SELF-SUPERVISED REPRESENTATION LEARNING FOR UNSUPERVISED ANOMALOUS SOUND DETECTION UNDER DOMAIN SHIFT

Han Chen[1], *Yan Song*[1], *Li-Rong Dai*[1], *Ian McLoughlin*[1,2], *Lin Liu*[3]

[1]National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China.
[2]ICT Cluster, Singapore Institute of Technology, Singapore.
[3]iFLYTEK Research, iFLYTEK CO., LTD, Hefei, China.

ABSTRACT

In this paper, a self-supervised representation learning method is proposed for anomalous sound detection (ASD). ASD has received much research attention in recent DCASE challenges. It aims to identify whether a sound emitted from a machine is anomalous or not, given only normal sound data. This is a challenging task due to highly variable time-frequency characteristics of sounds from different machine types, and the fact that many attributes affect machine without being anomalous. This is especially true for domain shift tasks, where only a few training sound clips are available. From the perspective of self-supervised learning, each given sound clip can be considered as a transformation of an original clean sound, where the attribute of each clip may indicate different supervision signals. We propose a unified representation learning framework, equipped with a time-frequency attention mechanism, to perform ASD for different machine types and attributes. For domain shift, a centre imprinting method, which directly sets centres for target domain attributes, is presented. This provides immediate good representation and an initialization for further fine-tuning. Evaluation on DCASE2021 ASD task demonstrates the effectiveness of the proposed method.

Index Terms— anomalous sound detection, representation learning, self-supervised learning

1. INTRODUCTION

Unsupervised anomalous sound detection (ASD) has attracted significant research interest due to its wide application in machine condition monitoring. Since 2020, DCASE challenges include ASD as a new task, which aims to identify whether the sound emitted from a target machine is anomalous or not, given only normal sound training samples. Anomalous sounds rarely occur and are highly diverse, which makes ASD significantly different to sound event detection, while unsupervised ASD is very different to supervised ASD. Furthermore, different machines can emit sounds with various time-frequency characteristics, making it challenging to exploit an appropriate model based on normal sound data. In DCASE2021, an additional difficulty was introduced by providing only a small number of normal training sound clips for target domain.

Many methods have recently been applied to ASD, which mainly focus on exploiting machine learning methods to model the distribution of normal sound data. This includes generative methods such as Autoencoder [1], WaveNet [2], GAN [3, 4] and flow [5], plus several classification based methods [6, 7, 8, 9, 10, 11, 12, 13, 14], which take advantage of advanced convolutional neural networks (CNN), like ResNet [15] and MobileNetV2 [16]. The anomaly scores can be calculated based on the reconstruction error, log-likelihood or classification confidence of the observed sound. Performance can be improved by using an ensemble of generative and discriminative models, or models with different configurations (*e.g.* input features, architectures) [7, 8, 9, 11, 12, 13, 14]. In addition, several domain adaptation methods [17, 18, 19, 20] have been proposed to reduce the performance gap between source and target domain. The key to the success in ASD is to learn a compact and discriminative representation for sounds from different machine types and attributes.

From the perspective of self-supervised learning, the given sound clips can be considered as transformations of original clean sounds, where the attributes of each clip may indicate different supervision signals. In this paper, we propose a unified self-supervised representation learning framework for both source and target domain for all provided machine types and attributes, as shown in Fig. 1. Specifically, a time-frequency attention pooling (TF_attention) mechanism is combined with the ResNet feature extractor to introduce a weighted sum vector along time and frequency axes, to effectively learn a unified representation space. Moreover, to facilitate the anomaly score computation, length normalization is utilized for cosine similarity measures for both training and test. Training sound samples are organized in sections which may involve one or more attributes [14]. To cater for this, we propose a mixup_intype method, motivated by mixup [21] and manifold mixup [22], to improve the modeling capability for each machine type. As for the domain shift test, we propose a centre imprinting method under the unified representation framework. Specifically, given the additive target domain data, the centres can be directly set by averaging the activations of the feature extractor, which can be seen as ASD prototypes, or be further adaptively fine-tuned. Extensive experiments on task2 of DCASE2021 demonstrates the advantage of the proposed method. For domain-shift evaluation, further ablation experiments are conducted to evaluate the effectiveness of centre imprinting.

