Investigating the usage of Proxy-*A*-Distance as a measure of dataset shift detection and quantification in an automotive booming noise classification setting

Deepti KUNTE[1,2], Bram CORNELIS[1], Claudio COLANGELI[1], Konstantinos GRYLLIAS[2,3,4]

[1]Siemens Digital Industries Software NV, Belgium
[2]KU Leuven, Department of Mechanical Engineering, Belgium
[3]Flanders Make @ KU Leuven, Belgium
[4]Leuven.AI - KU Leuven Institute for AI, Belgium
deepti.kunte@siemens.com

Abstract

Machine learning methods, which can be effective tools for NVH end-of-line testing applications, are typically data-demanding. Techniques like transfer learning and data augmentation have been proposed to overcome this need. For effective transfer learning, it is imperative to compare the source and target datasets and assess the disparity between the two.

In this paper, we study the applicability of Proxy-*A*-distance for dataset shift detection and quantification. We use the case-study of booming noise detection in automotive end-of-line quality testing with simulated class distribution and mixture component shifts for this investigation. The Proxy-*A*-distance method works with the help of a domain classifier and emerges as a straightforward and reliable procedure for shift detection. Furthermore, its gradual monotonic rise with increasing dataset divergence makes it suitable for shift quantification. However, it is important to note that there seems to be a low correlation between Proxy-*A*-distance and transferability, which warrants further exploration.

Keywords*: Dataset shift, transfer learning, higher order discrepancy measures, proxy-A-distance, booming noise, end-of-line testing.*

1 Introduction

In recent years, machine learning (ML) methods have emerged as powerful tools for various NVH applications (Noise, Vibration, and Harshness). They can enable efficient and automated analysis of NVH-related issues. However, their effectiveness in NVH-based automotive testing applications is often limited by the need for large amounts of data. ML algorithms require a significant amount of data to effectively learn patterns and make accurate predictions or classifications. In the context of NVH-based end-of-line testing, acquiring a substantial dataset that accurately represents the real-world scenarios can be a complex and resource-intensive task. Furthermore, labelling these large datasets is time-consuming due to the subjective aspect of many NVH issues. The scarcity of labeled data poses a significant obstacle to the successful deployment of ML models.

To mitigate the data scarcity issue, researchers have proposed several strategies, such as transfer learning and data augmentation. Transfer learning allows the utilization of knowledge learned from a source dataset, which typically has abundant data, to improve the performance on a target dataset, with limited data. In the automotive end-of-line testing context, transfer learning can be used to transfer knowledge from an existing vehicle model to a new vehicle model, from one production unit to another, and so on.

The success of transfer learning depends on multiple factors, primary among which is the similarity between source and target domains. Understanding the differences in class distributions, input feature representations, and potential shifts in data characteristics is crucial for successful knowledge transfer. In this paper, we delve into the study of comparing source and target datasets to assess the disparity between them for effective transfer learning in NVH-based end-of-line testing applications.

The study analyses the specific case of booming noise classification. Detection of booming noise is crucial for vehicle manufacturers since it is a significant source of acoustic discomfort in vehicle cabins. We generate the data needed for this investigation with the help of a vehicle sound synthesis technique where synthetic booming noise events are added to baseline vehicle sound recordings. Data generation and shift simulation is further elaborated on in section 3.1. Synthesizing the data gives us the flexibility to generate different sets of data viz. source and target, and to introduce shifts due to different causes and of varying magnitudes. Shift due to two causes, mixture component shift and class distribution shift, is simulated. Mixture component shift is simulated by using two different vehicles as the source and target domains while different severities of class distribution shift are achieved by using the same vehicle for source and target datasets but changing the prevalence of booming. Section 3 describes the methodology used. Section 4 discusses the results and limitations followed by conclusions in section 5.

