highest impact weight on the *empirical* loss,
    $\left(\hat{j}, \hat{s}_{\hat{j}}\right) = \underset{j, s_j = \pm\varepsilon}{\arg\min} \; J\left(y_l^{(i)}, \hat{y}_l^{(i)}; s_j \boldsymbol{e}_j\right)$
    ▷$\hat{j}$ is the highest impact weight for the given step size, $s_{\hat{j}}$. $\boldsymbol{e}_j$ is a matrix of 0 s except for a 1 in the $j^{th}$ feature column (for both tasks).
Initialize weight matrix,$\boldsymbol{W}_{\boldsymbol{M}}^0 = \hat{s}_{\hat{j}} \boldsymbol{e}_{\hat{j}}$
Calculate the regularization parameter,
    $\lambda^0 = \frac{J\left(y_l^{(i)}, \hat{y}_l^{(i)}; 0\right) - J\left(y_l^{(i)}, \hat{y}_l^{(i)}; \boldsymbol{W}_{\boldsymbol{M}}^0\right)}{\|\boldsymbol{W}_{\boldsymbol{M}}^0\|_{2,1} - \|0\|_{2,1}}$
$t = 0$
while $\lambda^t \geq 0$ do
    Determine highest impact weight on the *total* loss,
        $\left(\hat{j}, \hat{s}_{\hat{j}}\right) = \underset{j, s_j = \pm\varepsilon}{\arg\min} \, \Gamma\left(y_l^{(i)}, \hat{y}_l^{(i)}; \boldsymbol{W}_{\boldsymbol{M}}^t + s_j \boldsymbol{e}_j, \lambda^t\right)$
    Calculate loss,
        $L = \Gamma\left(y_l^{(i)}, \hat{y}_l^{(i)}; \boldsymbol{W}_{\boldsymbol{M}}^t + s_j \boldsymbol{e}_j, \lambda^t\right) - \Gamma\left(y_l^{(i)}, \hat{y}_l^{(i)}; \boldsymbol{W}_{\boldsymbol{M}}^t, \lambda^t\right)$
    if $L < -\xi$ then
        Update weight matrix, $\boldsymbol{W}_{\boldsymbol{M}^{t+1}} = \boldsymbol{W}_{\boldsymbol{M}}^t + \hat{s}_{\hat{j}} \; \boldsymbol{e}_{\hat{j}}$
        Update regularization parameter, $\lambda^{t+1} = \lambda^t$
    else if $L \geq -\xi$ then
        Determine high impact weight on the *empirical* loss,
            $\left(\hat{j}, \hat{s}_{\hat{j}}\right) = \underset{j, s = \pm\varepsilon}{\arg\min} \; J\left(y_l^{(i)}, \hat{y}_l^{(i)}; \boldsymbol{W}_{\boldsymbol{M}}^t + s_j \boldsymbol{e}_j\right)$
        Update weight matrix,$\boldsymbol{W}_{\boldsymbol{M}}^{t+1} = \boldsymbol{W}_{\boldsymbol{M}}^t + \hat{s}_{\hat{j}} \boldsymbol{e}_{\hat{j}}$
        Calculate regularization parameter,
            $\lambda^{t+1} = \min\left[\lambda^t, \frac{J\left(y_l^{(i)}, \hat{y}_l^{(i)}; \boldsymbol{W}_{\boldsymbol{M}}^t\right) - J\left(y_l^{(i)}, \hat{y}_l^{(i)}; \boldsymbol{W}_{\boldsymbol{M}}^{t+1}\right)}{\|\boldsymbol{W}_{\boldsymbol{M}}^{t+1}\|_{2,1} - \|\boldsymbol{W}_{\boldsymbol{M}}^t\|_{2,1}}\right]$
    end if
    $t = t + 1$
end while
```

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

Algorithm 1 shows the equations that would be used for the MTL approach; however, the methodology is similar in the case of single-task learning models too (utilizing equations (3.2) and (3.4)). To initialize the algorithm, a step size needs to be selected (ε); this is the increment that one of the weights will change by in each iteration, it is a measure of the sensitivity of the resulting weight matrix. In addition, a *tolerance* needs to be specified (ξ), used to determine whether a *forward* or *backward* step will be taken. The condition specified in Zhao and Yu (2004), is that $\varepsilon \geq \xi$. Larger values of these parameters will result in sparser weight matrices. Only the features that have a significant impact on the loss result in updated weights. With reduced sensitivity, the algorithm becomes less accurate, with less iterations, therefore, an increased level of sparsity. With smaller values of the hyperparameters there will be increased sensitivity, more iterations of the algorithm and hence a more accurate (but potentially less sparse) solution.

Following initialization, iterations continue by identifying the weight, which results in the largest reduction in total loss (independent learner: (3.4) and MTL: (3.7), Algorithm 1 line 7). There now exists the original weight matrix and the new weight matrix (with the alteration to the weight, which results in the largest reduction in total loss). The difference between the total loss with the new weight matrix and the total loss with the old matrix represents the loss, L (Algorithm 1 line 8).

A forward step is taken if $L < -\xi$; the weight matrix is updated and there are no changes to the regularization parameter (Algorithm 1, line 10–11). However, if $L \geq -\xi$, then a backward step is taken. For a backward step, the original weight matrix is used and the weight that results in the largest reduction in empirical loss is identified (Algorithm 1, line 13). The weight matrix is updated with the alteration to the weight, which results in the largest reduction in empirical loss (Algorithm 1 line 14). The regularization parameter is also updated (Algorithm 1, line 15), as the minimum value between the original regularization parameter and a new regularization parameter. The iterations of the algorithm continue until the regularization parameter is less than, or equal to, 0. The result of gradient boosting is a sparse solution with weights that do not impact the loss remaining at 0.

3.4. Algorithm performance

Two elements will determine the success of this algorithm: firstly, how close the algorithm is to finding the true labels, and secondly, how sparse the resulting weight matrix is. In this case, we assume sparsity is a good indication of the interpretability of features. Often in machine-learning problems, it is only the accuracy of the algorithm that is assessed; however, a reduction in features can result in increased simplicity and interpretability, as well as reduced processing times, which would be desirable outcomes for SHM applications. Both studies demonstrate how sparse solutions can lead to more meaningful features and better generalization.

While iterating the algorithm, the value of the total loss function is used to assess performance via (3.4) and (3.7); however, this loss is not intuitive when comparing different models against each other. Therefore, to compare independent learning models and MTL models against each other, the F1 score is used. The F1 score is suitable as it is bounded between 0–100% and applies if class sizes are uneven. The F1 score is the harmonic mean of *precision* and *recall*, where precision is the percentage of accurate positive classifications from all of the *predicted* positive classifications and recall is the percentage of accurate positive classifications from all of the *actual* positive classifications.

Hurley and Rickard (2009) provide a comparison of different measures of sparsity; among several methodologies for measuring sparsity, the Gini Index was highlighted as the most robust measure. Originally used to measure wealth distribution by Farris (2010), it has since been applied to various disciplines including ecology (Cordonnier and Kunstler, 2015) and medicine (Bandara et al., 2022). In 2017, the Gini Index was introduced in encoder-based applications to assess the health condition of rotating machinery (Zhao and Lin, 2018). Hurley and Rickard (2009) defines the Gini Index as,

$$G(\boldsymbol{W}) = 1 - 2\sum_{j=1}^{M} \frac{w_j}{\|\boldsymbol{W}\|_1}\left(\frac{M - j + \frac{1}{2}}{M}\right) \tag{3.8}$$

where the $\boldsymbol{W}$ vector used in equation (3.8) has been ordered by magnitude such that $|w_x| < |w_{x+1}|$ for $x \in \{1, \dots, M\}$. The Gini Index will not only consider if a weight has been activated or not, but the magnitude of the weight too; this measure gives a result $G(\boldsymbol{W}) \in [0, 1]$.