2. DETAILED DESCRIPTION OF PROPOSED METHOD

The overall structure of our method is shown in Fig. 1. We assume that sound of each type of machine has a clean mode, which contains essential information that differentiates it from other types of machine. Sounds from the same type of machine but with different attributes, whether in the source domain or in the target domain, are all transformations of the original clean sound. The main difference between domains is that there are very few samples in the target do-

Yan Song is the corresponding author. This work was supported by the Leading Plan of CAS (XDC08030200).

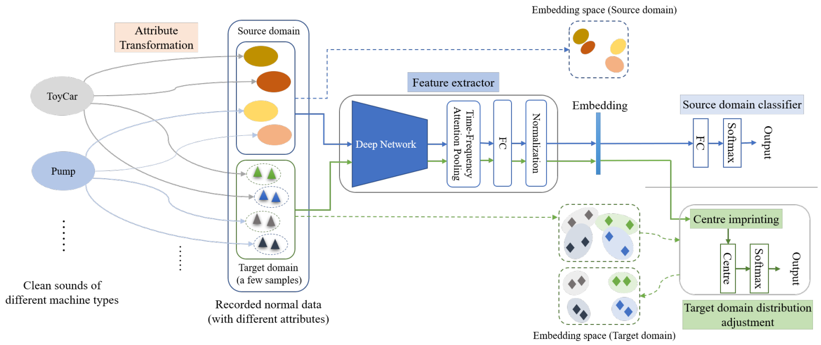
Embedding space (Source domain)
Attribute Transformation
Source domain
ToyCar
Pump
Feature extractor
Deep Network
Time-Frequency Attention Pooling
FC
Normalization
Embedding
Source domain classifier
FC
Softmax
Output
Centre imprinting
Centre
Softmax
Output
Target domain (a few samples)
Clean sounds of different machine types
Recorded normal data (with different attributes)
Target domain distribution adjustment
Embedding space (Target domain)

Fig. 1. The overall structure of our method. The source domain data is used to train the feature extractor. A few samples from the target domain are used to imprint the proxy class centre and adjust the distribution of the target domain. The solid line with arrow indicates the data calculation process, and the dotted line with arrow indicates the change of embedding distribution.

main.

Firstly, we use all the source domain data to train the feature extraction network, from which an effective embedding can be learned by distinguishing different attribute transformations. The model f maps each input sample x into a fixed dimension embedding vector $f(x)$ with unit length, *i.e.* $||f(x)||_2$=1. Cosine distance is used to measure the distance between the test sample and the class centres. The embedding is also normalized in training, so as to keep the optimization goal consistent with the distance measurement during testing.

Since there are not enough samples in the target domain to participate in the training of the feature extractor, the real distribution of the target domain in the embedded space may be scattered, or even overlap between classes. After the feature extractor is trained, the embedding vectors of the target domain are used to imprint the proxy class centres and adjust the distribution through fine-tuning.

2.1. Self-supervised Learning

2.1.1. Data Augmentation

Mixup [21] and Manifold Mixup [22] generate a large number of virtual samples to improve the generalization performance of a model. According to our view of attribute transformation, interpolation between the features within the same type of machine can produce richer and more meaningful transformations. When training with all the source domain data, we shuffle the data and corresponding attribute labels in each mini-batch, and then group the sample and label sequences before and after shuffling according to the machine type, and finally linearly interpolate the grouped sample sequences and attribute label sequences:

$$\begin{aligned} \widetilde{x} &= \lambda * x_i + (1-\lambda) * x_j \\ \widetilde{y} &= \lambda * y_i + (1-\lambda) * y_j \end{aligned} \tag{1}$$

where mixing coefficient $\lambda \sim \text{Beta}(\alpha, \alpha)$. Obviously, this interpolation method can be used in both input or hidden layers, we refer to them as mixup_intype and manifold mixup_intype respectively.

2.1.2. Time-frequency Attention Pooling

Since different machine sounds have different characteristics, it is often necessary to use different feature extraction structures to obtain respective appropriate representations. In order to alleviate this problem, we introduce the time-frequency attention mechanism to unify the feature extractor. We keep the temporal dimension unchanged in the network, and aggregate the frame-level features in different frequency bands into utterance-level representations with attention. Each map $\mathbf{X} \in \mathbf{R}^{T\times D}$ (T: number of frames, D: feature dimension) on the frequency axis of the final feature maps before aggregation contains the information within a certain frequency range in the original spectrum. We feed each $\mathbf{X}$ into a 1×1 convolutional layer followed by softmax non-linear activation along the temporal axis to get the attention map $\mathbf{A} \in \mathbf{R}^{T\times K}$($K$: number of attention heads). The weighted first-order statistic μ and second-order statistic σ of each $\mathbf{X}$ can then be calculated similar to bilinear pooling [23]:

