DESCRIPTION AND DISCUSSION ON DCASE 2021 CHALLENGE TASK 2: UNSUPERVISED ANOMALOUS SOUND DETECTION FOR MACHINE CONDITION MONITORING UNDER DOMAIN SHIFTED CONDITIONS

Yohei Kawaguchi[1]*, Keisuke Imoto*[2]*, Yuma Koizumi*[3]*, Noboru Harada*[4]*, Daisuke Niizumi*[4]*, Kota Dohi*[1]*, Ryo Tanabe*[1]*, Harsh Purohit*[1]*, and Takashi Endo*[1]

[1] Hitachi, Ltd., Japan, yohei.kawaguchi.xk@hitachi.com
[2] Doshisha University, Japan, keisuke.imoto@ieee.org
[3] Google, Japan, koizumiyuma@google.com
[4] NTT Corporation, Japan, noboru.harada.pv@hco.ntt.co.jp

ABSTRACT

We present the task description and discussion on the results of the DCASE 2021 Challenge Task 2. Last year, we organized unsupervised anomalous sound detection (ASD) task; identifying whether the given sound is normal or anomalous without anomalous training data. In this year, we organize an advanced unsupervised ASD task *under domain-shift conditions* which focuses on the inevitable problem for the practical use of ASD systems. The main challenge of this task is to detect unknown anomalous sounds where the acoustic characteristics of the training and testing samples are different, i.e. domain-shifted. This problem is frequently occurs due to changes in seasons, manufactured products, and/or environmental noise. After the challenge submission deadline, we will add challenge results and analysis of the submissions.

***Index Terms*—** anomaly detection, dataset, acoustic condition monitoring, domain shift, DCASE Challenge

1. INTRODUCTION

Anomalous sound detection (ASD) [1–7] is the task of identifying whether the sound emitted from a machine is normal or anomalous. Automatic detection of mechanical failure is an essential technology in the fourth industrial revolution, which includes artificial intelligence (AI)–based factory automation, and also prompt detection of machine anomaly by observing its sounds may be useful for machine condition monitoring.

Last year, we organized "*unsupervised ASD*" as the Task 2 of Detection and Classification of Acoustic Scenes and Events (DCASE) challenge 2020 [8] for connecting academic tasks and real-world problems. The main challenge of this task was to detect unknown anomalous sounds under the condition that only normal sound samples have been provided as training data [1–7]. In real-world factories, actual anomalous sounds rarely occur but are highly diverse. Therefore, exhaustive patterns of anomalous sounds are impossible to collect. This means that we must detect unknown anomalous sounds that were not in the given training data. This unique and real-world oriented task attracted the interest of many participants, and resulting in we received 117 submissions from 40 teams, and several novel approaches have been developed through this challenge [9–12].

In this year, we have organized a follow-up unsupervised ASD task under domain shifted condition, which simulates a more challenging issue in real-world applications. The main challenge of this task is that the acoustic characteristics of training and testing phase are different due to changes in the normal condition such as motor speed and signal-to-noise (SNR). A frequent real-world example of this problem is that the motor speed in a conveyor for transporting products varies in response to product demand; for a product whose demand changes with the seasons, training data recorded in summer was 300–400 rotations per minute (RPM) (i.e. source domain), but the demand drops in the winter resulting in the motor speed decreased to 100–200 RPM (i.e, target domain). Since a normal motor sound at 100 RPM is an unknown sound for the ASD system, it could incorrectly be detected as an anomalous sound. Therefore, methods to deal with such drift in normal conditions are required to accelerate the real-world application of ASD.

As the first benchmark task for domain-shift problems in ASD, we have designed DCASE challenge 2021 Task 2 "*Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring under Domain Shifted Conditions*". The scope includes differences in operating speed, machine load, environmental noise, and so on. After the challenge submission deadline, we will add challenge results and analysis of the submissions.

