EXPLAINING THE DECISIONS OF ANOMALOUS SOUND DETECTORS

Kimberly T. Mai[1,3]*, Toby Davies*[1,3]*, Lewis D. Griffin*[1]*, Emmanouil Benetos*[2,3]

[1] University College London, UK
[2] Queen Mary University of London, UK
[3] The Alan Turing Institute, UK
{kimberly.mai, toby.davies, l.griffin}@ucl.ac.uk emmanouil.benetos@qmul.ac.uk

ABSTRACT

Deciding whether a sound is anomalous is accomplished by comparing it to a learnt distribution of inliers. Therefore, learning a distribution close to the true population of inliers is vital for anomalous sound detection (ASD). Data engineering is a common strategy to aid training and improve generalisation. However, in the context of ASD, it is debatable whether data engineering indeed facilitates generalisation or whether it obscures characteristics that distinguish anomalies from inliers. We conduct an exploratory investigation into this by focusing on frequency-related data engineering. We adapt local model explanations to anomaly detectors and show that models rely on higher frequencies to distinguish anomalies from inliers. We verify this by filtering the input data's frequencies and observing the change in ASD performance. Our results indicate that sifting out low frequencies by applying high-pass filters aids downstream performance, and this could serve as a simple pre-processing step for improving anomaly detectors.

Index Terms— anomaly detection, DCASE challenge, data engineering, interpretablity

1. INTRODUCTION

Anomalous sound detection (ASD) is the task of deciding whether a sound produced from an object is normal or anomalous. This is conducted by comparing the sound to a learnt distribution of inliers. ASD is useful for machine condition monitoring and can result in more timely repairs after unusual sounds are identified. In practice, this task is difficult because only inliers are available for training. In addition, it is hard to anticipate the types of anomalies that may appear (for example, if they manifest in certain frequency bands or at specific times). The task has received more attention in recent years, featuring within the Detection and Classification of Acoustic Scenes and Events (DCASE) challenges for three consecutive years to date [1]. In 2021, the ASD challenge received a total of 75 submissions [2]. Analysing the top submissions, we note they share many similarities. The majority utilise ensembles and use metadata to pre-train their anomaly detection models. However, there is more variability in the usage of data engineering. Notably, the top three entrants do not apply any data engineering through either data augmentations or pre-processing [3, 4].

Data engineering typically modifies the training data using pre-defined rules to improve model generalisation. Whether data engineering helps or harms anomaly detection is debatable within the machine learning community. However, existing studies have focused on the computer vision domain [5, 6]. In this paper, we seek to address this question for ASD. There are many types of data engineering steps that could be applied to audio such as speed perturbations, adding noise, and frequency masking [7]. We choose to focus on frequency-related data engineering, as [8] proposes the *high-frequency hypothesis*: anomaly detectors use higher frequencies to identify anomalies. If this is the case, methods such as random frequency masking could remove discriminative features between inliers and anomalies. This hypothesis could partly explain why the top three submissions outperform the others.

However, [8] does not explicitly substantiate their claim. We conduct a series of experiments in this paper to verify the high-frequency hypothesis. We firstly adapt Sound LIME (SLIME) [9] to a one-class anomaly detector. SLIME can provide an insight into how modifications at an individual sound clip level can affect model predictions through approximating changes in output using a linear model, hence even providing explanations for models with complex global behaviour. We find the majority of the datasets in the DCASE ASD challenge rely on higher-frequency information to make the correct decisions. Moreover, we quantitatively verify SLIME's result by re-training and re-testing our anomaly detectors by applying low and high pass filters to input data in a sequential manner. Our results corroborate that focusing on higher frequencies helps ASD, and simple frequency filtering instead of complicated data engineering is a more measured approach to improving performance.

2. APPROACH

2.1. Dataset

We use the DCASE 2021 Task 2 dataset for our experiments. This dataset combines subsets of the ToyADMOS2 [10] (*ToyCar, ToyTrain*) and the MIMII DUE [11] (*Fan, Gearbox, Pump, Slider, Valve*) datasets. All recordings are single channel, 10 seconds in duration and downsampled to 16 kHz. We refer to the subsets as different machines.