$$\begin{aligned} p &= sign(vec(\mathbf{X}^\top\mathbf{A}))\sqrt{|vec(\mathbf{X}^\top\mathbf{A})|} \\ \mu &= p/||p||_2 \\ q &= vec((\mathbf{X}\odot\mathbf{X})^\top\mathbf{A}) - vec(\mathbf{X}^\top\mathbf{A})\odot vec(\mathbf{X}^\top\mathbf{A}) \\ q &= sign(q)\sqrt{|q|} \\ \sigma &= q/||q||_2 \end{aligned} \tag{2}$$

where $vec(x)$ reshapes x into a vector, $sign(x)$ obtains the sign of x and $\odot$ represents the Hadamard product. The output of the time-frequency attention pooling layer is given by concatenating μ and σ of all $\mathbf{X}$. As we can see, the proposed time-frequency attention pooling can endow different attention weights to different frequency bands in each frame.

2.2. Centre Imprinting

The proposed method unifies the test process of the source and target domains. In the source domain, we use the embeddings of a large number of training set samples to calculate the centre of each class. Our approach, we name *centre imprinting*, directly determines the proxy centres based on the extremely limited samples of the target domain. Although the proxy centre only carries the semantic information of a few samples, the test samples from the target domain should contain more semantic similarity to the corresponding proxy centre than to other class centres. Based on this consideration, we derive an approach to adjust the embedding space distribution of the target domain.

Calculate proxy centre – We average embedding vectors for each class k in the target domain as the proxy centre C_k,

$$C_k = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{3}$$

Fine-tuning – The probability distribution over target classes for a sample x can be calculated based on softmax,

$$p(y=k|x) = \frac{exp(C_k^\top \cdot f(x))}{\sum_{i=1}^{K} exp(C_i^\top \cdot f(x))} \tag{4}$$

and then the corresponding loss function can be written as,

$$Loss = -log\frac{exp(C_k^\top \cdot f(x))}{\sum_{i=1}^{K} exp(C_i^\top \cdot f(x))} \tag{5}$$

which is similar to softmax cross-entropy loss. The embedding space of the target domain is adjusted by minimizing this loss function with SGD as used for parameter optimization, and the proxy centres are updated for each iteration. The fine-tuning process should make the target domain tend to a unimodal distribution and reduce the overlap between classes, so that following fine-tuning, the new proxy centre determined by these few samples becomes a better representation of their class.

3. EXPERIMENTAL EVALUATION

3.1. Dataset

We conducted experiments using the DCASE2021 task2 development dataset [24, 25]. Each recording is a single-channel, 10-second audio clip from one of seven machine types. The dataset consists of three sections (Section 00, 01, 02) for each machine type. The section is a unit for calculating performance metrics. For each section: (1) around 1000 clips of normal training sounds in the source domain and three clips of normal training sounds in the target domain are provided, (2) around 100 clips each of normal and anomalous test sounds in the source and target domain are provided respectively. There are differences in operating speed, machine load, SNR and other attributes between the source domain and the target domain. The training set lists this attribute information. The specific number of attribute classes are presented as *machine type(number of source domain attributes, number of target domain attributes)* as follows: ToyCar(8,9), ToyTrain(8,9), fan(3,3), gearbox(35,9), pump(4,3), slider(9,7), valve(8,6).

3.2. System

Input features are log-Mel scale spectrograms using 128 Mel-scale filters and a window size of 1024 with hop size of 512. In the training stage, 64 frames are randomly cropped from the spectrograms, that is, the input size is 64 × 128. Batch size is 32 and the loss function is softmax cross-entropy loss. When calculating the class centre and testing, we take the complete spectrograms as the input.

The feature extractor shown in Fig. 1 is adapted from the ResNet18 [15] architecture, with the detailed network structure presented in Table 1. The number of attention heads is set to 2. The network is optimized using stochastic gradient descent with momentum of 0.9 and weight decay of 1e-4. Each network is trained for 120 epochs with learning rate gradually reducing from 0.1 to 0.001 (0.1×50 epochs, 0.01×40 epochs, 0.001×30 epochs).

3.3. Evaluation Metrics

Performance metrics are calculated by sections. For each test sample, we calculate the cosine distance between it and the centres of all classes in the corresponding section, and take the negative value of the maximum cosine distance as the anomaly score.