2 Dataset shift

Dataset shift is a difference in the joint distributions of the source and target datasets [1], [2]. In the context of transfer learning, the authors in [3] have identified and categorized the use-cases of dataset shift detection and quantification into three primary families:

- Data selection: To select a source dataset(s) with the best alignment with the target dataset.
- Learning representations: To learn domain-invariant representations of the source and target data through the process of transfer learning.
- Decisions in the wild: To get an estimate of the pre- and post- transfer learning performance of the machine learning models on the target domain.

From these applications, we can formulate two necessary conditions for an ideal domain divergence metric.

1. Monotonic increase with discrepancy: The metric should increase gradually and monotonically as the source and target datasets grow apart.
2. Estimation of transferability: The metric should be correlated to the performance of the target domain on a machine learning model trained solely on the source domain.

2.1 Types of dataset shifts

Dataset shift can be categorized into four main groups depending on the causal variable, the probability distributions which remain constant and the probability distributions which change [1]. Let X and Y be the set of inputs and output of the machine learning model, and P_{source} and P_{target} be the probability distributions corresponding to the source and the target datasets respectively. A causal model would be represented as $X \rightarrow Y$ and an anticausal model would be represented as $Y \rightarrow X$. The four types of dataset shift can then be given as follows [1]:

- Covariate shift: Covariate shift, applicable only to $X \rightarrow Y$ scenarios, is defined as the situation where $P_{source}(x) \neq P_{target}(x) \quad but \quad P_{source}(y|x) = P_{target}(y|x)$
- Prior probability shift: Prior probability shift, applicable only to $Y \rightarrow X$ scenarios, is defined as the situation where $P_{source}(y) \neq P_{target}(y) \quad but \quad P_{source}(x|y) = P_{target}(x|y)$
- Concept shift: In $X \rightarrow Y$ scenarios, concept shift is said to occur when $P_{source}(y|x) \neq P_{target}(y|x) \quad but \quad P_{source}(x) = P_{target}(x)$. While in $Y \rightarrow X$ scenarios, concept shift is said to occur when $P_{source}(x|y) \neq P_{target}(x|y) \quad but \quad P_{source}(y) = P_{target}(y)$.
- Other types of shifts: All other varieties of dataset shifts which cannot be classified into the three groups above are assigned to the last category. This would include scenarios where both the conditional and marginal distributions vary from the source to the target domain making this the most difficult category.

2.2 Measures of dataset shift

We refer to [3], for categorization of domain divergence measures. They classify the measures into three groups:

- Geometric measures: These measures calculate the divergence between the two datasets as a distance between the input features (X), in a metric space. Some examples of geometric measures are Manhattan distance, Euclidean distance, Cosine distance (1-Cosine).
- Information theoretic measures: Methods which attempt to measure the distance between the probability density functions of the two datasets fall under this category. f-divergences (e.g. KL and JS divergence), α-divergences (e.g. Renyi divergence), Wasserstein distance and the Bhattacharya coefficient are all information theoretic measures.
- Higher order measures: These measures measure the divergence between the datasets in a projected space after having matched their higher order moments. Maximum mean discrepancy, CORAL, CMD, proxy-A-distance and KL-divergence fall in this category.

3 Methodology

3.1 Data generation

The data is created using a Virtual Sound Synthesis tool based on [4] and [5]. The tool uses data from healthy vehicle recordings and knowledge of the fault to generate new sounds. More details can be found in [6]. This method lets us create a diverse and large dataset to simulate real life conditions of a fleet of machines/vehicles. We start by creating a pool of 5000 faulty (booming) + 5000 healthy (non-booming) sounds for two vehicles. The source and target datasets are made by drawing sound samples from this pool as per requirement. The source dataset is made up entirely from one vehicle (referred to as the 'Primary car') while the target dataset contains some percentage of data from the other vehicle too (referred to as the 'Contaminant car'). All sampling is done without replacement to avoid data leakage. The process of forming the target dataset is subject to the dataset shift to be inserted and is explained in subsection 3.1.1.

