Transfer learning for unsupervised booming noise classification

D.S. Kunte [1,2], **B. Cornelis** [1], **C. Colangeli** [1], **C. De Veuster** [1], **K. Gryllias** [2,3]
[1] Siemens Digital Industries Software NV,
Interleuvenlaan 68, Leuven, Belgium
e-mail: **deepti.kunte@siemens.com**

[2] KU Leuven, Department of Mechanical Engineering,
Celestijnenlaan 300, B-3001, Heverlee, Belgium

[3] DMMS Core lab, Flanders Make, Belgium

Abstract
Booming noise is one of the primary sources of acoustic discomfort in a vehicle cabin at low frequencies. It is thus of interest to detect booming noise in end-of-line NVH quality testing. However, it becomes infeasible to collect and label large datasets to build reliable booming noise classification models for each vehicle. We therefore study the application of transfer learning techniques, specifically domain adversarial neural networks, for classification of unlabelled datasets using knowledge from a labelled source dataset. The approach was tested on three different vehicles and six combinations of source and target datasets. The datasets are generated by adding booming features to baseline vehicle sound recordings through alteration of the engine orders by means of vehicle sound synthesis techniques. An additional setting where the inputs are detrended, to reduce the gap between the source and target distributions, was also studied. The domain adversarial training approach led to performance enhancements, especially when using the raw inputs.

1 Introduction

Cabin sound quality is one of the key influencers of perceived vehicle quality. The following are the major contributors to vehicle interior noise: engine and powertrain noise, aerodynamic noise, road noise, brake noise, squeak, rattle and tizzes, and noise from auxiliary components [1]. The inconvenience caused by these noise sources has worsened with the recent trend towards lightweighing technologies, thus increasing the need for effective sound quality prediction models [2].

In this paper, we target one specific sound quality issue: powertrain booming noise, caused by excitation of the passenger cavity by narrow-band disturbances closely associated with the fundamental engine firing frequency [3]. Although there have been multiple attempts to establish a booming noise metric, it remains a subjective measure [3], [4], [5]. We study the phenomenon in the context of fleet testing. For the same, we generate a simulation-based dataset for three vehicles: Ford Focus, Ford Mondeo and Opel Vectra.

Multiple recent studies have shown the capability of machine learning models, in particular Convolutional Neural Networks [6], [7], [8], [9], for audio classification. Machine learning models are also being increasingly used for sound quality prediction [10], [11], [12] owing to their effectiveness in modelling non-linear phenomena. These techniques require large labelled training datasets. Moreover, a high prediction performance is not guaranteed for samples outside of the training distribution.

To overcome this need for large datasets, we explore the use of transfer learning techniques for detection of booming noise. The data scarcity in the target dataset is analysed under the following scenarios: a) Scarcity in number of samples in the target dataset: Here we assume access to a small set of labelled samples from the target dataset. b) Scarcity of labels: In this scenario, one has access to a sufficient number of samples in

the target dataset, albeit without labels. An earlier work investigated the first scenario and found significant improvement in model performance with fine tuning and feature extraction techniques [13]. The current study examines the second scenario.

Domain adaptation is a special case of (transductive) transfer learning where the source and target domains share a feature space, but differ in their marginal probability distributions, while the source and target tasks are the same [14]. The domain is defined as the input feature space and its corresponding marginal probability distribution, while the task is defined as the label space and the objective predictive function mapping the inputs to the outputs [14]. Most of the domain adaptation techniques work by aligning the source and target domains. Depending on the method of alignment used, they can be classified as divergence-based methods, reconstruction-based methods and adversarial methods [15]. As the name suggests, Domain Adversarial Neural Networks (DANN) is a domain adaptation method based on adversarial training. It is effective at mapping the inputs to intermediate features which are indistinguishable with regards to their origin, while still containing discriminatory information regarding the output class [16]. The method is further elaborated upon in Section 3.4.