Table 1. Detailed configuration of the feature extraction network.

Layer	Structure	Stride	O/p size
Conv1	$7\times 7,\ 16$	1×1	$T\times 128\times 16$
Res1	$\begin{bmatrix}3\times 3,\ 16\\ 3\times 3,\ 16\end{bmatrix}\times 2$	1×1	$T\times 128\times 16$
Trans1	$3\times 3,\ 32$	1×2	$T\times 64\times 32$
Res2	$\begin{bmatrix}3\times 3,\ 32\\ 3\times 3,\ 32\end{bmatrix}\times 2$	1×1	$T\times 64\times 32$
Trans2	$3\times 3,\ 64$	1×2	$T\times 32\times 64$
Res3	$\begin{bmatrix}3\times 3,\ 64\\ 3\times 3,\ 64\end{bmatrix}\times 2$	1×1	$T\times 32\times 64$
Trans3	$3\times 3,\ 128$	1×2	$T\times 16\times 128$
Res4	$\begin{bmatrix}3\times 3,\ 128\\ 3\times 3,\ 128\end{bmatrix}\times 2$	1×1	$T\times 16\times 128$
Trans4	$3\times 3,\ 128$	1×2	$T\times 8\times 128$
TF_AP	-	-	$1\times(128\times 2K\times 8)$
FC	$(128\times 2K\times 8)\times 128$	-	1×128
L2norm	-	-	1×128

ASD methods are evaluated using the area under the receiver operating characteristic curve (AUC) and the partial-AUC (pAUC), which is defined as the AUC over a low false-positive-rate (FPR) range [0, 0.1] in this task.

3.4. Discussion of Results

To verify the effectiveness of the proposed method, we set up two baseline systems. Baseline 1: DCASE2021 Challenge Task 2 official MobileNetV2 based baseline [14] using section number as the classification label; Baseline2: the original ResNet18 network [15] based model with the same channel change process as in our network, trained to identify the 75 attribute classes of the source domain. Again, testing is performed in the same way as our method. We implemented two systems according to our proposed method, using all the source domain data to classify 75 attribute classes without target domain fine-tuning. One system uses mixup_intype (mixup within same machine type, α=1.0). The second system uses manifold mixup_intype (manifold mixup within same machine type, α=1.5). We also compared with the highest ranked model fusion based systems in the DCASE2021 Challenge Task 2, in which Top1 (XVector1D) [11] is a classification based fusion system of the first ranked team.

Table 3 lists the harmonic mean of the AUC and pAUC results over all machine types, sections, and domains of various methods. Since we only use source domain data to train the model, Table 2 provides the harmonic mean of the AUC and pAUC over all sections of different machine types in the source domain. It can be seen that our method significantly outperforms the baseline systems and is comparable to the highest results obtained by fusing many models at present, indicating clearly that our model has strong representation ability for different machine sounds.

We also compared the overall effects of different feature aggregation methods without using data augmentation in Table 4, in which average pooling and max pooling are widely used in anomalous sound detection task. It can be seen from these results that the introduction of an attention mechanism enables the network to better capture the diverse time-frequency characteristics of different machine sounds.

In Table 5, we study the overall effects of incorporating the proposed mixup_intype and manifold mixup_intype methods, compared

Table 2. Harmonic mean of AUC and pAUC of different methods in the source domain (%). baseline1: MobileNetV2 [14], baseline2: ResNet18, MI: Mixup_intype, MMI: Manifold Mixup_intype, total: the harmonic mean over all the machine types (%).

Type	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve	total
Method	AUC/pAUC	AUC/pAUC	AUC/pAUC	AUC/pAUC	AUC/pAUC	AUC/pAUC	AUC/pAUC	AUC/pAUC
baseline1	55.80/58.64	67.89/51.87	61.02/65.79	70.21/61.45	65.48/59.24	72.67/59.50	55.95/52.17	63.53/58.01
baseline2	84.87/74.64	78.25/60.61	86.24/78.12	78.46/59.18	82.68/68.99	88.93/71.80	72.22/54.33	81.31/65.78
Top2 [12]	91.06/78.25	86.13/65.36	**90.36**/79.66	77.76/**67.67**	82.52/66.50	90.83/83.05	**95.87/84.78**	87.42/**74.24**
Top3 [13]	88.52/71.40	**90.99/66.75**	81.61/73.18	74.57/56.94	83.83/**71.40**	90.99/76.52	88.88/72.00	85.24/69.19
ours(MI)	**92.77/84.00**	88.58/64.29	86.41/**79.86**	**83.18**/60.31	**87.77**/70.25	91.57/**83.70**	88.35/73.28	88.28/72.62
ours(MMI)	92.38/82.50	90.15/66.18	86.05/79.54	82.21/62.80	86.67/70.42	**92.41**/82.71	91.21/79.48	**88.58**/73.99

Table 3. AUC and pAUC harmonic mean of different systems.