Three booming related profiles: order 2, time varying loudness, and time varying sharpness tracked with the engine rotational speed, are extracted from the sounds to form the inputs to the machine learning models. The preprocessing steps are elaborated on in appendix A.

3.1.1 Dataset shift simulation

In practice, it might be difficult to determine the type of shift present in the data. However the causes of shift might be easily ascertainable. In this study, we simulate dataset shift not of a specific type but due to a specific cause.

In [2], the author details practical causes of dataset shift and their effect on the probability distributions. In the current work we focus on two of these viz., class distribution shift and mixture component shift. As mentioned earlier, the source dataset consists of samples drawn solely from the primary car, with a balanced distribution of healthy and faulty samples (50% healthy and 50% booming). The target dataset is sampled in an explicit manner to introduce dataset shift due to the two causes:

- Class distribution shift (Imbalanced data): Imbalanced data is the shift caused due to a difference in the proportion of different fault classes in the source and target datasets [2]. Here we simulate the dataset shift due to a change in class distribution by deliberate alteration of the ratio of healthy to faulty sound samples. The percentage of faulty samples in the target dataset is gradually varied from 0 to 100 in intervals of 10.
- Mixture component shift: If the global distribution (from which the source and target sets are drawn) is made up of data from different sub-populations with varying characteristics, the differences in the proportions of these sub-populations in the source and target sets leads to a dataset shift called the mixture component shift [2]. It is important to note that this difference is not due to the sampling process but due

to change in the underlying causes. We simulate mixture component shift by contaminating the target dataset with sound samples from another vehicle (Contaminant car). Similar to class distribution shift, the contamination percentage is also varied from 0 to 100 in intervals of 10.

3.2 Proxy-*A*-distance

The proxy-*A*-distance (PAD) has its roots in the quest for estimation of target domain performance bounds. In [7] and [8], the authors postulate that the target domain error can be bounded by knowing the source domain error, the divergence between the source and target domains (difference in marginal distributions) and the difference between the true labelling functions of the source and target domains (difference in conditional distributions). The domain divergence can be measured using L1 or variational distance. However, since it is not feasible to calculate the variational distance from finite samples, domain divergence can instead be estimated via the *A*-distance [9], [7]. They further show that the *A*-distance can be approximated from the error of a domain classifier leading to the metric called *'Proxy-A-distance'* (PAD). A classifier is trained to separate the source and target domains (the domain classifier). PAD is calculated from the classifier error as follows [10]:

$$PAD = 2 * (1 - 2 * \varepsilon) \tag{1}$$

where ε is the domain classifier error.

The PAD thus has a linear relation with the accuracy of a domain classifier. For a properly trained classifier, a PAD of 0 signifies negligible distance between the source and target domains. It corresponds to a classifier accuracy of 50%, which means that the domain classifier is unable to separate the source and target domains. A PAD value of 2 on the other hand shows that the classifier is able to distinguish between the two domains as the classifier reaches a 100% accuracy ($\varepsilon = 0$).

3.3 Hypothesis testing

In order to evaluate if the source and target datasets are significantly different, we formulate the following null hypothesis (H_0): '*there is no statistical difference between the source and target datasets*' (the distance between the source and target dataset is not significantly more than the distance between the source dataset and itself).

$$H_0 : d(S \Rightarrow T) - d(S \Rightarrow S) = 0 \tag{2}$$

The alternative hypothesis (H_A) would then be: '*the source and target datasets are significantly different*' (the distance between the source and target datasets is higher than the distance between the source dataset and itself).

$$H_A : d(S \Rightarrow T) - d(S \Rightarrow S) > 0 \tag{3}$$

Here, $d(A \Rightarrow B)$ represents the measured distance between sets A and B.

We use a right-tailed hypothesis test with an α of 0.01. The null hypothesis is rejected if $p - value < \alpha$.