The rest of the paper is organised as follow: Section 2 briefly describes the methodology used for simulating the dataset. Following this, we discuss the procedure to build a booming noise classification model. The various steps involved from feature extraction to domain adaptation are explained in Section 3. The results are analytically presented and discussed in Section 4. The paper closes with some conclusions in Section 5.

2 Data simulation

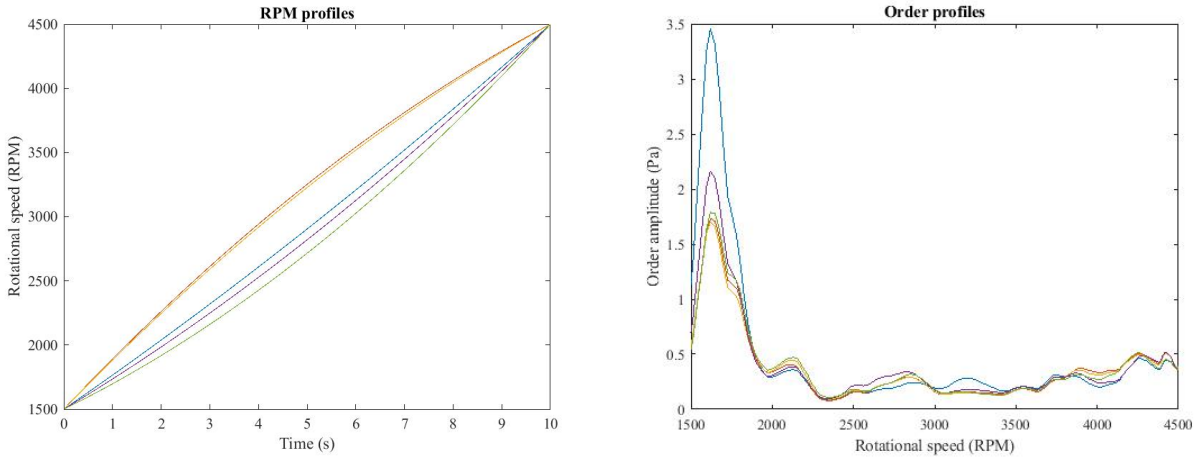
Sounds for the three vehicles were simulated using a sound quality equivalent model built on a small set of experimental recordings at 100% throttle condition. The tonal and broadband components were then extracted, modified and re-synthesized to generate new sounds. The modification was done with the aim to introduce the manufacturing deviations and the booming phenomenon. The sounds were synthesized in a run-up condition with the initial and final rotational speeds fixed at 1500RPM and 4500RPM respectively. Small random variations, were introduced in the speed profile as shown in Figure 1a. Low frequency random fluctuations up to 10dB were added to all the order profiles. An example is shown with the second order of a vehicle in Figure 1b. A Hanning window defined by its height, width and placement in the RPM range was added to the second order profile to model a booming-like event as shown in Figure 2. In an earlier internal jury review it was found that such an event with a width greater than 280RPM and a height greater than 0.5Pa led to a perceivable booming sensation. This criterion was used to label the simulated data. A more detailed explanation of the simulation procedure can be found in [17]. The simulation method is based on the sound synthesis procedure proposed in [18], [19].

2.1 Data distribution

The width and height of the Hanning window was sampled randomly from a beta distribution. Specifications of the distribution are given in Table 1. The same is visualized in Figure 3. The location of the window in the run-up was sampled randomly from a uniform distribution spanning the range of the run-up. Overall, this manner of sampling led to an intentionally imbalanced distribution with $\approx$20% booming samples.