3.5. Benchmark tests

To provide a benchmark that relates closely to MTL LASSO, the performance will be analyzed against JDA, which is supervised in both the source and target domains (using the F1 score). JDA aims to find a space by using the kernel-trick and then projecting the data into a space where they become aligned across domains. To provide classification, a two-step process is required: feature extraction, via JDA, followed by a classifier (in this case, k-nearest neighbor (kNN)). JDA does not encourage sparsity, since the linear projection on the kernel is not regularized in the same way. Hence, sparsity will not be assessed for the benchmark.

4. Case study 1: Piper aircraft tailplane

Bull et al. (2021) conducted an experimental campaign on tailplanes, the data were used to determine whether domain adaptation could be used for transfer learning between structures and hence to improve novelty detection. One of the areas identified for future work was automatic feature selection, so *Joint Feature Selection using the LASSO* is explored here to determine if classification accuracy can be improved. The classification task is synonymous with analyzing structures prerepair and postrepair. When a structure undergoes repair, the response of the structure is different from the prerepair condition; two structures that are made to the same manufacturing specifications will be similar but not the same, just as for the pre- and postrepair conditions of the same structure.

4.1. Dataset generation

Two tailplanes labeled A and B from a PA-28 ‘Arrow’ aircraft were used to create the dataset. The tailplane had elevators and wing tips removed and each was cut in half to create a port and a starboard side. Tailplanes A and B have more or less the same geometry (although B was cut asymmetrically). Had the tailplanes been cut symmetrically, then they would be classed as a *homogeneous* population. A population is considered as homogeneous if the structures are *nominally identical* (Gosliga et al., 2022) and there is *structural equivalence* (Gosliga et al., 2021). Thus, tailplanes from A and B are not homogeneous as they were cut asymmetrically and are considered here as *weakly homogeneous.*

The task analyzed as part of this case study is to determine which data are from which tailplane (A or B) using the frequency response function (FRF). This classification is viewed as similar to reviewing a structure in normal condition (A tailplane) and some other condition such as operating, environmental or damage conditions (B tailplane), or vice versa. As there are port and starboard parts of the tailplane, there are two classification tasks.

During the experiments, Gaussian white noise excitation was applied to each of the tailplanes over a frequency bandwidth of 1 kHz with a resolution of 0.3125 Hz; this resulted in 3200 points in the frequency spectrum. The useful range of the FRF was deemed to be between 33.75 Hz and 217.1875 Hz (points 107 to 695). Frequencies lower than this were considered to be influenced too strongly by rigid body movements and higher frequencies were highly influenced by noise. The port and starboard sections each had 180 measurement response points, which were averaged and then normalized across each tailplane, to generate the normalized FRF. The resulting FRF can be seen in the top row of Figure 3.

In the frequency range, there are 588 measured frequencies, each corresponding to one feature. Only one data point exists for each frequency (feature) corresponding to the summed average across all of the 180 response locations. Jain and Waller (1978) proposed that for uncorrelated features, the optimal sample size is $N = M + 1$ (where N is the total number of samples and M is the total number of features), whereas for highly correlated features the optimal sample size is $N = M^2$. This relationship is further backed up by research of Hua et al. (2005). To increase the dataset from 1 to N, a demo dataset was generated using Monte-Carlo sampling, with the mean from the experiments and the variance estimated using the coherence function as in Worden (1998),

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

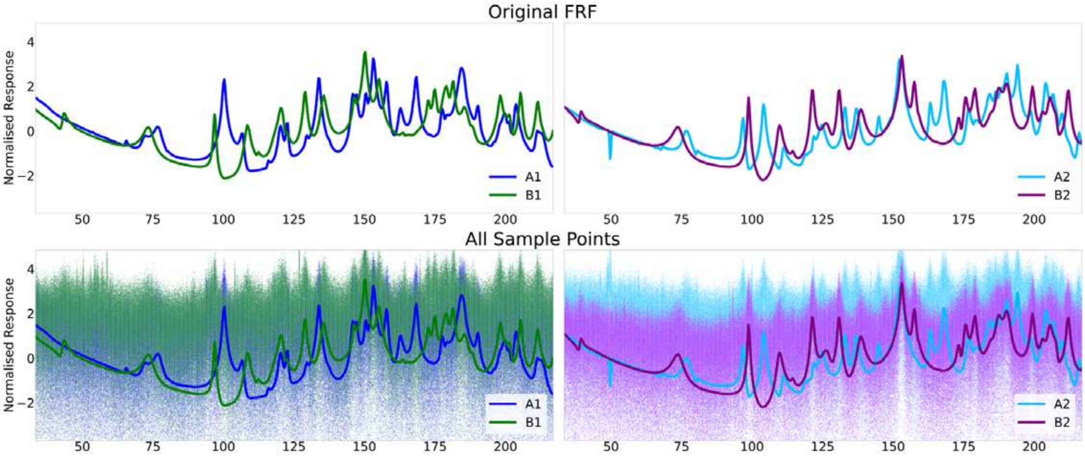


Figure 3. *Port tailplane measurements (Task 1,* left*, blue for A and green for B) and starboard measurements (Task 2,* right*, sky blue for A and purple for B), based on the average response of the 180 measurement points.* Upper: *Mean values only.* Lower: *Mean values and all 750 sample points per structure shown as small translucent points.*

$$\sigma\left(H_p(\omega)\right) = \frac{\sqrt{1-\gamma_p^2(\omega)}}{|\gamma_p(\omega)|\sqrt{2n}} H_p(\omega) \tag{4.1}$$

where $H_p(\omega)$ is the measured frequency response at a given frequency, $\gamma_p(\omega)$ is the standard coherence function and $n = 6$ is the number of values used to compute the FRF (that is, the number of average values used to generate the resulting measured value).

Random sampling (10,000 samples used here) can be used to generate a dataset if the results are assumed to have a Gaussian distribution. As sparsity is encouraged in the algorithm, there is the implicit assumption that correlation between features exists, for the task of distinguishing A from B. The number of meaningful features is unknown at this stage in the statistical model development.

It is hoped that there will be high correlation between the features, so that a sparse solution can be found. A high correlation between tasks would suggest that the number of samples required would be 354,744 (588^2, or 172,872 per class). However, such a high number of samples drastically increases the processing time required by the algorithm. Instead, a sample size of 1500 per task (that is 750 in each class) was generated for training and validation.

To increase the training and validation datasets, fivefold cross-validation was implemented. Fivefold cross-validation is generally accepted as the lower end of the permissible values for k-fold cross-validation (Priddy and Keller, 2005; Mirkin, 2011). As opposed to separate training and validation sets, fivefold cross-validation reduces bias within the model, hence enabling optimal values for the hyperparameters to be selected. A further 500 samples per task (250 per class) were generated for testing purposes. The hyperparameters that need to be selected are the step size, ε, and tolerance, ξ. All three models were trained with the same hyperparameters, $\varepsilon \in \{1, 0.3, 0.1, 0.03\}$, and $\xi \in \{0.1, 0.01, 0.001\}$, and all combinations of ε and ξ were tried subject to $\varepsilon > \xi$.