3.4 Experiments

The methodology is summarized in figure 1. We start by sampling the source and target datasets. A Support Vector Machine (SVM) [11] with linear kernel and random initial seed is used as a domain classifier to differentiate between the two datasets. The classifier error is used to calculate PAD. This process is repeated 10 times with randomly drawn data samples for the hypothesis testing. The result of this test is the rejection or non-rejection of the null hypothesis (Equation 2).

Table 1 shows the number of samples used for training and testing. The two shifts were executed simultaneously giving us 11x11 target datasets. The design matrix is shown in table 2.

We form 6 test cases by choosing one of the three vehicles, Ford Mondeo, Ford Focus and Opel Vectra, as a primary car and another one as a contaminant car.

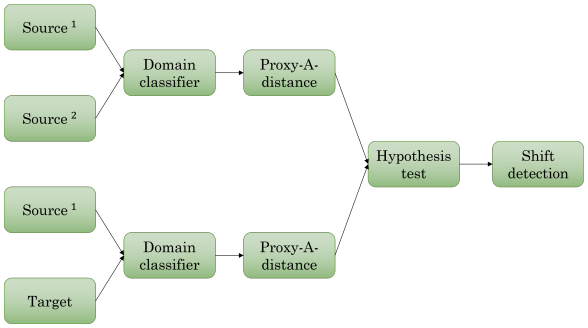


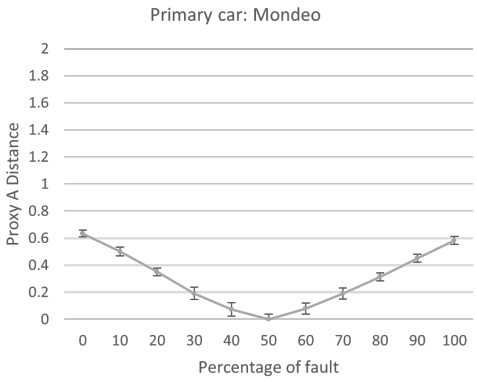
Figure 1: Methodology (The superscripts, 1 & 2, represent disjoint sets drawn from the source domain)

Table 1: Number of samples in training and testing sets of the source and target datasets

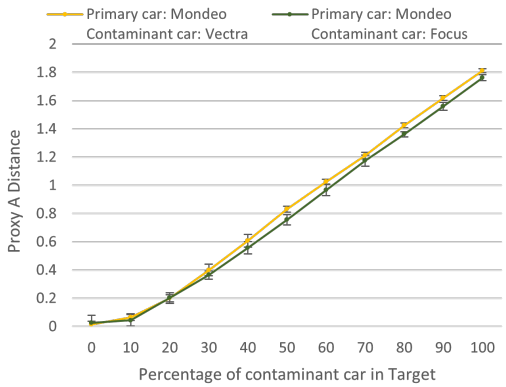
Domain	Set	Dataset size
Source	Train	1000
	Test	1000
Target	Train	1000
	Test	1000

4 Results and discussion

Figures 2a and 2b show the variation of PAD with different levels of solely class distribution shift and mixture component shift respectively. We can see that as the class distribution gradually diverges from the balanced setting of the source dataset (50% healthy and 50% faulty), PAD distance gradually increases c.f. figure 2a . Similarly, as the contamination due to car 2 in the target dataset increases (source dataset has 0% contamination), so does PAD c.f. figure 2b. A special case of this shift is observed on the rightmost side of figure 2b, when the entire source and target datasets are sampled from two different cars. Here the PAD metric is the highest.



(a) Class distribution shift



(b) Mixture component shift

Figure 2: Variation of proxy-*A*-distance with the class distribution shift and mixture component shift

With the simultaneous introduction of the two different shifts, the variation of PAD can be observed in a matrix plot as shown in 3. Each cell corresponds to a target dataset with the level of shift indicated by the two axes. The values in the box correspond to the average PAD between that particular target dataset and the source dataset. The cell colours correspond to the significance of the test. The blue colour of the cells represents rejection of the null hypothesis thus indicating the success of the PAD metric in detecting dataset shift. The red colour, on the other hand, indicates a non-rejection of the null-hypothesis with $p - value \geq \alpha$.