Table 1: Beta distribution specifications for sampling height and width of the booming event

Parameter	Height	Width
Lower limit	0.1	80
Upper limit	1.5	560
Mean	0.5	280
Alpha	2	2



(a) Speed profile

(b) Second order amplitude profile

Figure 1: Example of run up simulations with five randomly picked samples

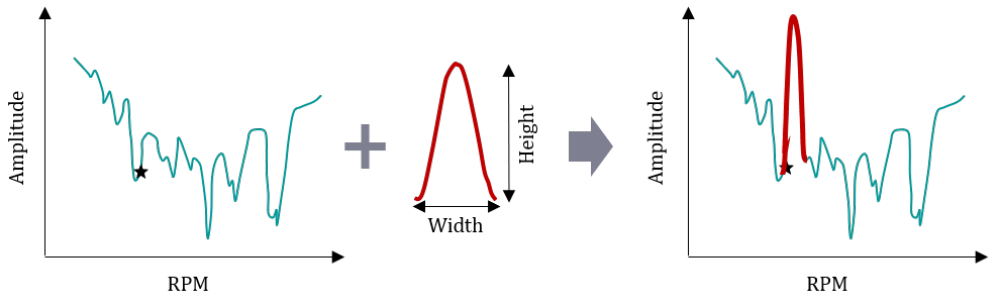


Figure 2: Booming event simulation through the addition of Hanning window

3 Methodology

3.1 Feature extraction

Psychoacoustic metrics are instrumental in capturing perceived sound quality. Several studies have reported high correlation of booming sensation with Loudness and Sharpness, with little to no correlation with Roughness and Fluctuation Strength [20], [3]. To capture the transient nature of the booming phenomenon, time varying Loudness and Sharpness were chosen as inputs to the booming noise classification model. Additionally the dominating order profile associated with the booming phenomenon in internal combustion (IC) engine vehicles was also used. In this study, involving four cylinder engine vehicles, this corresponded to the second order. The three input features, tracked with the rotational speed, were computed using Process Designer in Simcenter Testlab 2021.2 Neo General Processing.

3.2 Preprocessing

The extracted inputs are stacked to create a matrix of size NxTxF of N samples with F features of length T. This input matrix was pre-processed as in the following steps:

1. Detrending: In order to remove the trends associated with the different vehicles, the median feature-profile was removed from all the features. This was done separately for source and target datasets.

2. Scaling: The second step was to scale the features to a range of 0 to 1. The scaling factor was calculated on the source dataset and the same was used for the target dataset.

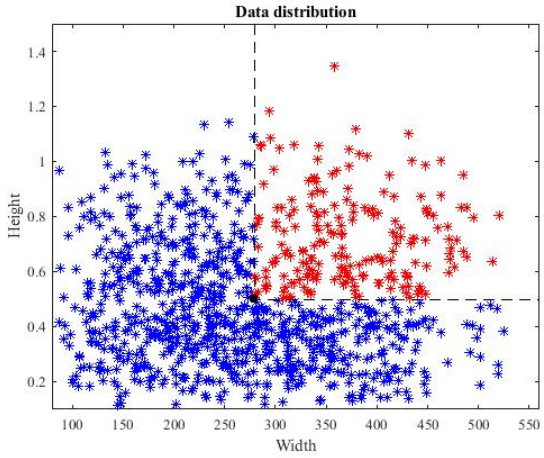


Figure 3: Data distribution in the booming event's width-height space. The red samples correspond to booming and blue samples indicate no booming. The black dot indicates the mean of the distribution

The above two operations for the source and target domains were performed according to the equations below. Validation and test sets are processed similarly, hence only the validation processing is shown in the following equations. Figures 4a and 4b show the effect of the above preprocessing steps on the order 2 profile.

$$S'^{f}_{train} = S^{f}_{train} - M(S^{f}_{train}) \tag{1}$$

$$S'^{f}_{val} = S^{f}_{val} - M(S^{f}_{train}) \tag{2}$$

$$S''^{f}_{train} = \frac{S'^{f}_{train} - min(S'^{f}_{train})}{max(S'^{f}_{train}) - min(S'^{f}_{train})} \tag{3}$$

$$S''^{f}_{val} = \frac{S'^{f}_{val} - min(S'^{f}_{train})}{max(S'^{f}_{train}) - min(S'^{f}_{train})} \tag{4}$$