Each machine is subdivided into six sections, which correspond to different properties. Anomaly detectors are evaluated at the section level, using area under the curve (AUC) as the performance metric. Furthermore, each section contains data from two domains: the source domain (for which there is an abundant number of audio clips available) and the target domain representing a domain shift (where only a few clips are available).

The dataset is curated in a one-class fashion: the training split only contains inliers, whereas the test split contains both inliers and anomalies. The training split contains about 1,000 inliers for each of the six sections and is further subdivided into a development set

K. T. Mai, T. Davies and E. Benetos are supported by The Alan Turing Institute under grant EP/N510129/1. K. T. Mai is also supported by EPSRC under grant EP/R513143/1.

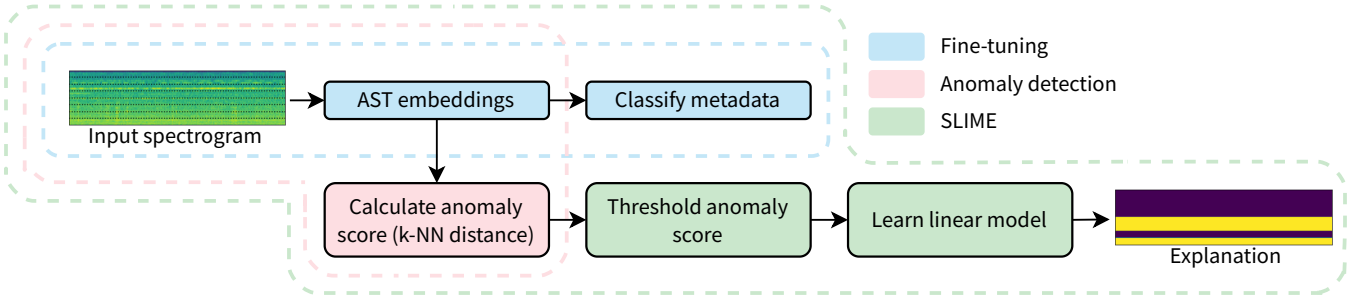


Figure 1: Schematic of the experimental workflow. The different coloured backgrounds correspond to the various stages. For fine-tuning, the AST is tasked with classifying metadata (the correct subset and section). For anomaly detection, the metadata and corresponding classification layer are disregarded. Instead, the embedding at the pre-logits layer of the fine-tuned model are extracted and used to train a k-NN. For SLIME, the input spectrograms are perturbed by segmenting via the frequency axis (denoted by the dotted lines in the spectrogram). The predicted distances produced by the trained k-NN are converted to a classification decision by using the equal error rate as a threshold. SLIME then approximates the classifier locally with an interpretable linear model to produce a heatmap explanation.

containing the first three sections and an evaluation set consisting of the latter three sections. The test split contains the same number of samples for both domains: about 100 inliers and 100 anomalies per section.

2.2. Anomaly detection model

Table 1: Mean AUC scores (%) across all six subsections (source and target) by anomaly detection model. **Bold** denotes the best result.

	Fan	Gearbox	Pump	Slider	ToyCar	ToyTrain	Valve
Autoencoder (Baseline)	64.0	66.8	63.7	69.2	63.2	63.0	53.7
MobileNetV2 (Baseline)	64.7	68.2	64.2	62.6	60.0	59.2	57.1
AST with Mel spectrogram	**74.5**	73.9	69.3	**76.5**	75.9	64.9	**76.4**
AST with STFT spectrogram	71.6	**81.0**	**73.2**	75.5	**78.9**	**65.0**	75.1

We use the same anomaly detection method for our experiments. Namely, we extract features from fine-tuned Audio Spectrogram Transformers (ASTs, 87m parameters) [12] and feed these to a shallow anomaly detection model. We choose to use ASTs as Transformers have demonstrated good anomaly detection performance in other domains [13, 14, 15]. Based on the success of using metadata in the DCASE 2021 submissions for learning representations, we fine-tune ASTs to classify both the machine and section (resulting in a 42-dimensional classification layer) using a cross-entropy loss. We choose to include all seven machines in the fine-tuning stage instead of fine-tuning an AST per machine, as we found that this improved anomaly detection performance. We choose cross-entropy instead of angular losses to reduce the number of hyperparameters that need to be tuned. Starting from ASTs pre-trained on AudioSet [16], we trained our models for a maximum of 10 epochs using the Adam optimiser and a learning rate of $1e-5$. Our fine-tuned model achieved a classification accuracy of 94%.