2. UNSUPERVISED ANOMALOUS SOUND DETECTION UNDER DOMAIN SHIFTED CONDITIONS

Let the L-sample time-domain observation $\boldsymbol{x} \in \mathbb{R}^L$ be an audio clip that includes a sound emitted from a machine. ASD is the determination of whether a machine is in a normal or anomalous state from $\boldsymbol{x}$. To determine the state of the machine, an anomaly score is calculated; it takes a large value when the machine is anomalous, and vice versa. To calculate the anomaly score, we have to prepare an anomaly score calculator $\mathcal{A}$ with parameter θ. The input of $\mathcal{A}$ is the audio clip $\boldsymbol{x}$ and additional information, and one anomaly score $\mathcal{A}_\theta(\boldsymbol{x})$ is output. Then, the machine is determined to be anomalous when the anomaly score $\mathcal{A}_\theta(\boldsymbol{x})$ exceeds a pre-defined threshold value ϕ as

$$\text{Decision} = \begin{cases} \text{Anomaly} & (\mathcal{A}_\theta(\boldsymbol{x}) > \phi) \\ \text{Normal} & (\text{otherwise}). \end{cases} \tag{1}$$

The primal difficulty in this task is to train $\mathcal{A}$ so that $\mathcal{A}_\theta(\boldsymbol{x})$ becomes a large value when the machine is anomalous, even though only normal sounds are available as training data.

In addition to the unsupervised ASD, we have to solve the domain-shift problem in real-world cases. As above, domain shifts

mean a difference of conditions between training and testing phases. The conditions differ in operating speed, machine load, viscosity, heating temperature, environmental noise, SNR, etc. The difference of conditions causes the gap of the sound characteristics, i.e., the distribution of the observation in the feature space changes. Here, two conditions are defined: **source domain** and **target domain**. The source domain means the original condition with enough training clips, and the target domain means another condition where only a few audio clips are available as training data. Let $\mathcal{D}_S$, $\mathcal{D}_T$, and $\mathcal{D}_{TA}$ be the distributions of $\boldsymbol{x}$ under the normal condition in the source domain, the normal condition in the target domain, and the anomalous condition in the target domain, respectively. The basic unsupervised ASD task is to determine if $\boldsymbol{x}$ is generated from $\mathcal{D}_T$ or $\mathcal{D}_{TA}$ under that clips from $\mathcal{D}_{TA}$ are unavailable as training data and enough training clips from $\mathcal{D}_T(= \mathcal{D}_S)$ are available. On the other hand, in the domain-shift scenario, the difference in conditions causes $\mathcal{D}_S \neq \mathcal{D}_T$, so the decision must be performed under that clips from $\mathcal{D}_{TA}$ are unavailable as training data and only a few training clips from $\mathcal{D}_T(\neq \mathcal{D}_S)$ are available. Therefore, the problem is more difficult than the basic unsupervised ASD.

3. TASK SETUP

3.1. Dataset

The data used for this task comprises parts of ToyADMOS2 [13] and MIMII DUE [14] consisting of the normal/anomalous operating sounds of seven types of toy/real machines. We intentionally damaged machines to collect the anomalous sounds in these datasets. We provide the following types of machines: ToyCar and ToyTrain from ToyADMOS2, and fan, gearbox, pump, slide rail, and valve from MIMII DUE. To simplify the task, we use only the first channel of multichannel recordings; all recordings can be regarded as single-channel recordings of a fixed microphone. Each recording is 10-sec-long audio that includes both the machine's operating sound and environmental noise. The sampling rate of all signals is 16 kHz. We mixed machine sounds with environmental noise, and only noisy recordings are available as training/test data. The environmental noise samples were recorded in several real factory environments. For the details of the recording procedure, please refer to the papers on ToyADMOS2 [13] and MIMII DUE [14].

In this task, we define two important terms: **machine type** and **section**.

- **Machine type** means the kind of machine, which can be one of seven in this task: fan, gearbox, pump, slide rail, ToyCar, ToyTrain, and valve.
- **Section** is defined as a subset of the data within one machine type and consists of data from the source and the target domains. A section is a unit for calculating performance metrics and is almost identical to "machine ID" in the 2020 version. In the 2020 version, there was a one-to-one correspondence between machine IDs and products, but in the 2021 version, the same product may appear in different sections, and multiple products may appear in the same section.