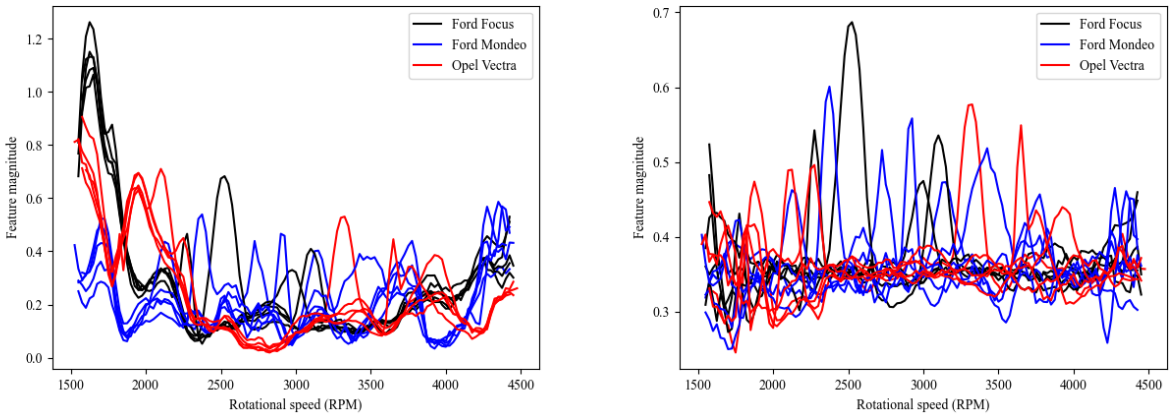
$$T'^{f}_{train} = T^{f}_{train} - M(T^{f}_{train}) \tag{5}$$

$$T'^{f}_{val} = T^{f}_{val} - M(T'^{f}_{train}) \tag{6}$$

$$T''^{f}_{train} = \frac{T'^{f}_{train} - min(S'^{f}_{train})}{max(S'^{f}_{train}) - min(S'^{f}_{train})} \tag{7}$$

$$T''^{f}_{val} = \frac{T'^{f}_{val} - min(S'^{f}_{train})}{max(S'^{f}_{train}) - min(S'^{f}_{train})} \tag{8}$$

where *S* and *T* denote the source and target domains respectively, the subscripts *train* and *val* refer to training and validation sets respectively, the superscript f refers to a feature $f \in F$, *A, A'* and *A"* represent raw, detrended and scaled data respectively, *min(A)* and *max(A)* denote the minimum and maximum values in the data matrix A, and $M(A^f)$ represents the median profile of A^f, obtained by taking the median over all N samples in A $\forall t \in T$.