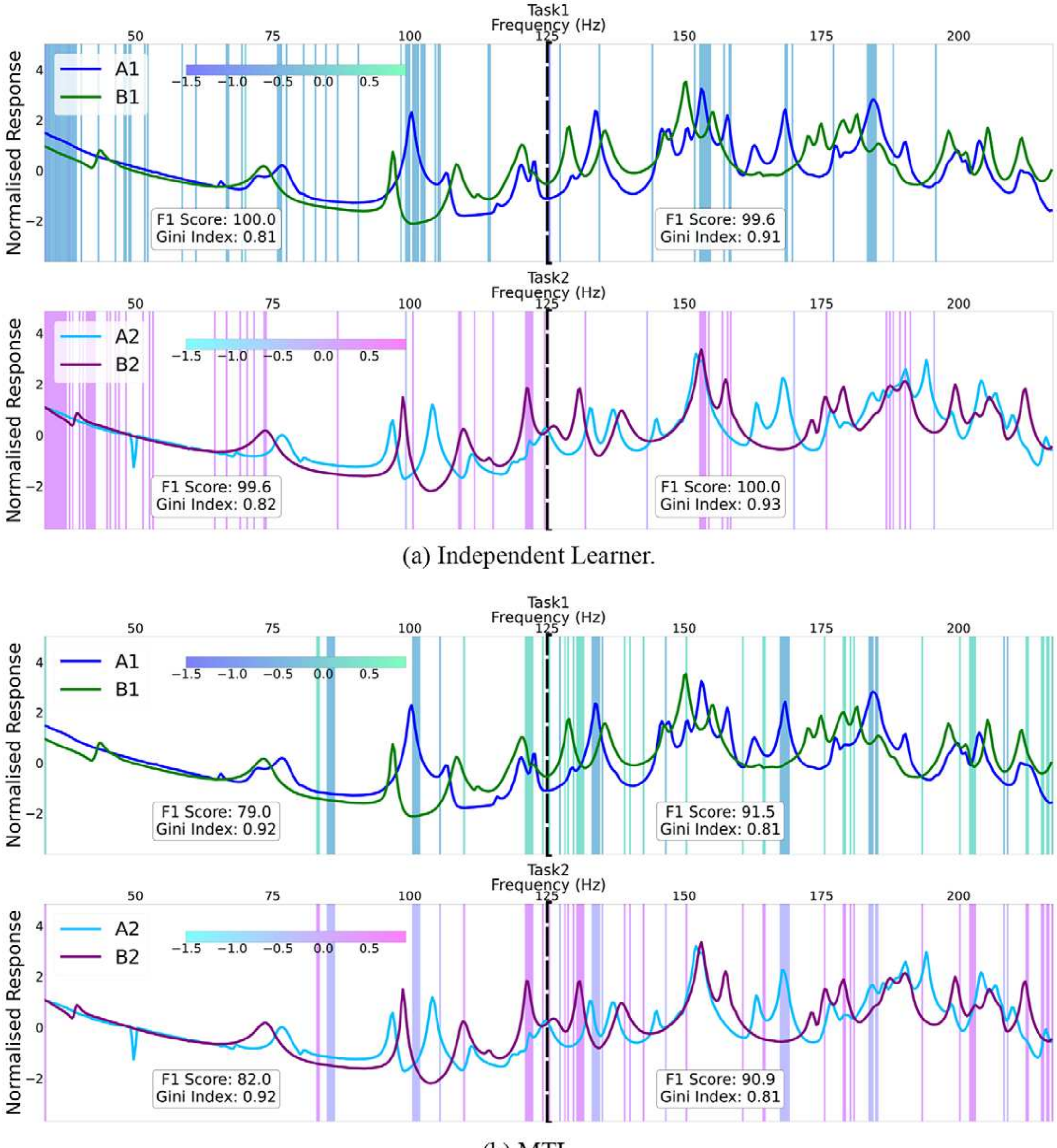
Three models were produced: an independent learner for Task 1, an independent learner for Task 2 and a multitask learner for Tasks 1 and 2. To aid the review of the relative performance of the three models, the FRF is split into 2 windows. The first window contains frequencies less than 125 Hz, it does not contain a lot of information about the structures; however, the second window contains a lot more information about the structures.

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

4.2. Trained model comparison

Following cross-validation, the resulting values are: $\varepsilon = 0.3$ and $\xi = 0.01$ for both the independent and the multitask models. Figure 4 shows the FRF overlaid onto vertical lines, which represent the activated nonzero weights, and therefore the activated frequencies of the solution.

The independent learners (Figure 4(a)) both have activated frequencies at the lower end of the frequencies analyzed. The large number of activated weights at this lower end is not informative of the differences in the structure, but rather of the differences in the rigid body movements created because of



(a) Independent Learner.

(b) MTL.

Figure 4. *FRF and activated weights for test data for both Task 1 (*upper, *Class 1 blue and Class 2 green) and Task 2 (*lower, *Class 1 sky blue and Class 2 purple) for (a) LASSO and (b) Joint Feature Selection with LASSO,* $\varepsilon = 0.3$ *and* $\xi = 0.01$. *The activated weights are shown as vertical lines and the color of the vertical line represents the value of the weight. The dashed black vertical line represents the boundary for the two windows.*

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

the experimental setup. For Window 2 (frequencies > 125 Hz), there appear to be similarities in the activated weights between Task 1 and Task 2. This is to be anticipated, as the samples are from the same aircraft and the motivation for using a multitask learner on this dataset! The benefit of the MTL will become evident in the next section when the model is applied to an unseen task.

Figure 4(a) shows that near-perfect F1 scores are obtained for both tasks and both windows. However, the Gini Index for Window 1 across both tasks is 10% points lower than the Gini Index for Window 2. For Task 1, it could be argued that the feature set from either of the windows could be used; Window 1 has a perfect F1 score; however, there is a marked improvement in Gini Index in Window 2, which is arguably worth the small reduction in F1 score. The selection of the window for Task 2 would be Window 2 as it has a perfect F1 score and a high Gini Index.

For Figure 4(b), the F1 scores are lower than the independent learner equivalent. For Window 1, the frequencies that have been activated are at the higher end of the frequency range, and there is only one frequency activated less than 80 Hz. The absence of low frequencies being activated indicates that the experimental setup (which would be different across all four samples), is no longer useful to differentiate between the two classes. Window 2 outperforms Window 1 for both tasks, with higher F1 scores but a lower Gini Index.

4.3. Transfer of results

To determine the success of the models in finding general features, rather than experiment-specific ones, it is useful to implement the weight matrix (features selected) from the two independent learners and the multitask learner on the two existing tasks and on a third, unseen task. This can be viewed as a form of transfer learning.

In the tailplane dataset, there is a third tailplane, C, which is taken from a PA-28 ‘Cherokee’ aircraft. The structure set is no longer weakly homogeneous, rather it is heterogeneous, as C is a different variant of aircraft. The starboard side of the tailplane (C2) was damaged and the port side of the tailplane (C1) was not damaged. To formulate a task that has similarities with the original task set, a useful classification is to differentiate from the port side of tailplane C and tailplane B. Figure 5 shows the FRF for the two original tasks (*upper* and *middle*) and the third task (*lower*), which has the FRF data from C1.