As an initial experiment to examine the importance of high frequencies, we use both STFT spectrograms and Mel spectrograms (128 bands) as input. We use Torchaudio [17] to compute the input features, using a frame length of 25ms and a hop size of 10ms. In line with the original AST implementation, we also normalise the spectrograms across both the time and frequency axes so that the dataset mean and standard deviation are 0 and 0.5 respectively. After fine-tuning, we extract training features at the pre-logits layer, resulting in a 768-dimensional representation. We use the features to train a k-NN. For inference, we extract test features in the same manner and use the mean distance from a test datum to its nearest neighbours as the anomaly score. We set $k = 1$ after validation. We tried other methods such as maximum softmax probability [18], Mahalanobis distance and isolation forest [19] and found the nearest neighbour approach worked best.

Table 1 illustrates our initial results. Excluding *Fan*, the AST trained with STFT spectrograms either matches or exceeds the AST trained with Mel spectrograms. As Mel weighting loses spectral resolution at high frequencies, this substantiates the high-frequency hypothesis for the majority of machines.

2.3. Preliminaries: Is it appropriate to analyse frequencies?

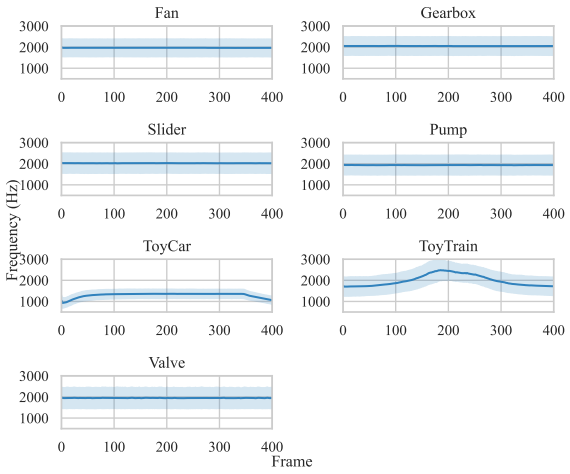


Figure 2: Mean spectral centroids (with standard deviations) across all inlier training samples, calculated using Librosa [20] with `n_fft`$= 2048$ and `hop_length`$= 512$.

To verify whether it is appropriate to examine how frequencies affect ASD, we checked whether the frequency distributions remain stable across different audio clips and times. We measured stability

