MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Disentangled surrogate task learning for improved domain generalization in unsupervised anomolous sound detection

Venkatesh, Satvik; Wichern, Gordon; Subramanian, Aswin Shanmugam; Le Roux, Jonathan

TR2022-092 August 06, 2022

Abstract

We present our submission to the DCASE2022 Challenge Task 2, which focuses on domain generalization for anomalous sound detection. We investigated a novel multitask learning framework that disentangles domain shared features and domain-specific features. Disentanglement leads to better latent features and also increases flexibility in post-processing due to the availability of multiple embedding spaces. Our disentangled model obtains an overall harmonic mean of 74.57% on the development set, surpassing the MobileNetV2 baseline, which obtains 56.01%. Lastly, we explore the use of machine-specific loss functions and domain generalization methods, which improves our overall performance to 76.42%.

DCASE2022 Challenge

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

DISENTANGLED SURROGATE TASK LEARNING FOR IMPROVED DOMAIN GENERALIZATION IN UNSUPERVISED ANOMALOUS SOUND DETECTION

Technical Report

Satvik Venkatesh[1,2]*, Gordon Wichern*[1]*, Aswin Subramanian*[1]*, Jonathan Le Roux*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2]Interdisciplinary Centre for Computer Music Research, University of Plymouth, UK
satvik.venkatesh@plymouth.ac.uk,{wichern,subramanian,leroux}@merl.com

ABSTRACT

We present our submission to the DCASE2022 Challenge Task 2, which focuses on domain generalization for anomalous sound detection. We investigated a novel multi-task learning framework that disentangles domain-shared features and domain-specific features. Disentanglement leads to better latent features and also increases flexibility in post-processing due to the availability of multiple embedding spaces. Our disentangled model obtains an overall harmonic mean of 74.57% on the development set, surpassing the MobileNetV2 baseline, which obtains 56.01%. Lastly, we explore the use of machine-specific loss functions and domain generalization methods, which improves our overall performance to 76.42%.

***Index Terms*—** Anomaly detection, disentanglement, multi-task learning, domain generalization

1. INTRODUCTION

Machine condition monitoring using acoustic sensors is an important topic for industry with applications such as factory automation and predictive maintenance. Automatic detection of anomalous sounds is a particularly important application, however, all possible types of anomalous sounds may not be known in advance, and purposefully damaging machinery to collect anomalous sound recordings is undesirable. Thus, there has been much recent research interest in the field of unsupervised anomalous sound detection, where only data collected under normal operating conditions is available for training machine learning models.

Much of the recent progress in unsupervised anomalous sound detection has been driven by DCASE challenges on the topic [1–3]. Typical approaches include those based on autoencoder-like architectures [4–10], where a model trained only on normal data to reconstruct its input should exhibit large reconstruction error when presented with an anomalous example at inference time. Another class of approaches, which we refer to as surrogate task models, use an alternative supervised training task to learn a model of normality, and then measure deviations from normal to predict anomalies. Example surrogate tasks include outlier exposure [6, 11], predicting metadata (e.g., machine instance) or attributes (e.g., operating load) [12–14], and learning to predict what augmentations (e.g., time-stretching or pitch-shifting) were applied to an audio clip [15].

As in many areas where deep learning-based models have become the predominant approach, unsupervised anomalous sound detection suffers from issues related to robustness. To better tackle such issues, the anomalous sound detection tasks of both the 2021 and 2022 DCASE challenges focused on performance under domain shift, where acoustic conditions differ based on environmental background noise or other machine operating conditions. The goal is to develop methods that should perform equally well in a source domain, where most of the (normal) training data comes from, and in a target domain, where only a few normal examples are available. The 2021 challenge task [2] assumed the domain (source or target) of the audio sample was known at inference time (a configuration referred to as domain adaptation), while the 2022 task [3] assumes the domain is unavailable at inference time (a configuration referred to as domain generalization).