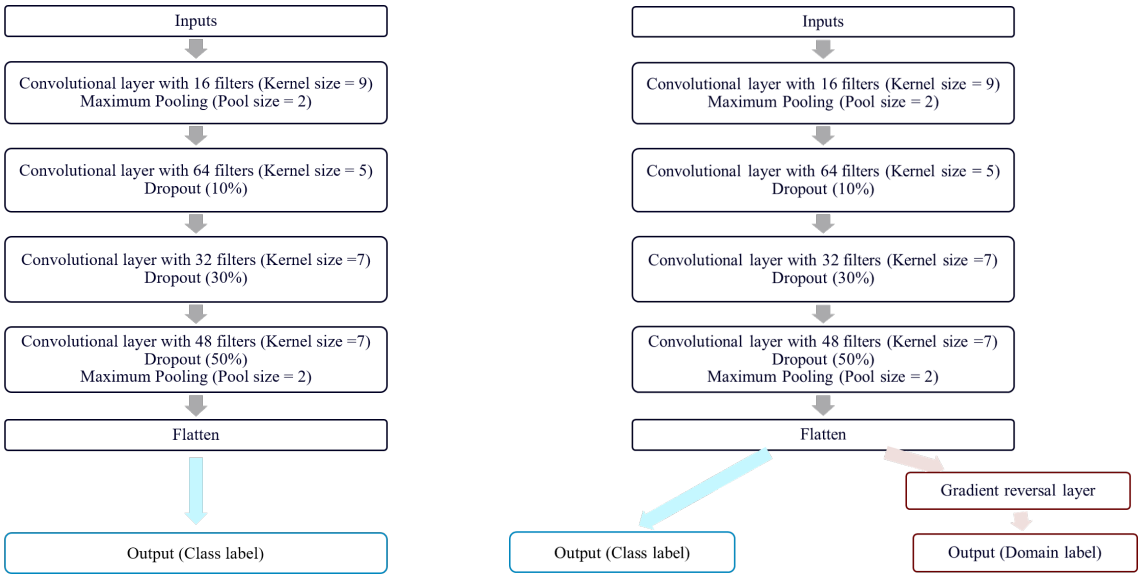
(a) Raw input

(b) After detrending and scaling

Figure 4: Effect of preprocessing on vehicle second order profile shown on 5 randomly picked samples from each of the datasets

3.3 Classification model

A CNN architecture was chosen as the classification model due to its capability to extract local features and a shared weights architecture resulting in shift invariance and reduced number of parameters [21]. Figure 5a shows the CNN model architecture used. A Random search operation was carried out with the help of keras tuner [22] to find the optimal network architecture for a classical convolutional neural network trained and validated only on the source dataset. The tuned parameters include number of layers, number of filters, kernel size, dropout rate, number of neurons, output layer activation function and learning rate.



(a) Conventional CNN architecture

(b) DANN architecture

Figure 5: Model architectures (A Parametric ReLU (PReLU) [23] activation function is used for all convolutional layers and sigmoid activation function for the fully connected layer. Stride of all convolutional kernels is 1. The models were built in the keras framework [24], [25])

3.4 Domain adaptation

The Domain Adversarial Neural Network (DANN) architecture was proposed in [16]. Its conception was done with the aim to generate a representation of the source and target inputs which does not include any discriminatory information about the origin of the samples (source or target domains) while preserving the discriminatory information with respect to the class of the samples (booming or non-booming). A supervised classifier trained on these (source domain) intermediate features can then be used to make predictions about unlabelled target samples thus facilitating unsupervised training on the target dataset.

Figure 6 shows a block diagram of the general architecture of the DANN model. A brief description of the DANN method is given in this section, for more details we refer to the original paper [16]. DANN consists of a feature extractor block followed by a label classifier block, as is the norm for many deep learning architectures. The unique aspect of the model comes from the gradient reversal layer and the domain classifier block. In the forward propagation step, the gradient reversal layer behaves as a unity multiplier i.e., the features pass unchanged through it. During the backward propagation however, the gradient reversal layer multiplies the gradients by a negative constant (-λ). During training, the labelled source data and the unlabelled target data are both passed through the feature extractor to generate input representations. The source domain features are then passed through the label classifier. The label classifier attempts to predict the class of the samples (booming or non-booming) and is trained on the labelled source samples only. Both source and target features are also passed through the gradient reversal layer and the domain classifier. The domain classifier tries to predict the domain of the samples (source or target) and is trained on the combined source and target data since we know the domain of all the samples. The adversarial training of the feature extractor and the domain classifier occurs due to the gradient reversal layer. This forces the feature extractor to generate representations which are conducive for the label classifier but ineffective for the domain classifier.