There are three tasks and there are also three potential weight matrices plus the benchmark JDA; this results in 12 different models in which the F1 score can be analyzed, Figure 6 shows the results for Window 2. Figure 6 shows that the multitask learner is successful in the classification of the third task, to the same accuracy as the two tasks that it was trained on. The independent learner weight matrix for ‘A1 vs

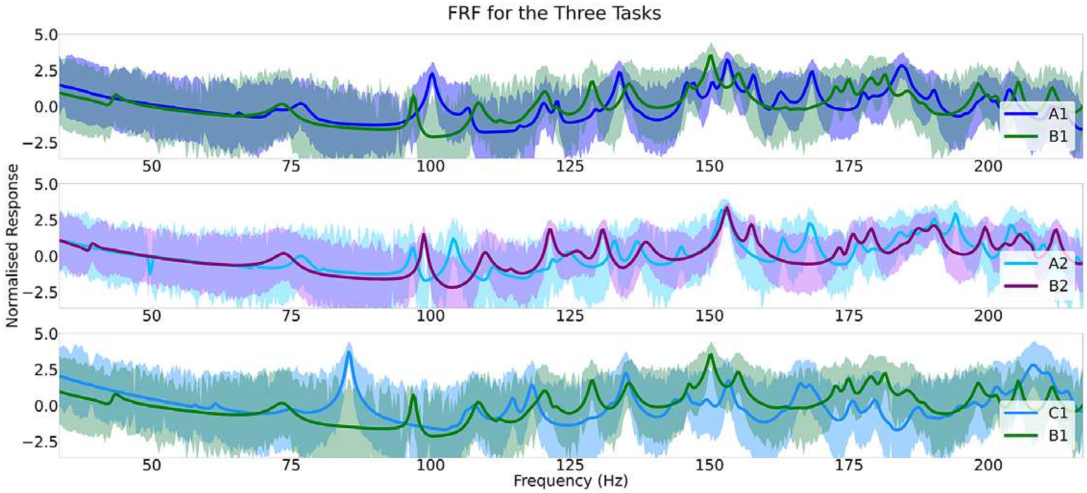


Figure 5. *FRF for the three different tasks showing plus and minus one standard deviation (shaded band).* Upper: *A1 vs B1,* Middle: *A2 vs B2, and* Lower: *C1 vs B1.*

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

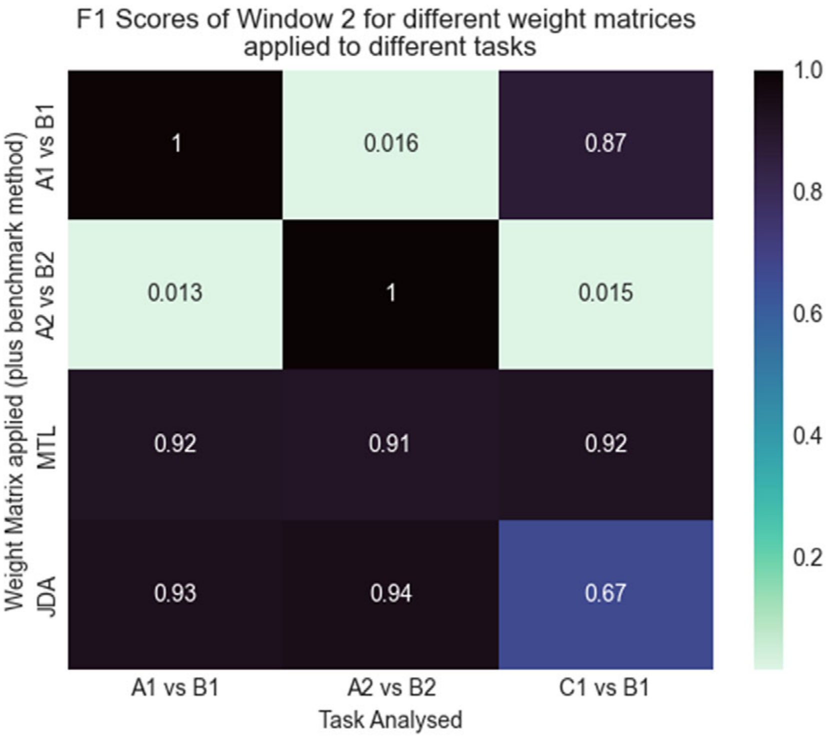


Figure 6. *F1 results for three different weight matrices and JDA (y-axis) applied to three different tasks (x-axis) for Window 2.*

B1' (Task 1) scores perfectly on the task it was trained for, however, there is a drop in the F1 score (to below the performance of the multitask learner) when applied to the unseen third task of 'C1 vs B1'. This is because the independent model learned experiment-specific features, as opposed to features relating to the physics of the structure. As the features were less *structurally meaningful*, the model fails to generalize to the third unseen task.

For Task 1 and Task 2, there appears to be an issue with labeling, such that the classification of Task 1 is perfect with the weight matrix trained on Task 1; however, when the weight matrix for Task 1 is applied to Task 2 there is almost a *perfect misclassification.* When looking at the FRF and weight matrices for comparison (Figure 4), it becomes apparent that, although the same data are used in all models, the features selected for Task 2 are at frequencies that provide the opposite classification for Task 1. This is particularly apparent at around 150 Hz in Figure 4. While JDA outperforms MTL for the tasks that it was trained on (that is, Task 1 and Task 2) for the third *unseen* task, the JDA features result in a decline in kNN performance, demonstrating that the transformation learned by JDA is less general.

In applying the matrices from the three models to a third, unseen task, it is shown that good generalization has occurred in the case of the multitask learner. The multitask learner did not see a reduction in the F1 score when applied to the third task, which demonstrates that the selected features are representative of the physics of the structure as opposed to differences in experimental setup. Positive transfer, to a previously unobserved experimental setup, demonstrates the advantage of more general feature selection by using MTL in SHM.

4.4. Discussion

Successful automatic feature selection has been demonstrated on the tailplane dataset. Gradient boosting yielded results with good sparsity in both independent and MTL settings. The following

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

Model	Window	Features selected
Task 1 independent learner	1	59
	2	27
Task 2 independent learner	1	51
	2	19
Multitask learner	1	21
	2	55

table details the number of features selected out of the 295 potential features for each model and each window:

Classification of the two tasks was near-perfect (that is, 100% F1 score) for the independent learner; however, Window 1 shows that differences in experimental setup was a key driver in feature selection. The features that were consequently selected were specific to the one task, and did not generalize between tasks.

Compared to independent learning, and without any engineering judgment, the result of the MTL is such that the F1 score is lower, as is the Gini Index; therefore, the MTL has not outperformed the independent learner. However, the inference that can be taken from the result of transfer learning is quite powerful. The performance of the MTL on the third unseen task is comparable with the initial two tasks; however, the performance of the third task when using the matrices from the independent learners is poor. This supports the theory that the MTL has selected features that are more meaningful and representative of changes in the structure, rather than the independent learner, which has found differences in the data.

5. Case study 2: GNAT aircraft wing

5.1. The dataset

To further demonstrate how joint feature selection can aid interpretability and general features, a subset of the GNAT dataset used by Manson et al. (2003b) will be tested. Ground-vibration tests were conducted on the wing by applying white Gaussian excitation via an electrodynamic shaker. Figure 7 shows a schematic of the wing; it contained nine panels, which were removed and replaced in a series of experiments to mimic damage followed by maintenance, as described in Manson et al. (2003b).

Each panel has an associated transmissibility, that is, the ratio of the response transducer spectrum with the reference transducer spectrum. The transmissibilities were measured and converted into magnitudes (Manson et al., 2003b). The lower frequencies were deemed to be insensitive to damage (Manson et al., 2003a) and hence the frequency range of the spectrum is 1024 Hz to 2048 Hz, with 1024 spectral lines.