While many well-known techniques exist for domain generalization (see [16] for an overview), we focus our efforts on disentangled representation learning [17], where subsets of learned feature dimensions correspond to specific factors in the dataset. Disentanglement has been successfully applied for music information retrieval in the audio domain [18] and in approaches to domain adaptation for image classification [19]. Specifically, we consider learning feature representations for each normal sound example in the training set, where subsets of features are learned using different surrogate tasks. In the case of the DCASE 2022 Task 2 dataset, we learn a subset of *domain-shared* features, whose surrogate task is to predict the section index regardless of domain (each section is dedicated to a specific type of domain shift, with other conditions being shared across domains), and subsets of *domain-specific* features each associated with a surrogate task consisting of predicting a particular machine attribute (e.g., specific states or environmental conditions of the machine), which are typically different across domains and sections. We demonstrate experimentally that our disentangled model performs better than a multi-task learning model where features are not disentangled, and further show that by weighting individual anomaly scores computed over different disentangled dimensions, we obtain an ensemble-like system using a single model. Furthermore, by examining the anomaly score in specific disentangled dimensions, we can better understand what may have caused an anomaly based on the attributes with high anomaly scores, helping to improve upon the lack of explainability present in many modern deep learning models.

While our proposed disentangled model outperforms the challenge baselines, we found that it was not optimal for all seven machines on the DCASE 2022 Task 2 dataset. For this reason, we also explore machine-specific variations to the loss function, and ensemble it with an autoencoder based on the attentive neural process [10] in a subset of our submissions.

This work was performed while S. Venkatesh was an intern at MERL.

2. DISENTANGLED ANOMALY DETECTOR

In this paper, we investigate an approach that disentangles a learned latent representation into domain-shared and domain-specific features for domain generalisation in anomalous sound detection, as illustrated in Fig. 1. In particular, we refer to sections as domain-shared features and to attributes as domain-specific features. For example, in Fan's *section 00*, machine noises occurring in the source domain are of type W and X, while those occurring in the target domain are of type Y and Z. Therefore, *section 00* is common to both domains but the machine noises are different across domains.

2.1. Surrogate Task Training

During training, we have a dataset of N normal training examples for a given machine type, $\mathcal{D} = \{(X^{(n)}, y^{(n)})\}_{n=1}^{N}$, where $X \in \mathbb{R}^{F \times T}$ is a magnitude spectrogram with F frequencies and T time frames, and $y = [y_s, y_{a_1}, \ldots, y_{a_M}] \in \mathbb{N}^{M+1}$ is a vector of categorical surrogate task labels, where y_s represents machine section and y_{a_m} represents the categorical label of the m-th attribute among the M different attributes available for the given machine type. We obtain a domain-shared (section) embedding z_S and a domain-specific (attribute) embedding z_A as:

$$z_S = \mathrm{L}^{Sec}[\mathrm{CNN}(X)] \in \mathbb{R}^{D_S}, \quad z_A = \mathrm{L}^{Att}[\mathrm{CNN}(X)] \in \mathbb{R}^{D_A} \tag{1}$$

where $\mathrm{CNN}(\cdot)$ is a shared convolutional neural network, while L^{Sec} and L^{Att} represent section and attribute specific linear embedding layers, respectively (implemented as 1×1 convolutions). All parameters are trained by minimizing $\mathcal{L} = \mathcal{L}^{Sec} + \mathcal{L}^{Att}$, where

$$\mathcal{L}^{Sec} = \log \frac{\exp(w_{0,y_s} \cdot z_S + b_{0,y_s})}{\sum_{c=1}^{C} \exp(w_{0,c} \cdot z_S + b_{0,c})}, \tag{2}$$

$$\mathcal{L}^{Att} = \sum_{m=1}^{M} \log \frac{\exp(w_{m,y_m} \cdot z_A + b_{m,y_m})}{\sum_{c_m=1}^{C_m} \exp(w_{m,c_m} \cdot z_A + b_{m,c_m})} \tag{3}$$

are the cross-entropy losses for section and attributes, respectively, $w_{i,j}$ and $b_{i,j}$ are learned weight vectors and biases of the associated classifiers, c indexes the $C = 6$ sections and c_m indexes the C_m values of the m-th attribute. Because not all attributes are present among all audio examples of a given machine type in the DCASE 2022 Task 2 dataset, the attribute loss in (3) is combined over all attributes in a multi-task learning fashion from the same embedding z_A, rather than learning disentangled feature dimensions for each attribute. If an attribute is unknown for an audio example, the corresponding term in the sum of (3) is ignored.