Method	AUC (%)	pAUC (%)
baseline (MobileNetV2) [14]	59.72	56.37
baseline (ResNet18)	70.27	59.00
Top1 (XVector1D) [11]	76.62	**69.50**
Top2 [12]	78.05	68.09
Top3 [13]	78.54	63.93
ours (Mixup_intype)	**79.20**	64.42
ours (Manifold Mixup_intype)	78.87	64.71

Table 4. Comparison of the harmonic mean of the AUC and pAUC among different pooling methods without data augmentation.

Method	AUC (%)	pAUC (%)
Average Pooling	72.76	62.38
Max Pooling	72.21	61.00
Time-Frequency Attention Pooling	**75.71**	**63.04**

Table 6. AUC(%) of simulated target domain(Section 00) (n: sampling number). All: training with data of all the three sections.

Type	**Method**	n=1	n=3	n=5	All
ToyCar	w/o fine-tuning	54.59	64.45	66.71	87.44
	w/ fine-tuning	72.12	83.31	82.52	
Gearbox	w/o fine-tuning	71.30	83.46	80.82	88.89
	w/ fine-tuning	72.04	85.54	83.07	
Slider	w/o fine-tuning	67.48	83.82	87.61	94.28
	w/ fine-tuning	64.97	81.07	85.12	

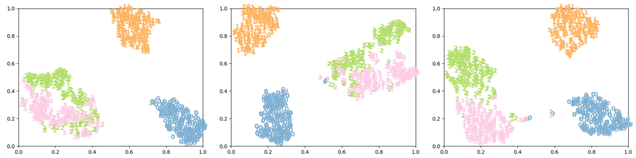

Fig. 2. t-SNE visualization of the embedding space for the remaining data in ToyCar simulation target domain after sampling. Left: no fine-tuning; Middle: fine-tuning (1 sample); Right: fine-tuning (3 samples). Different colored numbers represent different classes.

to vanilla mixup and manifold mixup. While the vanilla versions provide little improvement, it is clear that our proposed methods yield a strong improvement for both. This indicates that the new attribute transformations generated by mixing within the same machine type are more meaningful.

In order to further study domain shift, we carried out simulation experiments. We used the data from Section 01 and 02 in the source domain of each machine type as the simulated source domain(SSD) and the data from Section 00 as the simulated target domain(STD). For convenience, we will examine the simulation experiment results of just ToyCar (from ToyADMOS2 [24]), Gearbox and Slider (from MIMII DUE [25]). We trained a model with the simulated source domain data (around 2000 samples) for each machine type, and randomly selected a few samples (1, 3, 5) from the corresponding simulated target domain as available training data of STD. The AUC of ToyCar, gearbox and slider in the simulated source domain are 93.44%, 74.01% and 83.58% respectively. We use the available training data in the simulated target domain to apply the proposed centre imprinting method. As can be seen in Table 6, for gearbox and slider, when the number of training samples in STD increases to 3 or 5, significant improvement can be obtained by directly calculating the centres, but fine-tuning does not yield much improvement. However, for ToyCar, when the number of samples increases, fine-tuning significantly improves performance in STD compared to that of directly computing centres. Table 6 also provides the test results of Section 00 using data of all the three sections (around 3000 samples) for training, reflecting that centre imprinting greatly mitigates the domain shift by using only a few samples. Fig. 2 examines the embeddings of the remaining data after random sampling in the simulated target domain (Section 00) of ToyCar, revealing that with the increase of the number of samples in the target domain, fine-tuning gradually moves the embedding space to the unimodal distribution, and greatly reduces the class overlap.

Table 5. Harmonic mean of the AUC and pAUC of various data augmentation types (%).