We assess the monotonicity of PAD using contour plots. As the datasets grow gradually more different, a good metric is expected to give gradually increasing values of discrepancy/distance. The PAD metric fulfills this criterion as can be seen in figure 4. Similar to the matrix plot in figure 3, blue points represent rejection of

Table 2: Design of experiments: Target dataset formation (Each cell in the table represents a unique target dataset configuration formed with the given percentage of faulty and contaminant car samples.)

% Fault \ % Contaminant car	0	10	20	30	40	50	60	70	80	90	100
0	0 0	10 0	20 0	30 0	40 0	50 0	60 0	70 0	80 0	90 0	100 0
10	0 10	10 10	20 10	30 10	40 10	50 10	60 10	70 10	80 10	90 10	100 10
20	0 20	10 20	20 20	30 20	40 20	50 20	60 20	70 20	80 20	90 20	100 20
30	0 30	10 30	20 30	30 30	40 30	50 30	60 30	70 30	80 30	90 30	100 30
40	0 40	10 40	20 40	30 40	40 40	50 40	60 40	70 40	80 40	90 40	100 40
50	0 50	10 50	20 50	30 50	40 50	50 50	60 50	70 50	80 50	90 50	100 50
60	0 60	10 60	20 60	30 60	40 60	50 60	60 60	70 60	80 60	90 60	100 60
70	0 70	10 70	20 70	30 70	40 70	50 70	60 70	70 70	80 70	90 70	100 70
80	0 80	10 80	20 80	30 80	40 80	50 80	60 80	70 80	80 80	90 80	100 80
90	0 90	10 90	20 90	30 90	40 90	50 90	60 90	70 90	80 90	90 90	100 90
100	0 100	10 100	20 100	30 100	40 100	50 100	60 100	70 100	80 100	90 100	100 100

the null hypothesis while red points represent non-rejection of the null hypothesis. The vertical axis corresponds to the average PAD between a particular target dataset and the source dataset. The colourbar also corresponds to the PAD value.

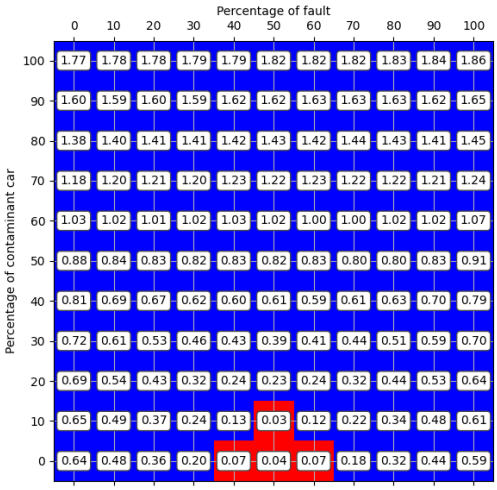


Figure 3: Variation of proxy-*A*-distance with combined class distribution shift and mixture component shift