We provide three datasets: **development dataset**, **evaluation dataset**, and **additional training dataset**.

- The **development dataset** consists of three sections for each machine type (Section 00, 01, and 02), and each section is a complete set of training and test data. For each section, this dataset provides (i) around 1,000 clips of normal sounds in a source domain for training, (ii) only three clips of normal sounds in a target domain for training, (iii) around 100 clips each of normal and anomalous sounds in the source domain for the test, and (iv) around 100 clips each of normal and abnormal sounds in the target domain for the test.
- The **evaluation dataset** provides test clips for the three sections (Section 03, 04, and 05). Each section consists of (i) test clips in the source domain and (ii) test clips in the target domain, none of which have a condition label (i.e., normal or anomaly). Note that the sections of the evaluation dataset (Section 03, 04, and 05) are different from the development dataset (Section 00, 01, and 02).
- The **additional training dataset** provides the three sections identical to the evaluation dataset (Section 03, 04, and 05). Each section consists of (i) around 1,000 clips of normal sounds in a source domain for training and (ii) only three clips of normal sounds in a target domain for training.

As described above, the modification from the 2020 version is that the training dataset is highly unbalanced. Most of the training data (1000 clips for each section) are from the source domain, and only a few normal clips (three clips for each section) are available for the target domain, even though the number of clips in the test data set is the same in both domains. Thus, the participants have to develop a system adapted to the target domain using a few clips while avoiding performance degradation for the source domain.

3.2. Evaluation metrics

This task is evaluated with the AUC and the pAUC. The pAUC is an AUC calculated from a portion of the ROC curve over the prespecified range of interest. In our metric, the pAUC is calculated as the AUC over a low false-positive-rate (FPR) range $[0, p]$. The AUC and pAUC for each machine type, section, and domain are defined as

$$\mathrm{AUC}_{m,m,d} = \frac{1}{N_- N_+} \sum_{i=1}^{N_-} \sum_{j=1}^{N_+} \mathcal{H}(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-)), \quad (2)$$

$$\mathrm{pAUC}_{m,m,d} = \frac{1}{\lfloor pN_- \rfloor N_+} \sum_{i=1}^{\lfloor pN_- \rfloor} \sum_{j=1}^{N_+} \mathcal{H}(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-)), \quad (3)$$

where m represents the index of a machine type, n represents the index of a section, $d = \{\mathrm{source}, \mathrm{target}\}$ represents a domain, $\lfloor \cdot \rfloor$ is the flooring function, and $\mathcal{H}(x)$ returns 1 when $x > 0$ and 0 otherwise. Here, $\{x_i^-\}_{i=1}^{N_-}$ and $\{x_j^+\}_{j=1}^{N_+}$ are normal and anomalous test clips in the domain d in the section n in the machine type m, respectively, and they have been sorted so that their anomaly scores are in descending order. Here, N_- and N_+ are the number of normal and anomalous test clips in the domain d in the section n in the machine type m, respectively.

The reason for the additional use of the pAUC is based on practical requirements. If an ASD system frequently gives false alarms frequently, we cannot trust it. Therefore, it is important to increase the true-positive rate under low FPR conditions. In this task, we will use $p = 0.1$.

The official score Ω for each submitted system is given by the harmonic mean of the AUC and pAUC scores over all the machine

types and all the sections as follows:

$$\Omega = h\left\{\mathrm{AUC}_{m,n,d},\ \mathrm{pAUC}_{m,n,d} \mid m \in \mathcal{M},\ n \in \mathcal{S}(m),\ d \in \{\mathrm{source}, \mathrm{target}\}\right\}, \quad (4)$$

where $h\{\cdot\}$ represents the harmonic mean (over all machine types, sections, and domains), $\mathcal{M}$ represents the set of the machine types, and $\mathcal{S}(m)$ represents the set of the sections for the machine type m.