	α=1.0		α=1.5	
Method	AUC	pAUC	AUC	pAUC
Vanilla Mixup	74.25	61.78	74.85	61.83
Vanilla Manifold Mixup	74.19	61.81	75.53	62.02
Mixup_intype	**79.20**	64.42	78.16	64.60
Manifold Mixup_intype	78.65	**64.59**	**78.87**	**64.71**
Without mixing	AUC=75.71		pAUC=63.04	

4. CONCLUSION

This paper has proposed an effective self-supervised representation learning method for anomalous sound detection. For a domain shift task, we alleviate mismatch problems through robust feature learning. Domain shift simulation experiments show that when the number of samples in the target domain increases slightly, domain shift is mitigated by directly calculating the class centres or through our fine-tuning method. Our single model thus achieves good performance on par with state-of-the-art multi-fusion models, and is suitable for domain shift tasks with few samples.

5. REFERENCES

[1] Kaori Suefusa, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, Takashi Endo, and Yohei Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 271–275.

[2] Ellen Rushe and Brian Mac Namee, "Anomaly detection in raw audio using deep autoregressive networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3597–3601.

[3] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, *GANomaly: Semi-supervised Anomaly Detection via Adversarial Training*, Computer Vision, ACCV 2018, 2019.

[4] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye, "Beatgan: Anomalous rhythm detection using adversarially generated time series.," in *IJCAI*, 2019, pp. 4433–4439.

[5] Kota Dohi, Takashi Endo, Harsh Purohit, Ryo Tanabe, and Yohei Kawaguchi, "Flow-based self-supervised density estimation for anomalous sound detection," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 336–340.

[6] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich, "Deep anomaly detection with outlier exposure," *Proceedings of the International Conference on Learning Representations*, 2019.

[7] Ritwik Giri, Srikanth V. Tenneti, Karim Helwani, Fangzhou Cheng, Umut Isik, and Arvindh Krishnaswamy, "Unsupervised anomalous sound detection using self-supervised classification and group masked autoencoder for density estimation," Tech. Rep., DCASE2020 Challenge, July 2020.

[8] Paul Primus, "Reframing unsupervised machine condition monitoring as a supervised classification task with outlier-exposed classifiers," Tech. Rep., DCASE2020 Challenge, July 2020.

[9] Phongtharin Vinayavekhin, Tadanobu Inoue, Shu Morikuni, Shiqiang Wang, Tuan Hoang Trong, David Wood, Michiaki Tatsubori, and Ryuki Tachibana, "Detection of anomalous sounds for machine condition monitoring using classification confidence," Tech. Rep., DCASE2020 Challenge, July 2020.

[10] Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, Toshiki Nakamura, Yuki Nikaido, Ryo Tanabe, Harsh Purohit, Kaori Suefusa, Takashi Endo, Masahiro Yasuda, and Noboru Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, November 2020, pp. 81–85.

[11] Jose Lopez, Georg Stemmer, and Paulo Lopez-Meyer, "Ensemble of complementary anomaly detectors under domain shifted conditions," Tech. Rep., DCASE2021 Challenge, July 2021.

[12] Kazuki Morita, Tomohiko Yano, and Khai Tran, "Anomalous sound detection using CNN-based features by self supervised learning," Tech. Rep., DCASE2021 Challenge, July 2021.

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

[14] Yohei Kawaguchi, Keisuke Imoto, Yuma Koizumi, Noboru Harada, Daisuke Niizumi, Kota Dohi, Ryo Tanabe, Harsh Purohit, and Takashi Endo, "Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions," *In arXiv e-prints: 2106.04492, 1–5*, 2021.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[17] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[18] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo, "Autodial: Automatic domain alignment layers," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[19] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua, "Beyond sharing weights for deep domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 801–814, 2018.

[20] Shuang Li, Chi Liu, Qiuxia Lin, Binhui Xie, Zhengming Ding, Gao Huang, and Jian Tang, "Domain conditioned adaptation network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 11386–11393.

[21] Yann N. Dauphin David Lopez-Paz Hongyi Zhang, Moustapha Cisse, "Mixup: Beyond empirical risk minimization," *International Conference on Learning Representations*, 2018.

[22] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proceedings of the 36th International Conference on Machine Learning*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds., Long Beach, California, USA, 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447, PMLR.

[23] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[24] Noboru Harada, Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Masahiro Yasuda, and Shoichiro Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," *arXiv preprint arXiv:2106.02369*, 2021.

[25] Ryo Tanabe, Harsh Purohit, Kota Dohi, Takashi Endo, Yuki Nikaido, Toshiki Nakamura, and Yohei Kawaguchi, "MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions," *In arXiv e-prints: 2006.05822, 1–4*, 2021.