There were 25 iterations of removing and replacing panels, each iteration containing 100 samples. For this work, rather than readings from all 9 transmissibilities, only transmissibilities and damage associated with panel 1 (A1) will be reviewed. There will be 1024 features corresponding to the frequency transmissibility measurements for A1. A lot of samples relate to damage occurring at different panels, and hence the dataset will be reduced to the following classes:

- Normal 1: Prerepair normal condition – The initial normal condition with all plates in place ($\boldsymbol{X} \in \mathbb{R}^{100 \times 1024}$)
- Damage 1: Prerepair Panel 1 removed – Plate P1 removed ($\boldsymbol{X} \in \mathbb{R}^{100 \times 1024}$)
- Normal 2: Postrepair normal condition – The normal condition after plate P1, plate P2 and plate P3 have been sequentially removed and replaced. ($\boldsymbol{X} \in \mathbb{R}^{100 \times 1024}$)
- Damage 2: Postrepair Panel 1 removed – plate P1 removed again after plate P1, plate P2 and plate P3 have been sequentially removed and replaced. ($\boldsymbol{X} \in \mathbb{R}^{100 \times 1024}$)

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

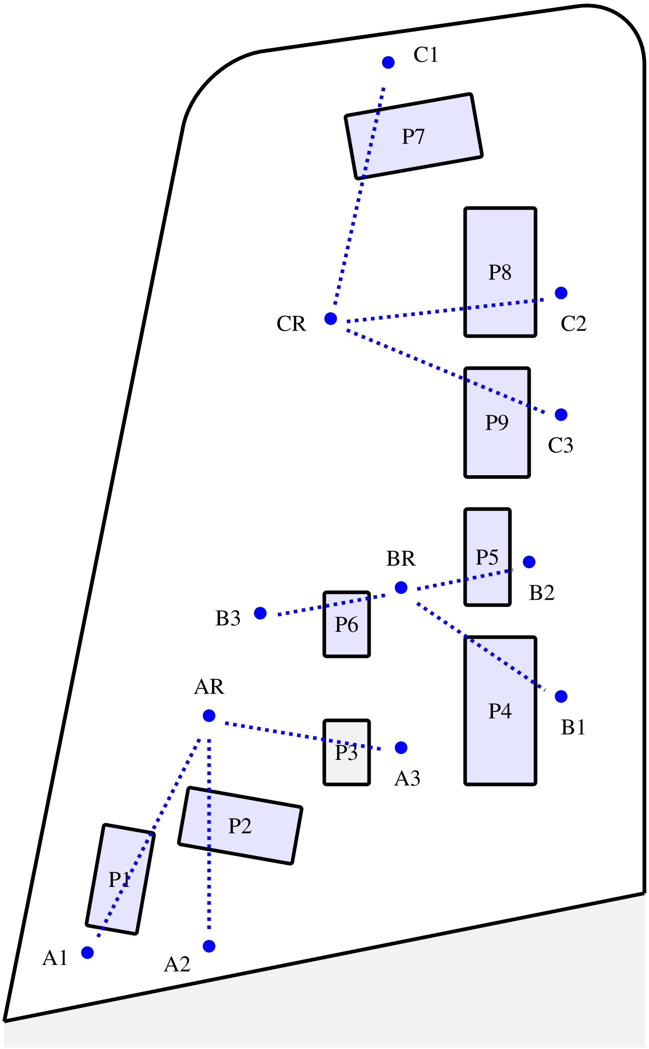


Figure 7. *GNAT plane schematic recreated from Gardner et al. (2021).*

- Normal 3: Post-2nd-repair normal condition – The normal condition after plate P1, plate P2 and plate P3 have been sequentially removed and replaced for a second time. ($\boldsymbol{X} \in \mathbb{R}^{100\times 1024}$)

Initially, the data will be split into two tasks: *Prerepair* and *Postrepair* (omitting the 2nd postrepair class). Each task will have a *normal condition* class and a *damage* class. Figure 8 shows the two tasks (the data undergo a log base-10 transformation). This transformation is useful for FRF analysis, with different magnitudes of responses. As before, for MTL, two hyperparameters need to be selected; the step size ε, and tolerance ξ.

Three models were produced; an independent learner for prerepair, an independent learner for postrepair and a multitask learner for task pre- and postrepair. All three models were trained with the same hyperparameters, the values of the step size, ε, were selected to be approximately three times smaller than the previous value, $\varepsilon \in \{10,3,1,0.3,0.1\}$, and the tolerance was trialed for two different orders of magnitude, $\xi \in \{1,0.1\}$. The value of ξ has less of an impact on the final results than ε, hence less sensitivity analysis for ξ. All combinations of ε and ξ were tried subject to $\varepsilon > \xi$.

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press

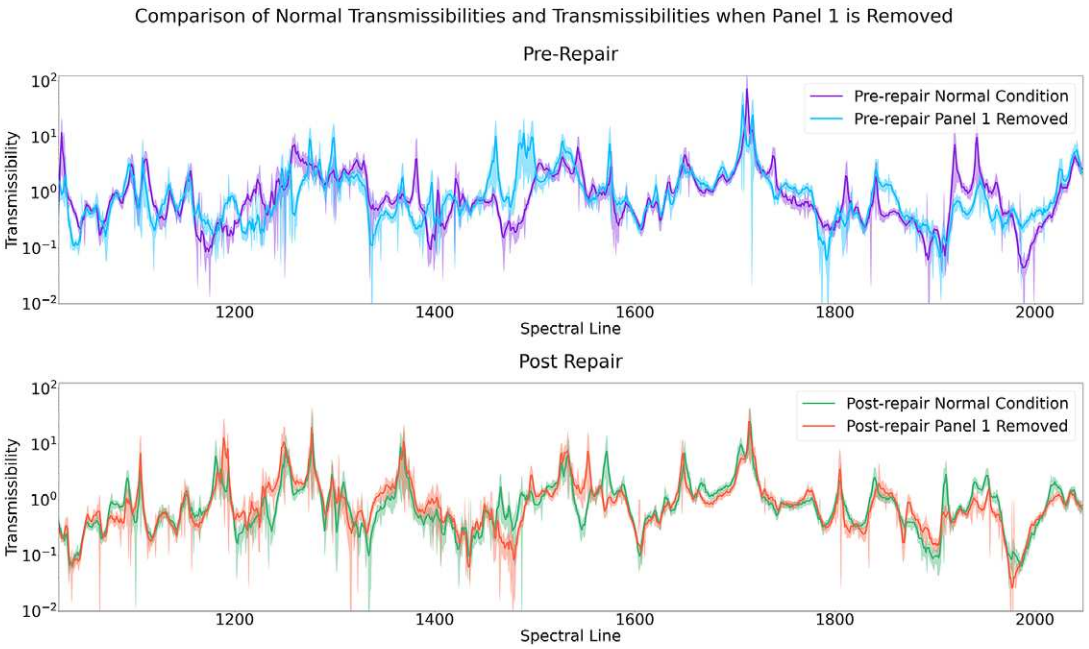


Figure 8. *The transmissibility prerepair (Task 1,* upper*) and postrepair (Task 2,* lower*) of reference transducer AR to response transducer A1 for normal condition and panel 1 removed. Prerepair, Task 1, has Class 1 as normal condition (purple) and Class 2 as Panel 1 removed (sky blue) and postrepair, Task 2 has Class 1 as normal condition (green) and Class 2 as Panel 1 removed (orange). One standard deviation of banding is shown for each class.*

5.2. Trained model comparison

Following tuning of the hyperparameters, the results are: $\varepsilon = 0.3$ and $\xi = 0.1$ for both the independent and the multitask models. Figure 9 shows the transmissibilities overlaid onto vertical lines, which represent the activated weights and therefore the activated frequencies of the solution. The prerepair independent learner (Figure 9(a), *top*), has the lowest F1 score of the models and also the highest Gini Index. The majority of the weights activated are within a band at around spectral line 1500 and there is another band of activated frequencies at around spectral line 1280. For the independent learner for postrepair (Figure 9(a), *bottom*) there are more spectral lines that have been selected and there appears to be less grouping of the activated spectral lines.