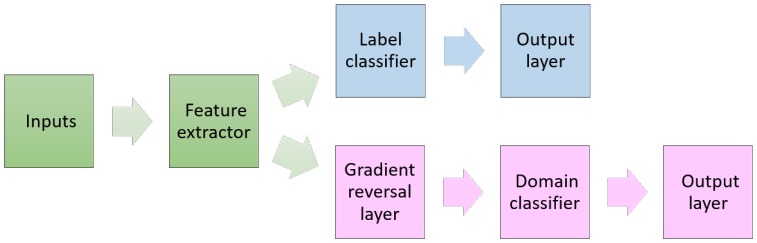


Figure 6: DANN architecture [16]

Let the feature extractor, the label classifier and the domain classifier be defined by a set of parameters θ_f, θ_y and θ_d respectively. For n source samples and m target samples, the DANN tries to minimize the loss given by Equation (9) [16].

$$E(\theta_f, \theta_y, \theta_d) = \frac{1}{n}\sum_{i=1}^{n} L_y^i(\theta_f, \theta_y) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n} L_d^i(\theta_f, \theta_d) + \frac{1}{m}\sum_{i=1}^{m} L_d^i(\theta_f, \theta_d)\right) \tag{9}$$

where $L_y^i(\theta_f, \theta_y)$ and $L_d^i(\theta_f, \theta_d)$ are the class and domain prediction losses respectively for the i^{th} sample. The value of λ in the loss function was set according to Equation (10). The variation of λ is shown in Figure 7.

$$\lambda = \lambda_{max} * \left(\frac{2}{1 + e^{-10*TF}} - 1\right) \tag{10}$$

where TF is the training fraction completed, and λ_{max} is the maximum value lambda will attain in the training loop.

Figure 5b shows the DANN architecture. The DANN model uses the same architecture as the baseline 1D CNN model for the feature extraction, but adds a gradient reversal layer and domain classifier. The learning

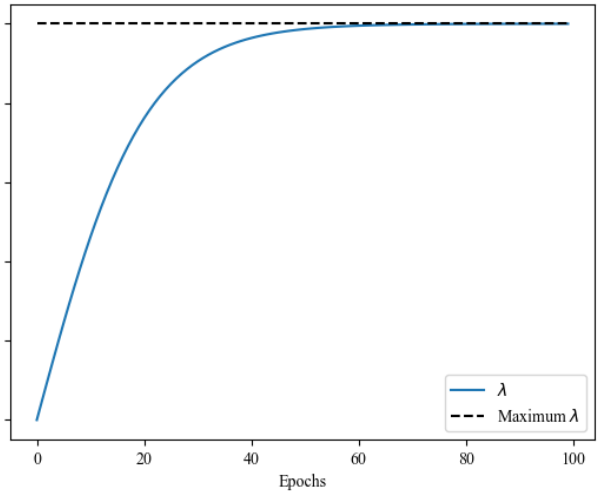


Figure 7: Variation of λ with training progress

rate decay and λ_{max} were the only hyperparameters tuned in the DANN model.

3.5 Experiments

The three vehicles formed a total of six pairings of source and target datasets, each of which was tested with and without domain adaptation, and with and without detrending (preprocessing). An ideal accuracy was also established for each vehicle, and each preprocessing technique, by creating a reference scenario for each vehicle where the CNN has access to the fully labelled dataset for training and the model is tested on samples of the same vehicle. For all the above mentioned experiments the datasets were divided into training + validation and testing according to Table 2.

Table 2: Data division into training, validation and testing

Dataset	Training + Validation	Testing
Source	1000 + 250	1000
Target	1000 + 250	1000

3.6 Evaluation

The models were evaluated by comparing their accuracy, cf. Equation (11). Each experiment was run five times with random seeds and random training-validation split to obtain mean and standard deviation of the model accuracy. This meant that the standard deviation would include the effect of variation in data and in initialization.

$$\text{Percentage accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Total number of samples}} * 100 \tag{11}$$

4 Results and discussion

The methodology discussed above is applied on the three vehicle datasets and their source-target pairings, and the results discussed below.