We note that our formulation of attribute learning in (3) as a multi-task learning problem with a different objective for each attribute differs from [20] where every possible combination of section and attribute corresponded to a different class.

2.2. Inference Approaches

The nearest neighbor (NN) algorithm is a simple and effective approach for anomaly detection [21, 22] given feature vectors of normal samples. As illustrated in Fig. 1, during inference we use the NN distance between a test embedding z_q and all corresponding training set embeddings $z_q^{(j)}$ for computing an anomaly score, i.e.,

$$D_{\mathrm{NN}}(z_q, \mathcal{D}) = \min_{j \in \mathcal{D}} D_{\cos}(z_q, z_q^{(j)}), \tag{4}$$

where $D_{\cos}(\cdot, \cdot)$ is the cosine distance between two embedding vectors. The disentangled model allows us to explore multiple inference approaches depending on which embedding dimensions we use for z_q in (4) as discussed below.

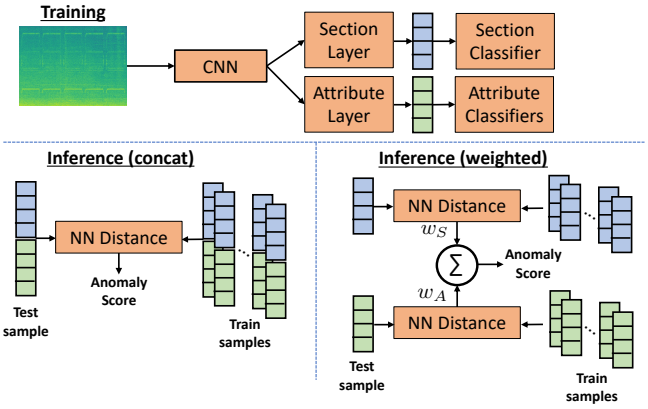

Figure 1: Block diagram of disentangled anomaly detector. In the figure, NN stands for Nearest Neighbor. In the training phase, exclusive latent spaces were assigned to sections and attributes.

Disentangled Concatenated: Use the concatenated embedding $z_C = [z_S^T, z_A^T]^T$ in (4) as shown in the bottom-left of Fig. 1.

Disentangled Weighted: As illustrated in the bottom-right of Fig. 1, we take a weighted average of NN distances separately computed for section embedding z_S and attribute embedding z_A, i.e.,

$$D_{\mathrm{NN}}^{\mathrm{wt}}(z_S, z_A, \mathcal{D}) = w_S D_{\mathrm{NN}}(z_S, \mathcal{D}) + w_A D_{\mathrm{NN}}(z_A, \mathcal{D}) \tag{5}$$

where w_S and w_A are scalar weights, which are optimized after training is complete based on dev set performance. The best weights for each machine are shown in Table 2.

Disentangled Sections: Use only section embedding z_S in (4).

Disentangled Attributes: Use only attribute embedding z_A in (4).

At test time, the section label of the test sample is known, therefore, we limit the training set samples from $\mathcal{D}$ when computing the NN distance to be only those samples belonging to the appropriate section. Furthermore, our CNN architecture, detailed in Section 3.3, operates on spectrogram chunks of $T = 32$ time frames ($\sim$1 s), while each test sample is 10 s long. Using a chunk hop size of one frame, we obtain 282 embedding vectors per 10 s audio file. Following [21], we merged the embedding vectors for each sample by calculating their mean, except for valve where merging based on standard deviation provided significant gains. We then use the merged embedding vectors for computing the anomaly score.