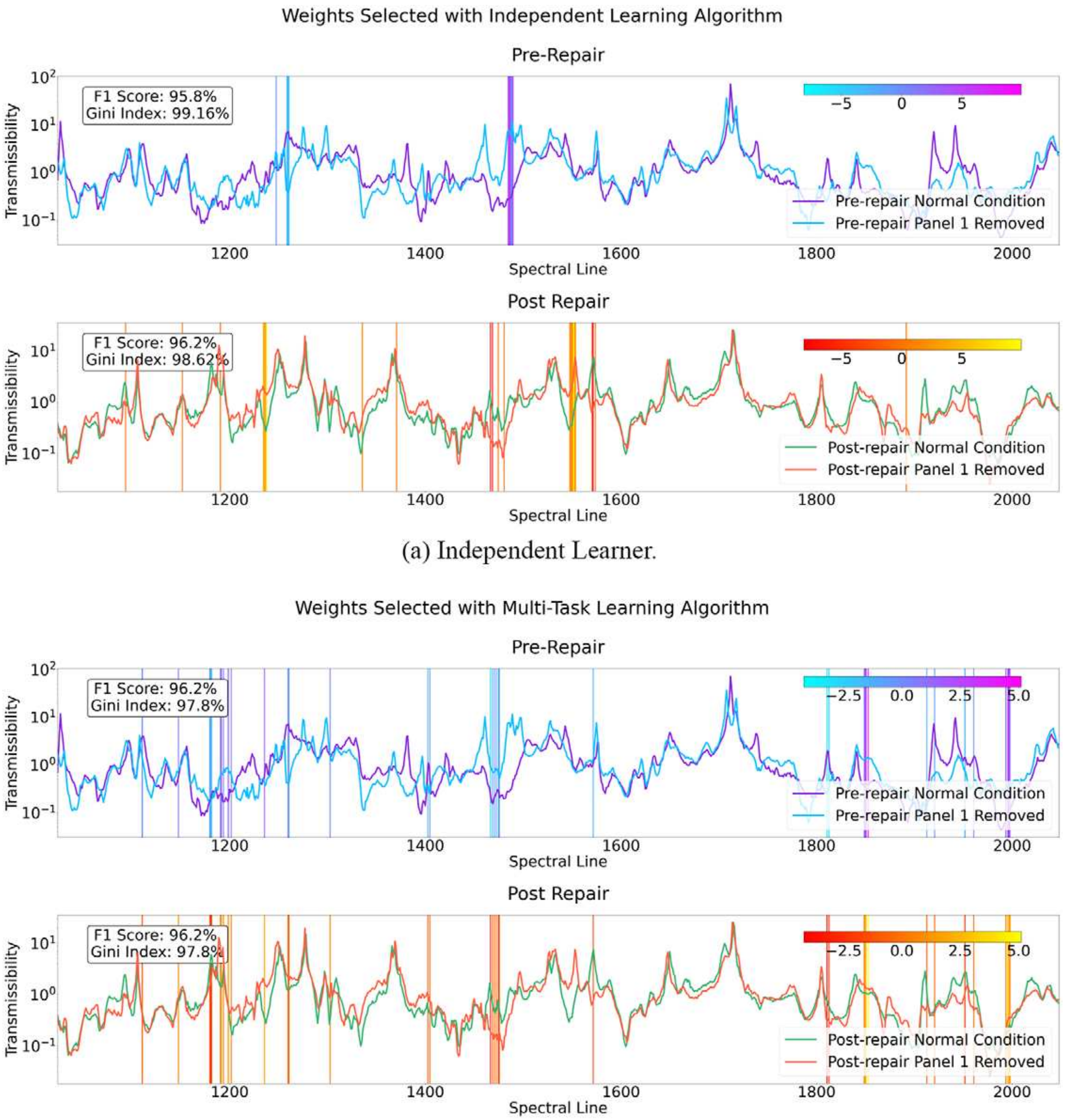
Figure 9(b) shows that the MTL has improved F1 scores for the prerepair task over the independent equivalent. However, as with the previous example dataset, the Gini Index is lower for the MTL than either of the two independent learners.

5.3. Domain adaptation as a result of the models

To understand how the multitask model aids the discovery of a general feature space, it is useful to visualize the PCA subspaces of the full features. In this paper *domain adaptation* refers to multiplying the original data by the weights that have been calculated in the corresponding models (either weights from the single-task learning models or weights from MTL). Figure 10 shows the four classes, following PCA transformation for three different scenarios.

Figure 10(a) shows the normalized data without any domain adaptation. There are four distinct clusters that are present. The clusters that are closest to each other are the *Postrepair Normal Condition* and the *Postrepair Panel 1* Removed. The classification boundary between normal condition and panel removal is not linear.

https://doi.org/10.1017/dce.2024.1 Published online by Cambridge University Press



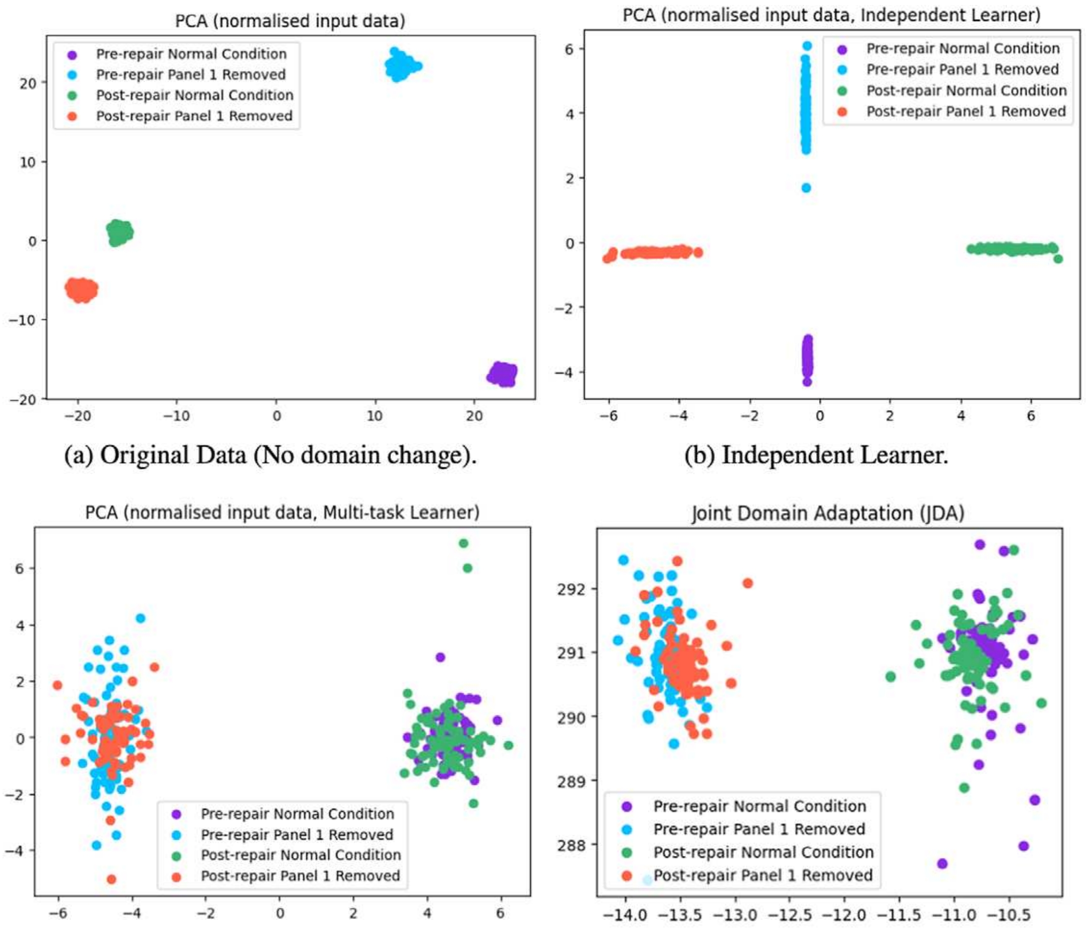
(a) Independent Learner.

(b) MTL.