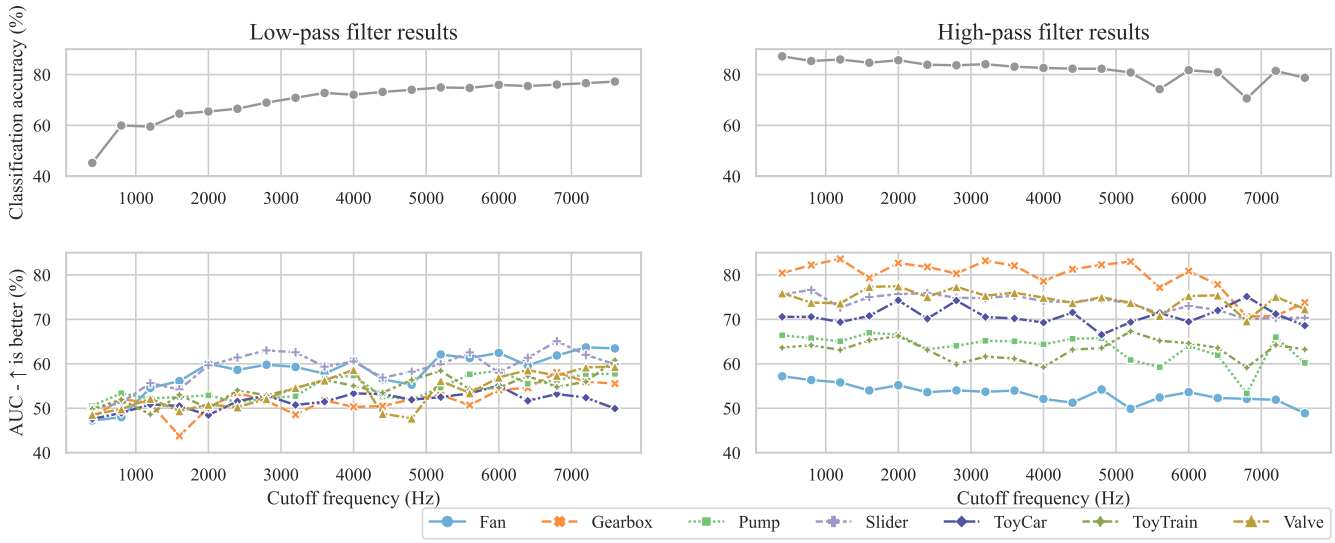


Figure 3: Top: change in classification accuracy after applying frequency filters on Mel spectrograms. Bottom: changes in mean AUC (across all six sections) after applying frequency filters. Applying low-pass filters (left) significantly affects fine-tuning classification accuracy and downstream ASD performance. Conversely, referring to Table 1, classification performance and ASD is remains stable after applying high-pass filtering (right). This indicates low-frequencies potentially obscure discriminative features. The same trends are observed when using STFT spectrograms as input.

by computing the average spectral centroids per frame in the training split for each machine. The spectral centroid is stable across time for most datasets (Figure 2), suggesting it is appropriate to examine the aggregate frequency features across time. We observe some variations to this in *ToyCar* and *ToyTrain*. For these machines, the spectral centroids are stable for most of the clips but dip at the start and end. We speculate this may be due to the 10-second clips being collected in ToyADMOS2 in a different systematic way.

2.4. Adapting local explanations to one-class detectors

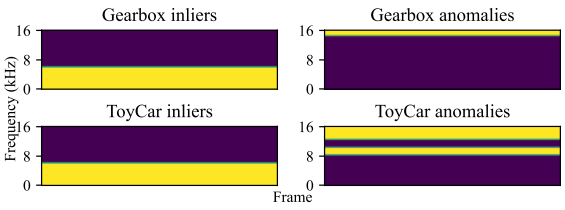


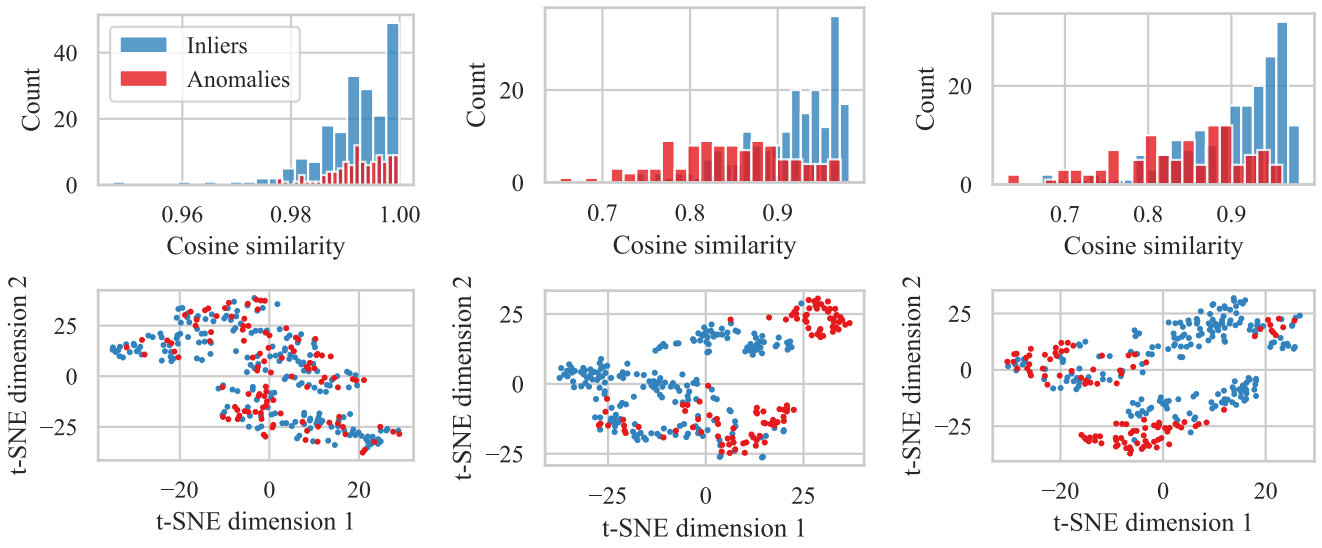
Figure 4: Median heatmaps generated by SLIME on Source Section 0, segmented by subset and condition {inlier, anomaly} . Yellow regions indicate frequency ranges that are more important for correct decisions while purple regions do not contribute to the ASD decision.