3. EXPERIMENTAL SETUP

3.1. Dataset

There are seven different machine types in the DCASE 2022 Task 2 dataset [3] — ToyCar, ToyTrain, Bearing, Fan, Gearbox, Slider, and Valve. ToyCar and ToyTrain are from the ToyADMOS2 dataset [23], and the five other machines are from the MIMII DG dataset [24]. The data under each machine type is divided into sections, each of which corresponds to a specific type of domain shift. For example, in Fan, *section 00* refers to different machine noise between source and target domains, while *section 01* refers to different factory noise. There are three sections of data in the development training set, and three additional sections in the additional training data, which was released one month after the development set.

For each audio file, information about its section as well as one or more attributes is given. For machines belonging to the MIMII DG dataset [3], only information on the domain shifting attribute, such as the type of machine noise in Fan's *section 00* and the type of factory noise in Fan's *section 01*, was present. For ToyCar and ToyTrain, which belong to the ToyADMOS2 dataset [23], information on all attributes was present in the filenames, even for those attributes that are not the domain shifting one. For the multi-task attribute learning (3), we make use of all present attributes, and represent them as categorical variables using all possible values found in the training set.

3.2. Audio Features and Training Strategy

The sampling rate of all audio files in the dataset was 16 kHz. The duration of each audio file is 10 s. We adopted short-time Fourier transform magnitude spectrograms as features for the neural network. The hop size was set to 32 ms and the window size was 128 ms (2048 samples). While training the neural network, the number of time steps for each audio example was 32 frames. Therefore, the input shape for the network was 1025×32. We adopted the following training pipeline. One epoch is defined as training the network on all 6000 audio files (six sections with 1000 examples in each section). For each audio file, a random chunk of 32 frames is selected for training. The advantages of this technique were reduced RAM usage, less chance of overfitting within epochs, and improved generalisation compared with the baseline.

We adopted the Adam optimizer using a batch size of 32 to train all our systems. In most cases, the learning rate was set to 10^{-4}. For ToyCar, we found a minor improvement by setting it to 10^{-5}. As our definition of an epoch is different from the baseline's definition, we had to tune the number of iterations in our training procedure. Therefore, we saved the model's weights every 5 epochs and tested the anomaly detector's performance on the development set. We trained the models for a maximum of 300 epochs. We were unable to observe a clear relationship between the performances on the surrogate task and detection of anomalies. For instance, an improvement in the classification accuracy of sections (the surrogate task) was not necessarily accompanied by an improvement in anomaly detection. A similar observation has been made by previous studies using autoencoder-based models [10].

3.3. Neural Network Architecture

Morita et al. [21] found that the MobileFaceNet architecture [25] performed better than MobileNetV2 [26] as a feature extractor. We observed a similar improvement in initial experiments, and hence adopted MobileFaceNet. The parameter settings for MobileFaceNet can be found in Table 1. The output of the global depth-wise convolution (GDC) layer is a 512-D embedding vector. This is connected to the linear embedding layers (i.e., 1x1 convolutions) L^{Sec} and L^{Att} defined in Section 2, and associated softmax classification layers. Additionally, we explore minor modifications to the embedding and softmax layers as explained in Section 4.1.

3.4. Evaluation Metrics

We evaluate our models independently for each section and machine type using the three official metrics [3]: area under the ROC curve in the source (AUC (S)) and target (AUC (T)) domains, where the normal test samples are compared against anomalies from both domains, along with the domain agnostic partial AUC (pAUC) computed under low false-alarm-rate conditions.

For threshold-dependent metrics, we followed a similar approach to the baseline [3] and assumed the scores follow a gamma distribution. The parameters of the gamma distribution are estimated from the NN anomaly scores computed on the training set samples independently for each section (excluding self neighbors). For five machines, we set the anomaly detection threshold as the 90th percentile of the gamma distribution. For Fan and Bearing, we observed low sensitivity and hence set the threshold to the 60th percentile.