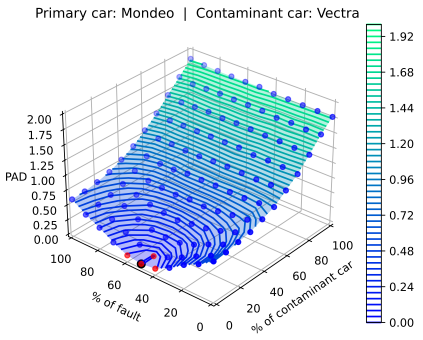


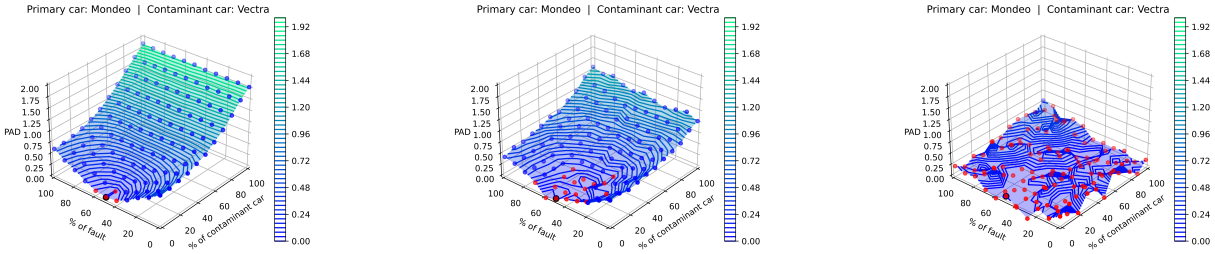
Figure 4: Variation of proxy-*A*-distance with combined class distribution shift and mixture component shift

4.1 Limitations

The major limitations we observed with the PAD metric are:

- Negative influence of data availability: This is intuitive as the domain classifier, which forms the core of PAD, like all learning algorithms has difficulty working with limited training data. This conflicts with the application at hand viz. transfer learning, whose aim is to address data scarcity and thus has limited target data. However, we argue that this drawback is likely to be seen in any method which estimates the divergence empirically.

  In our experiments, we reduce both the source and target datasets to see the effect of the reduction on the PAD metric. From figures 5b and 5c, we can see the decrease in the PAD metric as the classifier training data decreases (w.r.t. the baseline condition 5a). We can also observe the increase in the number of red points showing an increased inability to reject the null hypothesis. Lastly, the PAD metric surface is no longer monotonic, as can be witnessed by the appearance of loops in the contour plots (in the areas with red points). However, since the null hypothesis can no longer be rejected, making any conclusions about monotonicity would be irrelevant.



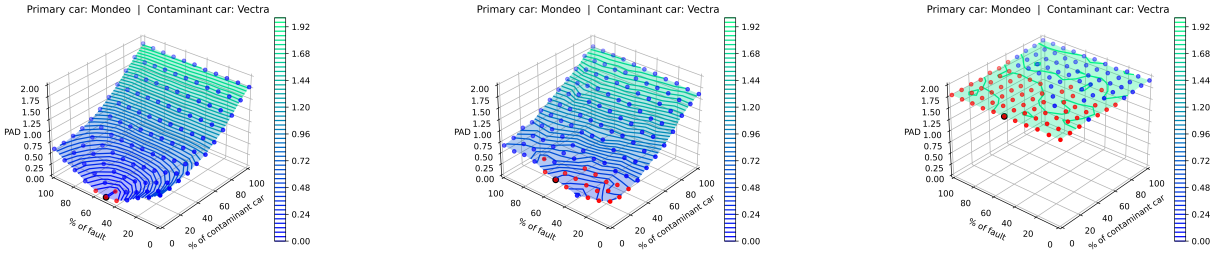
(a) Source 1000, Target 1000 (b) Source 100, Target 100 (c) Source 10, Target 10

Figure 5: Effect of dataset size on the PAD metric (The numbers adjacent to *source* and *target* represent the number of samples in those datasets)

- Unsuitability for unequal source and target datasets: This stems from the formulation of the PAD metric and the tendency of machine learning algorithms to learn trivial relations under heavy imbalance (prediction of majority class at all times). Up to some extend, this drawback appears to be easier to overcome than the previous one by using data imbalance management techniques like over/under-sampling, imposing class weights, using alternative evaluation metrics etc. One can also think of alternatives to PAD which use other evaluation metrics like recall, precision, AUC-ROC, etc., instead of domain classifier accuracy. However, their theoretical basis would need investigation.