The differences in ASD performance using STFT spectrograms instead of Mel spectrograms in subsection 2.2 indicates that generally, maintaining spectral resolution at high frequencies is helpful. However, these AUC scores only give us a global insight into a detector's behaviour and do not account for differences at a section level. Sound local interpretable model-agnostic explanations (SLIME) [9] can provide a localised understanding. SLIME generates synthetic samples by firstly segmenting input spectrograms into fixed components in the time or frequency axis. It then randomly occludes input components using the segments and observes the changes in a model's behaviour by approximating the output using a linear model. The changes in a model's decision are visualised in a heatmap. In particular, the heatmaps depict regions contributing or detracting from the prediction. However, SLIME is typically used to interpret classifiers with distinct decision boundaries. Such boundaries are not necessarily present in anomaly detectors. Instead, a detector's output often measures the distance from the inlier training distribution. Examples of these types of anomaly detectors include Mahalanobis distance, k-nearest neighbours and one-class support vector machines.

To adapt SLIME to work for distance-based ASD, we convert the distances to labels $L = \{0 = \text{inlier}, 1 = \text{anomaly}\}$ using the equal error rate (EER) as the threshold. EER indicates the distance where the false-positive rate equals the false-negative rate. We choose EER as the threshold because it summarises overall ASD performance. Given λ as the value at which the EER occurs, and s_i as the score output by the anomaly detector for a test datum x_i, this decision can be can be expressed as follows:

$$L = \mathbb{1}[s_i \geq \lambda] \tag{1}$$

The SLIME region of Figure 1 (denoted by green dashes) illustrates our workflow. We run adapted SLIME to conduct a post-hoc evaluation on the test splits and by section. We segment the spectrograms by frequency as section 2.3 indicates the frequency distributions are relatively stable across time for the subsets. The time domain is not in scope for our experiments. As SLIME's outputs are sensitive to its training hyperparameters [9, 21], we divide the input spectrograms into eight fixed interpretable components and set the number of synthesised samples to 1,000. Figure 4 shows median heatmaps for *Gearbox* and *ToyCar*, split by condi-



(a) 400Hz low-pass filter. AUC: 46.6%

(b) No filters applied. AUC: 86.7%

(c) 400Hz high-pass filter. AUC: 93.0%

Figure 5: Qualitative visualisations of *Gearbox* Source Section 1. Top: Histograms of normalised (cosine) k-nearest neighbour similarities. Bottom: t-SNE scatter plots. Applying a low-pass filter (left) decreases separability between inliers and anomalies while applying a high-pass filter increases separability and improves inlier generalisation, as evidenced through the change in scoring distribution.

tion. The heatmaps show that occluding the low-frequency regions makes inliers appear more anomalous, leading to more incorrect decisions. Correspondingly, occluding the high-frequency regions makes anomalies appear more benign. We observe similar patterns across sections and machines, and on the STFT spectrograms. This behaviour appears to validate the high-frequency hypothesis.

2.5. Re-training the detectors with filtered audio

Our final step to empirically validate the high-frequency hypothesis is to apply low and high-pass filters to the input data, repeat fine-tuning on the ASTs and redo anomaly detection. If high frequencies are important, then we would expect low-pass filters to decrease ASD. We filtered the input spectrograms at both the fine-tuning and anomaly detection stages to prevent discrepancies between the training and test distributions [22]. This intervention removes the possibility that any changes in performance are caused by distribution shifts. In line with previous studies that looked at frequency artefacts in the speech domain [23], we vary the cutoff frequencies between 400 Hz and 8 kHz in increments of 400 Hz for both the low-pass and high-pass filters. We applied the filters directly on the input audio before converting to spectrograms.