Figure 9. *Transmissibilities and activated weights for test data for both Task 1 (*upper*, normal condition purple and panel 1 removed blue) and Task 2 (*lower*, normal condition green and panel 1 removed orange) for (a) LASSO and (b) Joint Feature Selection with LASSO,* $\varepsilon = 0.1$ *and* $\xi = 0.1$. *The activated weights are shown as vertical lines and the color of the vertical line represents the value of the weight.*

The impact of the independent learning models on the clusters is shown in Figure 10(b). Alike clusters are closer together; note that the scale on (b) is different to the scale on (a). In addition, the clusters appear evenly spaced from one another, with each class having its own distinct cluster, as in (a).

Figure 10(c) shows how MTL has adapted the domains such that the two classes classified as *Normal Condition* overlap and the two classes classified as *Panel 1 Removed* overlap as well. This implies that the similarities between the two *Normal Condition* classes have been identified by the MTL, as have the similarities between the two *Panel 1 Removed* classes. In effect, the datasets and tasks have been



(a) Original Data (No domain change).

(b) Independent Learner.

(c) MTL.

(d) JDA.

Figure 10. *PCA of the four classes: prerepair normal condition (purple), prerepair Panel 1 removed (blue), postrepair normal condition (green) and postrepair Panel 1 removed (red). (a) PCA of normalized data. (b) PCA after data is multiplied by the weights and bias for the two independent learners (on the corresponding task), concatenated, and then normalized. (c) PCA after data has been multiplied by the weights and bias of the MTL, for both tasks, and then normalized. (d) Shared domain created with JDA.*

harmonized. A linear classification boundary can be readily applied to differentiate between the normal condition and panel removal, unlike the original data and single-task learning. In addition, the F1 score of the MTL shows that the model can accurately classify between the two different classes; this is a promising result, as MTL intends to learn similarities between the tasks for each class and then uses these similarities to select features that can accurately perform the binary classification. Figure 10(d) is included to show how MTL achieves similar domain adaptation to JDA, while also learning the classification task at the same time, and selecting from the original feature set, rather than transforming them.

5.4. Discussion

The results of the independent learner vs MTL for the GNAT dataset demonstrate successful automatic feature selection. Grouping is not encouraged in the algorithm; however, the results from the independent learner for prerepair (Figure 9(a), *top*) show that the activated weights occur, roughly, within two bands. This selection of features is intuitive: similar to the features that would be selected manually, where selections would lead to bands of spectral lines, which contain information about the structures across the multiple classes. In contrast, the independent learner for postrepair and the MTL do not have meaningful groupings of activated weights; instead, there are several places along the spectrum that have individual activated weights/spectral lines – which are harder to interpret in an engineering context.

The F1 scores of the MTL are on par with the scores of the independent learners. From visual inspection of Figure 9, the weights activated around spectral line 1500 provide opposite classification in the independent learner for prerepair, when compared to both postrepair and the MTL; this is the same phenomena previously seen in the tailplane (Figure 6).

A useful insight from this analysis is the effective domain adaptation. Previous work by Bull et al. (2021) used domain adaptation to provide transfer learning between *prerepair and postrepair* conditions. Figure 10(d) shows that JDA can separate the data into clusters for the separate classes. The PCA domain of the MTL shown in Figure 10(c) demonstrates that by learning the two tasks together, the domains for the two *Normal Condition* classes and the two *Panel 1 Removed* classes overlap, respectively. This demonstrates that MTL can identify shared features, which map similar classes onto one another (when applying domain adaptation) while providing good classification results. The additional benefit of the MTL is that the features selected are the original features (unlike the transformed features in JDA) and hence the MTL remains more interpretable.

6. Conclusion

This work has demonstrated how multitask boosted LASSO can select meaningful features for engineering datasets, by considering the *shared* patterns between multiple related tasks. The first study illustrated that MTL LASSO will select meaningful underlying features, as opposed to features that capture experiment-specific differences, but no structural significance. The improvement in the performance when transferring feature knowledge to a previously unobserved task shows that MTL LASSO can be applicable to transfer learning. Successful knowledge transfer also highlights the ability to find more general/meaningful features, by utilizing patterns shared *across* multiple tasks. MTL LASSO improved generalization and interpretability when compared to the benchmark JDA and kNN approach.

The second study showed how MTL selected features that allowed the domains (data) of two tasks to be mapped onto each other, further highlighting the ability to learn general representations, which can be shared between systems and/or structures. MTL LASSO matches the performance of the JDA benchmark, with the benefit of learning both the feature selection and classifier in the same algorithm. Additionally, MTL LASSO naturally provides shrinkage, which was utilized to provide interpretable features in the context of vibration-based monitoring.

Data availability statement. The code used to recreate the results can be found in: https://github.com/Sarah-Bee/MTL_Feature_Selection.

Acknowledgments. The authors wish to gratefully acknowledge Marcus Haywood-Alexander, Evangelos Papatheou and Robin Mills for the curation of the tailplane dataset, and, Graeme Manson and David Allman for the GNAT dataset.

Author contribution. Conceptualization: S.C.B, K.W., N.D., L.A.B.; Funding acquisition: K.W., N.D., L.A.B.; Investigation: S.C.B; Methodology: S.C.B, J.P, K.W., N.D., L.A.B.; Project administration, K.W., N.D., L.A.B.; Software: S.C.B; Supervision, K.W., N.D., L.A.B.; Validation: S.C.B; Visualization: S.C.B; Writing—original draft: S.C.B; Writing—review and editing: S.C.B, J.P, K.W., N.D., L.A.B.

Funding statement. This research was supported by grants from the Engineering and Physical Sciences Research Council (EPSRC), UK, and Natural Environment Research Council, UK via grant number, EP/S023763/1 and EP/W005816/1 - Revolutionising Operational Safety and Economy for High-value Infrastructure using Population-based SHM (ROSEHIPS). L.A. Bull was supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant EP/W006022/1, particularly the *Ecosystems of Digital Twins* theme within that grant and The Alan Turing Institute.

Competing interest. The authors declare none.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

Bandara M, **Gurunayaka B**, **Lakraj G**, **Pallewatte A**, **Siribaddana S and Wansapura J** (2022) Ultrasound based radiomics features of chronic kidney disease. *Academic Radiology 29*(2), 229–235. https://doi.org/10.1016/J.ACRA.2021.01.006.

Bellman R and Kalaba R (1959) On adaptive control processes. *IRE Transactions on Automatic Control 4*(2), 1–9.

Bolourani A, **Bitaraf M and Tak AN** (2021) Structural health monitoring of harbor caissons using support vector machine and principal component analysis. *Structure 33*, 4501–4513. https://doi.org/10.1016/j.istruc.2021.07.032.

Bull L, **Gardner P**, **Dervilis N**, **Papatheou E**, **Haywood-Alexander M**, **Mills R and Worden K** (2021) On the transfer of damage detectors between structures: An experimental case study. *Journal of Sound and Vibration 501*, 116072. https://doi.org/10.1016/J.JSV.2021.116072.

Caruana R (1997) Multitask learning. *Machine Learning 28*(1), 41–75. https://doi.org/10.1023/A:1007379606734.

Cordonnier T and Kunstler G (2015) The Gini index brings asymmetric competition to light. *Perspectives in Plant Ecology, Evolution and Systematics 17*(2), 107–115. https://doi.org/10.1016/J.PPEES.2015.01.001.

Dackermann U, **Smith W and Randall R** (2014) Damage identification based on response-only measurements using cepstrum analysis and artificial neural networks. *Structural Health Monitoring 13*(4), 430–444. https://doi.org/10.1177/1475921714542890.

Dhada M, **Girolami M and Parlikad AK** (2020) Anomaly detection in a fleet of industrial assets with hierarchical statistical modeling. *Data-Centric Engineering 1*, e21. https://doi.org/10.1017/dce.2020.19.