  Figure 6 shows us evolution of the PAD metric with diminishing target dataset size, while keeping the source dataset size constant. Even with a 'balanced' classifier setting (class weights are inversely proportional to the class size) [11] to account for the imbalance, we observe an increase in number of non-significant points (red) in figure 6b. However, the PAD values themselves, at the significant points (blue), seem to be less affected when compared to the previous case cf. figure 5b. In a more extreme case, cf. figure 6c, we get consistently high PAD values as the domain classifier struggles to learn a meaningful classification function. Interestingly, we see that a set of target datasets can still be distinguished from the source dataset (blue points), though with misleadingly high PAD values and lack of monotonic behaviour.

- Lack of correlation to pre-transfer learning performance of the target domain: To test the second requirement (Estimation of transferrability), we train a booming classifier (label classifier) on the labelled source data and test on the target data. The architectural details of the model are reported in appendix B. The Spearman correlation coefficient between target domain booming classification accuracy and PAD is given in table 3. The negative sign indicates degradation of performance with increase in PAD. This is a good sign, however, the absolute values show quite some variation and aren't reliably high.



(a) Source 1000, Target 1000 (b) Source 1000, Target 100 (c) Source 1000, Target 10

Figure 6: Effect of data balance on the PAD metric (The numbers adjacent to *source* and *target* represent the number of samples in those datasets)

Table 3: Spearman correlation coefficient between PAD and label accuracy of target domain data evaluated on source-trained model

Primary car	**Contaminant car**	**Correlation coefficient**
Mondeo	Focus	-0.1858
Mondeo	Vectra	-0.4093
Focus	Mondeo	-0.8506
Focus	Vectra	-0.6207
Vectra	Mondeo	-0.3215
Vectra	Focus	-0.3745

PAD measures distances between datasets via capturing differences in their input distributions, regardless of their effect on the output distribution. Although it is advantageous for a method to be able to work without the need for fault class labels, since it does not take the fault labelling function into consideration in any way, the PAD metric cannot guarantee correlation with the target label classification accuracy.

Authors in [12] suggest a way to determine the malignancy of shift. If one could label a small set of well-selected samples from the target domain, the performance of the label classifier on this set could give an estimate of the target domain performance. The samples selected for this purpose are the ones which are confidently labelled as 'target' by the domain classifier model.

- Susceptibility to saturation: Since the PAD metric is calculated from the domain classifier error, it will saturate once the domain classifier accuracy reaches 100%. If the datasets continue to grow apart, PAD value cannot increase beyond 2 to capture this. However, although this affects dataset shift quantification, it does not compromise the efficacy of PAD for dataset shift detection.

5 Conclusions

We test the application of PAD in the context of sound quality analysis, specifically booming noise classification. This is done by simulating dataset shift due to two causes viz., class distribution shift and mixture component shift. We highlight the strengths and weaknesses of the PAD metric and postulate ways to overcome the limitations. The PAD metric is adept at detecting dataset shifts. It is also easy to implement and works without target domain labels. Its main flaw appears to be its unsuitability to work with less and imbalanced data.

We started the study by formulating two requirements for a good discrepancy metric viz. monotonic increase with discrepancy and estimation of transferability. The PAD, when evaluated on the raw input space fulfills the first one but not the second. In our next work we would like to test other measures like MMD, Wasserstein distance, etc., and compare their effectiveness as discrepancy measures. We will also investigate if transferability can be better estimated by comparing the datasets in a latent space (of a fault classifier) instead of the input space [12].

Acknowledgement

We gratefully acknowledge the European Commission for its support of the Marie Sklodowska Curie program through the H2020 ETN MOIRA project (GA 955681).

References

[1] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[2] A. Storkey *et al.*, "When training and test sets are different: characterizing learning transfer," *Dataset shift in machine learning*, vol. 30, pp. 3–28, 2009.

[3] A. R. Kashyap, D. Hazarika, M.-Y. Kan, and R. Zimmermann, "Domain divergences: a survey and empirical analysis," *arXiv preprint arXiv:2010.12198*, 2020.

[4] M. Sarrazin, K. Janssens, H. Van der Auweraer, W. Desmet, and P. Sas, "Virtual car sound synthesis approach for hybrid and electric vehicles," *SAE International 2012*, 2012.