Table 1: The MobileFaceNet architecture. All the convolutions are 2D convolutions and dw-Conv refers to depth-wise convolution. In the network, Linear Conv 1×1 (sec) is connected to Softmax (sec), and Linear Conv 1×1 (att) is connected to the other softmax layers for attributes. For each layer, we also show the expansion factor (**t**), number of channels (**c**), number of repeats (**n**), and stride (**s**). All convolutions excluding the final linear layers use PReLU as the non-linearity. Refer to [25] for more details on the MobileFaceNet.

Input	Operator	t	c	n	s
1x32x1025	Conv 3x3	-	64	1	2
64x16x513	dw-Conv 3x3	-	64	1	1
64x16x513	Bottleneck	2	64	5	2
64x8x257	Bottleneck	4	128	1	2
128x4x129	Bottleneck	2	128	6	2
128x2x65	Bottleneck	4	128	1	2
128x1x33	Bottleneck	2	128	2	1
128x1x33	Conv 1x1	-	512	1	1
512x1x33	Linear GDC 1x33	-	512	1	1
512x1x1	Linear Conv 1x1 (sec)	-	128	1	1
512x1x1	Linear Conv 1x1 (att)	-	128	1	1
128x1x1	Softmax (sec)	-	6	-	-
128x1x1	Softmax (att_1)	-	C_1	-	-
⋮	⋮		⋮		
128x1x1	Softmax (att_M)	-	C_M	-	-

Table 2: S1 to S4 refer to our four submissions in Section 4.3. The best MSL for each machine is indicated under the MSL (S1) column. The ensemble weights (Ens. wt.) of S2 and the disentanglement weights (Disent. wt.) of S4 were calculated via a grid search.

		Ens. wt. (S2)		Disent. wt. (S4)	
Machine	**MSL (S1)**	**MSL**	**ANP**	w_S	w_A
ToyCar	Disentangled	0.60	0.40	0.90	0.10
ToyTrain	MTL	0.70	0.30	0.00	1.00
Bearing	Sections only	1.00	0.00	1.00	0.00
Fan	ArcFace	0.95	0.05	0.15	0.85
Gearbox	Adversarial	0.65	0.35	0.80	0.20
Slider	Disentangled	0.70	0.30	0.90	0.10
Valve	Disentangled	0.80	0.20	0.90	0.10

4. CHALLENGE SPECIFIC IMPROVEMENTS

4.1. Machine-Specific Loss (MSL)

In Section 2, we explained our implementation of disentanglement. However, we also explored other domain generalisation techniques. While they were not as effective as disentanglement in general, they did improve the performance for certain machines. All these techniques use the same underlying architecture explained in Section 3.3, but with modified embedding or classification layers.

Table 3: Results of different models on the development test set. We merge the three metrics and all sections to obtain a single number per machine using the harmonic mean. We also report the harmonic mean across machines and sections for each of the three metrics.

System	ToyCar	ToyTrain	Bearing	Fan	Gearbox	Slider	Valve	AUC (S)	AUC (T)	pAUC	Overall
Ensemble: MSL + ANP	76.43	**59.96**	**73.93**	**68.89**	**85.37**	**85.93**	**95.83**	**87.55**	**73.43**	**70.36**	**76.43**
Machine Specific Loss	76.43	59.37	73.93	68.85	83.03	85.37	95.63	86.78	73.34	69.68	75.93
Disentangled Weighted	**76.95**	59.74	72.07	63.91	81.38	85.14	94.50	86.09	71.65	68.21	74.57
Disentangled Concatenated	76.43	58.67	67.09	63.18	80.99	85.37	95.01	84.61	70.59	67.02	73.34
Disentangled Sections	76.84	56.64	72.07	62.35	81.04	84.84	94.42	86.43	68.64	68.01	73.45
Disentangled Attributes	75.26	59.74	60.82	63.02	78.86	78.88	92.72	82.12	69.31	64.09	71.08
Multi-task Learning	75.61	59.37	68.24	59.14	80.63	83.51	94.49	81.35	70.72	66.83	72.47
Sections ArcFace	72.31	58.09	71.30	68.85	79.37	82.50	92.87	86.50	71.19	66.04	73.62
Sections Softmax	76.20	52.85	**73.93**	64.39	81.43	85.89	90.11	86.05	67.34	67.92	72.82
ANP-Boot	59.84	50.87	55.54	55.31	64.38	64.11	52.63	69.26	50.87	54.24	57.10
AE Baseline	51.06	39.61	54.80	58.54	63.07	57.99	50.59	68.74	41.91	53.76	52.62
MN Baseline	54.23	51.18	59.16	57.21	59.91	50.26	62.42	63.87	50.14	55.69	56.01