In [26], the authors proposed a method to estimate the target generalization performance bound in terms of source dataset error and a measure of divergence between the source and target domains. In this study we do not assess the inter-domain divergence, however, we report the source dataset accuracy in Table 3.

Table 3: Percentage mean accuracy $\pm$ standard deviation on the test set with a fully labelled dataset

Vehicle	Scaling	Detrending + Scaling
Focus	93.8 ± 0.1	93.8 ± 0.7
Mondeo	93.2 ± 0.8	94.0 ± 0.7
Vectra	92.8 ± 0.4	93.8 ± 0.7

Table 4 shows the comparative model performance on a target test set with and without the DANN model. We see that the domain adaptation approach led to an increase in booming classification accuracy in all of the studied source-target vehicle pairings and both preprocessing conditions.

Detrending leads to higher test dataset accuracy. This might be due to better initial alignment between the source and target domains. Li et. al. [27] recommend the use of instance-based algorithms, such as weight-estimation based on kernel embedding techniques and the heuristic weighting strategy, in domain adaptation involving similar source and target domains. These techniques might be better suited in this case.

Although the domain adaptation approach resulted in at least marginal improvement in classification accuracy in all the studied cases, the need for hyperparameter tuning on the DANN model is apparent. In addition to optimizing the model architecture, further improvements in accuracy could be obtained through establishment of an early stopping criterion based on comparison of the feature distributions of the two domains.

The significant increase in target dataset accuracy post detrending could also be attributed to the nature of the data generation. All the datasets are simulated from a limited set of experimental recordings. Although random noise is added to the order profiles generation as shown in Figure 1b, it remains to be seen if detrending will lead to such substantial performance gain in non-simulated environments. The ability of the DANN approach to obtain a positive transfer also needs to be studied in this scenario.

Table 4: Percentage mean accuracy $\pm$ standard deviation obtained on target test set

Source dataset	Target dataset	With scaling		With detrending + scaling	
		w/o DANN	with DANN	w/o DANN	with DANN
Mondeo	Focus	78.9 ± 2.7	83.1 ± 1.5	91.4 ± 0.8	91.5 ± 0.3
Vectra	Focus	83.5 ± 3.6	85.2 ± 2.5	86.1 ± 1.1	88.0 ± 1.1
Focus	Mondeo	71.9 ± 14.3	86.6 ± 0.6	89.9 ± 1.3	91.5 ± 0.6
Vectra	Mondeo	70.9 ± 25.6	86.6 ± 0.8	88.9 ± 1.2	90.6 ± 0.3
Focus	Vectra	27.1 ± 3.9	82.3 ± 2.8	85.3 ± 1.0	88.1 ± 1.2
Mondeo	Vectra	35.9 ± 10.3	83.4 ± 1.0	87.1 ± 0.2	88.5 ± 0.5

Some vehicles seem to be more suited in their role as source datasets than others (e.g., Vectra). Further investigation is needed to understand what leads to this effect, which could in turn be instrumental in informed selection of source datasets.

Lastly, regarding the DANN model, the authors observed a significant increase in training time over a conventional CNN model. This could be attributed to its non-standard architecture in TensorFlow [25] and the more difficult convergence of the adversarial training process.

5 Conclusions

The purpose of this study was to investigate the efficacy of transfer learning techniques for detection of booming noise. We also studied the effect of detrending on the classification accuracy and on domain adaptation.

A 1D-CNN approach seems to be well suited for booming noise classification with classification accuracies $\approx 93 - 94\%$ for the three vehicles under study with fully labelled datasets. The highest accuracy on the unlabelled target dataset ($\approx 88 - 91\%$) was obtained through a combination of detrending (preprocessing) and the DANN approach in modelling. We thus demonstrated the feasibility of using a domain adversarial approach for sound quality prediction on unlabelled target datasets. Further research is however required to improve the training time and the accuracy of the DANN approach (e.g. through hyperparameter tuning). Furthermore, further investigations are required on the feature distributions of the source and target datasets, in order to understand why some vehicles seemed more suited to be used as source datasets and to understand whether alternative domain adaptation methods such as instance-based transfer may be effective for this use case.