[5] M. Sarrazin, C. Colangeli, K. Janssens, and H. van der Auweraer, "Synthesis techniques for wind and tire-road noise," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 247, pp. 2303–2312, Institute of Noise Control Engineering, 2013.

[6] D. Kunte, B. Cornelis, C. Colangeli, C. De Veuster, and K. Gryllias, "Transfer learning for unsupervised booming noise classification," in *International Conference on Noise and Vibration Engineering*, KU Leuven Mecha(tro)nic System Dynamics (LMSD) division, September 2022.

[7] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, 2006.

[8] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, pp. 151–175, 2010.

[9] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *VLDB*, vol. 4, pp. 180–191, Toronto, Canada, 2004.

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

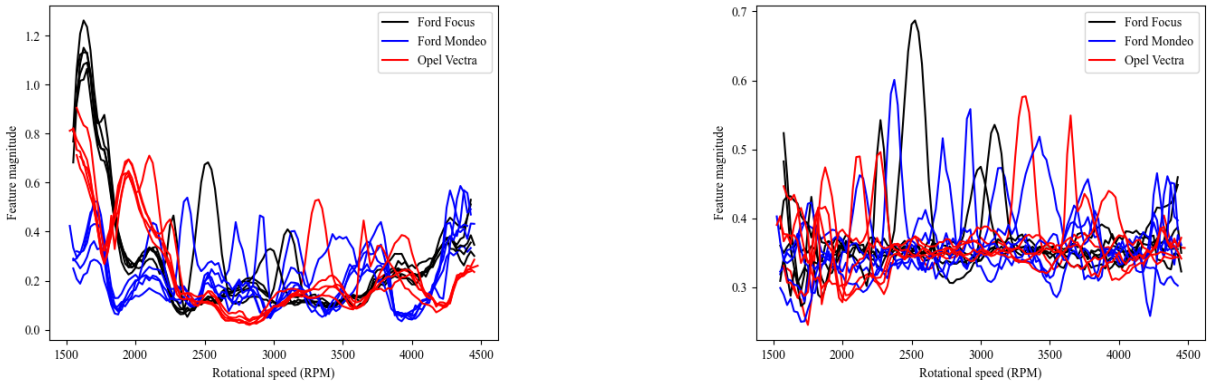
[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[13] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

A Pre-processing

The raw input features are scaled and detrended before going to the domain and label classifiers as shown in figure 7 with the order profile. Detrending makes the inputs more similar to each other.



(a) Raw input

(b) Detrended and scaled input

Figure 7: Effect of preprocessing on vehicle second order profile shown on 5 randomly picked samples from each of the datasets [6]

B Label classifier architecture

The label classifier is a 1D-Convolutional Neural Network with the architecture as shown in figure 8. The activation functions for all the layers is ReLU, except for the output layer which uses a sigmoid activation. The implementation was done with the help of the keras library [13]. The model was allowed to train for a 100 epochs with a batch size of 50 and 20% of the data reserved for validation. Optimization was done using the Adam algorithm with a learning rate of 0.001.

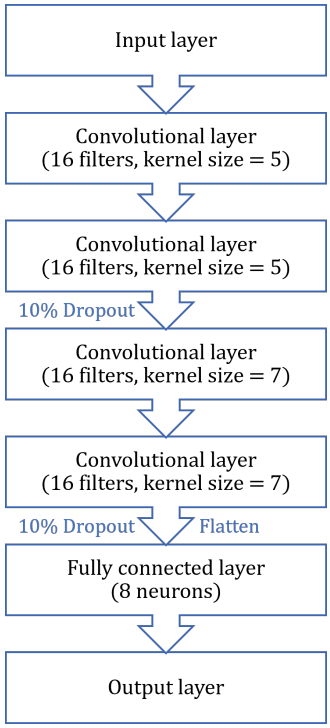


Figure 8: Label classifier architecture.