Figure 3 depicts our Mel spectrogram results. The low-pass results indicate removing high-frequency information decreases separability between classes, causing fine-tuning classification accuracy and ASD performance across subsets to drop significantly. Separability between classes is still good after applying high-pass filters, and so downstream ASD performance is retained or even improves. We note that classification accuracy drops from the baseline fine-tuning accuracy of 94% (Section 2.2), which suggests separability between classes is an important but not necessary component for ASD. Overall, our results illustrate that high frequencies are helpful for ASD, except for *Fan*, where informative frequencies likely lie in sub-bands. Although *Fan* exhibits similar trends to the other machines, applying high-pass filters causes performance to drop significantly compared to the benchmark AST. We show Figure 3's results qualitatively by visualising the change in normalised k-NN (cosine) similarities and t-SNE [24] on the 768-dimensional learnt embeddings. Figure 5 illustrates an example on *Gearbox*. Aggressive low-pass filtering decreases separability between inliers and anomalies while high-pass filtering increases the dissimilarity. We also visualise a similar phenomenon for the other machines.

3. CONCLUSION AND FUTURE WORK

We test if higher frequencies aid ASD in three ways: (1) by varying the input spectrograms used to train the anomaly detector, (2) evaluating the heatmaps produced by SLIME, and (3) re-training the anomaly detectors after filtering frequencies of differing magnitudes. Our results corroborate the high-frequency hypothesis and suggest high-pass filtering to remove noisy information is a simple pre-processing step to boost performance. The scope of adapted SLIME in our experiments was isolated to frequency segmentations. Some avenues for future work include analysing changes in ASD after segmenting input spectrograms by time, investigating SLIME outputs on alternative anomaly detection architectures, or using more sophisticated thresholds in place of EER. Furthermore, the exceptional results on *Fan* suggest that some anomalies may manifest in sub-band frequencies. Future work could extend the re-training approach by using band-pass filters, which could provide a more fine-grained analysis of anomalous artefacts.

4. ACKNOWLEDGMENT

This project made use of time on Tier 2 HPC facility JADE2, funded by EPSRC under grant EP/T022205/1.

5. REFERENCES

[1] http://dcase.community/.

[2] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 186–190.

[3] J. Lopez, G. Stemmer, and P. Lopez-Meyer, "Ensemble of complementary anomaly detectors under domain shifted conditions," DCASE2021 Challenge, Tech. Rep., July 2021.

[4] K. Morita, T. Yano, and K. Tran, "Anomalous sound detection using cnn-based features by self supervised learning," DCASE2021 Challenge, Tech. Rep., July 2021.

[5] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International Conference on Machine Learning (ICML)*, 2020, pp. 3247–3258.

[6] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, *et al.*, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8340–8349.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[8] Y. Liu, J. Guan, Q. Zhu, and W. Wang, "Anomalous sound detection using spectral-temporal information fusion," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 816–820.

[9] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*. ISMIR, Oct 2017, p. 537–543.

[10] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," *arXiv preprint arXiv:2106.02369*, 2021.

[11] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, "MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions," *arXiv e-prints: 2006.05822*, 2021.

[12] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Proceedings of Interspeech 2021*, 2021, pp. 571–575.

[13] D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song, "Pretrained transformers improve out-of-distribution robustness," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 2744–2751.

[14] S. Fort, J. Ren, and B. Lakshminarayanan, "Exploring the limits of out-of-distribution detection," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

[15] K. T. Mai, T. Davies, and L. D. Griffin, "Self-supervised losses for one-class textual anomaly detection," *arXiv preprint arXiv:2106.02369*, 2022.

[16] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[17] Y.-Y. Yang, M. Hira, Z. Ni, A. Astafurov, C. Chen, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Hwang, J. Chen, P. Goldsborough, S. Narenthiran, S. Watanabe, S. Chintala, and V. Quenneville-Bélair, "Torchaudio: Building blocks for audio and speech processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6982–6986.

[18] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *5th International Conference on Learning Representations (ICLR)*, 2017.

[19] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[20] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevichmorozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, D. Hereñú, F.-R. Stöter, P. Friesch, A. Weiss, M. Vollrath, T. Kim, and Thassilo, "librosa/librosa: 0.8.1rc2," May 2021.

[21] S. Mishra, E. Benetos, B. L. Sturm, and S. Dixon, "Reliable local explanations for machine listening," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020.

[22] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 9737–9748.

[23] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "An explainability study of the constant q cepstral coefficient spoofing countermeasure for automatic speaker verification," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 333–340.

[24] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.