Di Francesco D, **Chryssanthopoulos M**, **Faber MH and Bharadwaj U** (2021) Decision-theoretic inspection planning using imperfect and incomplete data. *Data-Centric Engineering 2*, e18. https://doi.org/10.1017/dce.2021.18.

Farris F (2010) The Gini index and measures of inequality. *American Mathematical Monthly 117*(10), 851–864. https://doi.org/10.4169/000298910X523344.

Gardner P, **Bull LA**, **Dervilis N and Worden K** (2021) Overcoming the problem of repair in structural health monitoring: Metric-informed transfer learning. *Journal of Sound and Vibration 510*, 116245. https://doi.org/10.1016/J.JSV.2021.116245.

Gardner P, **Bull L**, **Gosliga J**, **Poole J**, **Dervilis N and Worden K** (2022) A population-based SHM methodology for heterogeneous structures: Transferring damage localisation knowledge between different aircraft wings. *Mechanical Systems and Signal Processing 172*, 108918. https://doi.org/10.1016/j.ymssp.2022.108918.

Gordan M, **Ismail Z**, **Razak H and Ibrahim Z** (2017) Vibration-based structural damage identification using data mining. In *24th International Congress on Sound and Vibration*. Auburn, AL, USA: International Institute of Acoustics and Vibration (IIAV).

Gosliga J, **Gardner PA**, **Bull LA**, **Dervilis N and Worden K** (2021) Foundations of population-based SHM, part II: Heterogeneous populations – Graphs, networks, and communities. *Mechanical Systems and Signal Processing 148*, 107144. https://doi.org/10.1016/J.YMSSP.2020.107144.

Gosliga J, **Hester D**, **Worden K and Bunce A** (2022) On population-based structural health monitoring for bridges. *Mechanical Systems and Signal Processing 173*, 108919. https://doi.org/10.1016/J.YMSSP.2022.108919.

Hua J, **Xiong Z**, **Lowey J**, **Suh E and Dougherty E** (2005) Optimal number of features as a function of sample size for various classification rules. *Bioinformatics 21*(8), 1509–1515. https://doi.org/10.1093/BIOINFORMATICS/BTI171.

Huang Y, **Beck JL and Li H** (2019) Multitask sparse Bayesian learning with applications in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering 34*(9), 732–754. https://doi.org/10.1111/MICE.12408.

Hurley N and Rickard S (2009) Comparing measures of sparsity. *IEEE Transactions on Information Theory 55*(10), 4723–4741. Available at https://arxiv.org/pdf/0811.4706.pdf.

Jain A and Waller W (1978) On the optimal number of features in the classification of multivariate Gaussian data. *Pattern Recognition 10*(5–6), 365–374. https://doi.org/10.1016/0031-3203(78)90008-0.

Li Y, **Bao T**, **Chen Z**, **Gao Z**, **Shu X and Zhang K** (2021) A missing sensor measurement data reconstruction framework powered by multi-task gaussian process regression for dam structural health monitoring systems. *Measurement 186*, 110085. https://doi.org/10.1016/j.measurement.2021.110085.

Li L, **Liu H**, **Zhou H and Zhang C** (2020) Missing data estimation method for time series data in structure health monitoring systems by probability principal component analysis. *Advances in Engineering Software 149*, 102901. https://doi.org/10.1016/j.advengsoft.2020.102901.

Liu J, **Bergés M**, **Bielak J**, **Garrett JH**, **Kovačević J and Noh HY** (2019) A damage localization and quantification algorithm for indirect structural health monitoring of bridges using multi-task learning. *AIP Conference Proceedings 2102*(1), 090003.

Long M, **Wang J**, **Ding G**, **Sun J and Yu PS** (2013) Transfer feature learning with joint distribution adaptation. In *2013 IEEE International Conference on Computer Vision*. New York City, United States: IEEE, pp. 2200–2207. https://doi.org/10.1109/ICCV.2013.274.

Manson G, **Worden K and Allman D** (2003a) Experimental validation of a structural health monitoring methodology: Part II. Novelty detection on a Gnat aircraft. *Journal of Sound and Vibration 259*(2), 345–363. https://doi.org/10.1006/jsvi.2002.5167.

Manson G, **Worden K and Allman D** (2003b) Experimental validation of a structural health monitoring methodology: Part III. Damage location on an aircraft wing. *Journal of Sound and Vibration 259*(2), 365–385. https://doi.org/10.1006/jsvi.2002.5169.

Mirkin B (2011) *Core Concepts in Data Analysis: Summarization, Correlation and Visualization*. London: Springer. https://doi.org/10.1007/978-0-85729-287-2.

Mitra P, **Murthy C and Pal SK** (2002) Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*(3), 301–312.

Obozinski G, **Taskar B and Jordan M** (2006) *Multi-Task Feature Selection (Tech. Rep. No. Jul 2006)*. Berkeley: University of California.

Papadimas N and Dodwell T (2021) A hierarchical Bayesian approach for calibration of stochastic material models. *Data-Centric Engineering 2*, e20. https://doi.org/10.1017/dce.2021.20.

Priddy K and Keller P (2005) *Artificial Neural Networks: An Introduction*. Bellingham, Washington USA: SPIE—The International Society for Optical Engineering.

Rohrmann RG, **Baessler M**, **Said S**, **Schmid W and Ruecker WF** (2000) Structural causes of temperature affected modal data of civil structures obtained by long time monitoring. In *SPIE Proceedings Series*. Bellingham, WA, US: International Society for Optical Engineering, pp. 1–7.

Sohn H (2006) Effects of environmental and operational variability on structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 365*, 539–560. https://royalsocietypublishing.org/doi/10.1098/rsta.2006.1935.

Staszewski W (2002) Intelligent signal processing for damage detection in composite materials. *Composites Science and Technology 62*(7–8), 941–950. https://doi.org/10.1016/S0266-3538(02)00008-8.

Tibshirani R (1996) Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological) 58*(1), 267–288. https://doi.org/10.1111/J.2517-6161.1996.TB02080.X.

Wan H-P and Ni Y-Q (2019) Bayesian multi-task learning methodology for reconstruction of structural health monitoring data. *Structural Health Monitoring 18*(4), 1282–1309. https://doi.org/10.1177/1475921718794953.

Wang Z and Ong K (2009) Multivariate statistical approach to structural damage detection. *Journal of Engineering Mechanics 136* (1), 12–22. https://doi.org/10.1061/(ASCE)0733-9399(2010)136:1(12.

Worden K (1998) Confidence bounds for frequency response functions from time series models. *Mechanical Systems and Signal Processing 12*(4), 559–569. https://doi.org/10.1006/MSSP.1998.0156.

Worden K, **Manson G and Allman D** (2003) Experimental validation of a structural health monitoring methodology: Part I. Novelty detection on a laboratory structure. *Journal of Sound and Vibration 259*(2), 323–343. https://doi.org/10.1006/JSVI.2002.5168.

Worden K, **Manson G**, **Hilson G and Pierce S** (2008) Genetic optimisation of a neural damage locator. *Journal of Sound and Vibration 309*(3–5), 529–544. https://doi.org/10.1016/J.JSV.2007.07.035.

Zhao M and Lin J (2018) Health assessment of rotating machinery using a rotary encoder. *IEEE Transactions on Industrial Electronics 65*(3), 2548–2556. https://doi.org/10.1109/TIE.2017.2739689.

Zhao P and Yu B (2004) *Boosted LASSO (Tech. Rep. No. 2004)*. Berkeley: University of California.

Cite this article: Bee S, Poole J, Worden K, Dervilis N and Bull L (2024). Multitask feature selection within structural datasets. *Data-Centric Engineering*, 5, e4. doi:10.1017/dce.2024.1