ArcFace [27] was shown to improve class separability by adding angular margin to the loss. We investigated this technique's advantage by training on section indices. The feature scale and margin parameters were set to 32 and 0.5 respectively. We found ArcFace did not work well in a multi-task learning setting, probably because all attributes were not present in every example.

Multi-task Learning (MTL): In this framework, the GDC layer from Table 1 is connected to a single 2D convolutional 1×1 layer with 256 channels. In other words, the features are in an entangled latent space.

Adversarial Training: [28] proposed a gradient reversal layer that helps a model perform domain-adversarial training. In such a setting, there is a forward branch, which is a label classifier, and a reverse branch, which is a domain classifier. The GDC layer is connected to the reverse branch through a gradient reversal layer. Training this system had two phases — pre-training and fine-tuning. Pre-training was performed only using the forward branch in a multi-task learning fashion, training on sections and attributes. Subsequently, we cloned the weights of the forward branch to form the reverse branch. For fine-tuning, the reverse was trained only on attributes and the forward was trained on sections and attributes. The motivation behind placing attributes on both branches is to make the network domain-aware and domain-invariant at the same time.

The best performing loss function for each machine type is shown in the "MSL (S1)" column of Table 2.

4.2. Attentive Neural Process (ANP)

The winning team in the DCASE 2021 Task 2 challenge used an ensemble of autoencoder-like and classification-based models [29]. While ensembling models with different machine-specific loss functions did not show much benefit, likely due to the lack of complementary information, we consider our recently proposed ANP for unsupervised anomalous sound detection [10] as a complementary autoencoder-like member of a model ensemble. We use the ANP-Boot two-stage inference configuration, and all hyperparameters are identical to those described in [10], except that we train our models using the DCASE 2022 Task 2 dev and eval training sets. In [10], the flexible inference approach enabled by ANP led to better zero-shot adaptation to new machines, and our hope was that this may also lead to better domain generalization performance.

We ensemble the ANP and MSL systems following the approach in [29, 30] where we first standardize the training set anomaly scores over a section for each model to have zero-mean and unit variance. We then perform a grid search over weighted convex combinations of scores from the different models that perform best on the dev set. The selected ensemble weights for each machine are shown in Table 2.

4.3. Selected Submissions

Below are the four submissions submitted to the DCASE challenge:

- S1 **MSL** as explained in Section 4.1
- S2 **Ensemble: MSL + ANP** as explained in Section 4.2.
- S3 **Disentanglement Concatenated** as explained in Section 2.2.
- S4 **Disentanglement Weighted** as explained in Section 2.2.

5. RESULTS AND CONCLUSION

Table 3 shows the results of all our models. Training using only section labels obtains an overall harmonic mean of 72.82%, which is significantly higher than both the baselines. This improvement is attributed to adopting the Nearest Neighbor algorithm during post-processing [21] and to our new training strategy explained in Section 3.2. Adopting ArcFace, which is essentially training on section indices with additive angular margin, improved the overall performance to 73.62%, while the AUC(T) improved from 67.34% to 71.79%. Multi-task learning, which trains on sections and attributes, obtained a lower overall performance of 72.47%, but improved the AUC(T) to 70.72%. Note that the multi-task learning model does not use ArcFace. Disentangled Sections only considers the section embeddings during inference and obtains an overall performance of 73.45%. Although the overall performance is lower than that of ArcFace, it showed improvements for all machines except Bearing and Fan. The Disentangled Weighted model obtains the highest overall performance for a single model without ensembling and machine-specific losses. The MSL system obtains an overall performance of 75.93%, which shows that different domain generalisation techniques are useful for different machines. Finally, combining MSL and ANP obtains the highest performance, with 76.43%. This is interesting because ANP by itself only obtains an overall performance of 57.10%. This demonstrates that ensembling models that contain complementary information is helpful for anomaly detection.

6. REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, Nov. 2020, pp. 81–85.

[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 *Proc. DCASE*, Nov. 2021, pp. 186–190.

[3] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," *In arXiv e-prints: 2206.05876*, 2022.

[4] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Proc. ICASSP*, Apr. 2015.

[5] E. Cakır and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection," in *Proc. DCASE*, Nov. 2017.

[6] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 1, pp. 212–224, 2018.

[7] T. Hayashi, T. Komatsu, R. Kondo, T. Toda, and K. Takeda, "Anomalous sound event detection based on wavenet," in *Proc. EUSIPCO*, Sept. 2018, pp. 2494–2498.

[8] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *Proc. ICASSP*, May 2020, pp. 271–275.

[9] R. Giri, F. Cheng, K. Helwani, S. V. Tenneti, U. Isik, and A. Krishnaswamy, "Group masked autoencoder based density estimator for audio anomaly detection," in *Proc. DCASE*, Nov. 2020, pp. 51–55.

[10] G. Wichern, A. Chakrabarty, Z.-Q. Wang, and J. Le Roux, "Anomalous sound detection using attentive neural processes," in *Proc. WASPAA*, 2021, pp. 186–190.

[11] P. Primus, V. Haunschmid, P. Praher, and G. Widmer, "Anomalous sound detection as a simple binary classification problem with careful selection of proxy outlier examples," in *Proc. DCASE*, Nov. 2020, pp. 170–174.

[12] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proc. DCASE*, Nov. 2020, pp. 46–50.

[13] J. A. Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer, and J. Huang, "A speaker recognition approach to anomaly detection," in *Proc. DCASE*, Nov. 2020, pp. 96–99.

[14] K. Wilkinghoff, "Sub-cluster adacos: Learning representations for anomalous sound detection," in *Proc. IJCNN*, 2021, pp. 1–8.

[15] T. Inoue, P. Vinayavekhin, S. Morikuni, S. Wang, T. H. Trong, D. Wood, M. Tatsubori, and R. Tachibana, "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *Proc. DCASE*, Nov. 2020, pp. 66–70.

[16] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu, "Generalizing to unseen domains: A survey on domain generalization," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[17] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 830–838.

[18] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Metric learning vs classification for disentangled music representation learning," in *Proc. ISMIR*, 2020.

[19] Z. Ding and Y. Fu, "Deep domain generalization with structured low-rank constraint," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 304–313, 2017.

[20] K. Wilkinghoff, "Utilizing sub-cluster adacos for anomalous sound detection under domain shifted conditions," DCASE2021 Challenge, Tech. Rep., July 2021.

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

[22] M. Jones, D. Nikovski, M. Imamura, and T. Hirata, "Exemplar learning for extremely efficient anomaly detection in real-valued time series," *Data mining and knowledge discovery*, vol. 30, no. 6, pp. 1427–1454, 2016.

[23] 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.

[24] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "Mimii dg: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," *arXiv preprint arXiv:2205.13879*, 2022.

[25] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Chinese Conference on Biometric Recognition*. Springer, 2018, pp. 428–438.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[27] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[28] 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.

[29] J. A. Lopez, G. Stemmer, P. Lopez Meyer, P. Singh, J. Del Hoyo Ontiveros, and H. Cordourier, "Ensemble of complementary anomaly detectors under domain shifted conditions," in *Proc. DCASE*, Nov. 2021, pp. 11–15.

[30] P. Daniluk, M. Gozdziewski, S. Kapka, and M. Kosmider, "Ensemble of auto-encoder based systems for anomaly detection," DCASE2020 Challenge, Tech. Rep., July 2020.