Acknowledgements

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] M. Harrison, *Vehicle refinement: controlling noise and vibration in road vehicles*. Elsevier, 2004.

[2] F. L. M. d. Santos, T. Enault, J. Deleener, T. V. Houcke, and H. V. d. Auweraer, "The combination of testing and 1D modeling for booming noise prediction in the model based system testing framework," in *Special Topics in Structural Dynamics, Volume 6*. Springer, 2017, pp. 103–111.

[3] S.-H. Shin, J.-G. Ih, T. Hashimoto, and S. Hatano, "Sound quality evaluation of the booming sensation for passenger cars," *Applied acoustics*, vol. 70, no. 2, pp. 309–320, 2009.

[4] S. Hatano and T. Hashimoto, "Booming index as a measure for evaluating booming sensation," in *Proc. Inter-Noise*, no. 233, 2000, pp. 1–6.

[5] S.-K. Lee, H.-C. Chae, D.-C. Park, and S.-G. Jung, "Sound quality index development for the booming noise of automobile sound using artificial neural network information theory," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 2002, no. 1. Institute of Noise Control Engineering, 2002, pp. 35–40.

[6] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[7] X. Zhang, Y. Zou, and W. Shi, "Dilated convolution neural network with LeakyReLU for environmental sound classification," in *2017 22nd international conference on digital signal processing (DSP)*. IEEE, 2017, pp. 1–5.

[8] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment sound classification using a two-stream CNN based on decision-level fusion," *Sensors*, vol. 19, no. 7, p. 1733, 2019.

[9] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)*. IEEE, 2015, pp. 1–6.

[10] H. B. Huang, X. R. Huang, R. X. Li, T. C. Lim, and W. P. Ding, "Sound quality prediction of vehicle interior noise using deep belief networks," *Applied Acoustics*, vol. 113, pp. 149–161, 2016.

[11] H. Liu, J. Zhang, P. Guo, F. Bi, H. Yu, and G. Ni, "Sound quality prediction for engine-radiated noise," *Mechanical Systems and Signal Processing*, vol. 56, pp. 277–287, 2015.

[12] H. B. Huang, R. X. Li, M. L. Yang, T. C. Lim, and W. P. Ding, "Evaluation of vehicle interior sound quality using a continuous restricted boltzmann machine-based DBN," *Mechanical Systems and Signal Processing*, vol. 84, pp. 245–267, 2017.

[13] D. S. Kunte, C. Colangeli, B. Cornelis, K. Gryllias, C. De Veuster, and K. Janssens, "Transfer learning for booming noise classification," in *DAGA 2022 - 48. JAHRESTAGUNG FÜR AKUSTIK*. German Acoustical Society (DEGA), 2022, pp. 441–444.

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[15] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 5, pp. 1–46, 2020.

[16] 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.

[17] P. M. G. Leite, "Convolutional neural networks application for the classification of powertrain booming noise on internal combustion engine passenger cars," *MSc Thesis, Faculdade de Engenharia da Universidade do Porto (FEUP)*, 2021.

[18] 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, no. 6. Institute of Noise Control Engineering, 2013, pp. 2303–2312.

[19] 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.

[20] S. Lee and H. Chae, "The application of artificial neural networks to the characterization of interior noise booming in passenger cars," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 218, no. 1, pp. 33–42, 2004.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Kerastuner," https://github.com/keras-team/keras-tuner, 2019.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

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

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[26] 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, no. 1, pp. 151–175, 2010.

[27] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, and K. Gryllias, "A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges," *Mechanical Systems and Signal Processing*, vol. 167, p. 108487, 2022.