As the equations above show, a threshold value does not need to be determined to calculate AUC, pAUC, or the official score because the threshold value is the anomaly scores of normal test clips. However, in real applications, the threshold value must be determined, and a decision must be made as to whether it is normal or abnormal. Therefore, participants are also required to submit the normal/anomaly decision results.

3.3. Baseline systems and results

The task organizers provide two baseline systems. We present these baseline systems and their results.

3.3.1. Autoencoder-based baseline

The first baseline is an autoencoder (AE)-based anomaly score calculator and the same as the DCASE 2020 task 2. The anomaly score is calculated as the reconstruction error of the observed sound. To obtain small anomaly scores for normal sounds, the AE is trained to minimize the reconstruction error of the normal training data. This method is based on the assumption that the AE cannot reconstruct sounds that are not used in training, that is, unknown anomalous sounds.

In the baseline system, we first calculate the log-mel-spectrogram of the input $X = \{X_t\}_{t=1}^{T}$ where $X_t \in \mathbb{R}^F$, and F and T are the number of mel-filters and time-frames, respectively. Then, the acoustic feature at t is obtained by concatenating consecutive frames of the log-mel-spectrogram as $\psi_t = (X_t, \cdots, X_{t+P-1}) \in \mathbb{R}^D$, where $D = P \times F$, and P is the number of frames of the context window. The frame size of STFT is 64ms, and the hop size is 50 %. Also, since $F = 128$ and $P = 5$, $D = 640$.

The encoder/decoder of AEs consists of one input fully connected neural network (FCN) layer, three hidden FCN layers, and one output FCN layer. Each hidden layer has 128 hidden units, and the dimension of the encoder output is 8. The rectified linear unit (ReLU) is used after each FCN layer except the output layer of the decoder. The ADAM optimizer is used, and we fix the learning rate as 0.001. We stop the training process after 100 epochs, and the batch size is 512. We train models independently for each machine type, using normal clips from all sections of that machine type.

The anomaly score is calculated as:

$$A_\theta(X) = \frac{1}{DT}\sum_{t=1}^{T}\|\psi_t - r_\theta(\psi_t)\|_2^2, \quad (5)$$

where r_θ is the vector reconstructed by the autoencoder, and $\|\cdot\|_2$ is ℓ_2 norm.

To determine the anomaly detection threshold, we assume that A_θ follows a gamma distribution. The parameters of the gamma distribution are estimated from the histogram of A_θ, and the anomaly detection threshold is determined as the 90th percentile of the gamma distribution. If A_θ for each test clip is greater than this threshold, the clip is judged to be abnormal; if it is smaller, it is judged to be normal.

3.3.2. MobileNetV2-based baseline

The second baseline is an anomaly score calculator based on machine identification. In DCASE 2020, many teams used something like this approach [9, 11, 12], and mainly, a lot of them used MobileNetV2 [15]. With that in mind, this year, the organizers provide a MobileNetV2-based baseline.

The models of this baseline are trained to identify from which section the observed signal was generated. In other words, it outputs the softmax value that is the predicted probability for each section. The anomaly score is calculated as the averaged negative logit of the predicted probabilities for the correct section.

The acoustic feature (two-dimensional image) $\psi_t \in \mathbb{R}^{P\times F}$ is calculated in the same way as for AE. By shifting the context window by L frames, $B(= \lfloor\frac{T-P}{L}\rfloor)$ images are extracted. The frame size of STFT is 64ms, and the hop size is 50 %. Also, $F = 128$, $P = 64$, and $L = 8$. The ADAM optimizer is used, and we fix the learning rate as 0.00001. We stop the training process after 20 epochs, and the batch size is 32. We train models independently for each machine type, using normal clips from all sections of that machine type. The anomaly score is calculated as:

$$A_\theta(X) = \frac{1}{B}\sum_{b=1}^{B}\log\left(\frac{1-p_\theta(\psi_{t(b)})}{p_\theta(\psi_{t(b)})}\right), \quad (6)$$

where $t(b)$ is the beginning frame index of the b-th image, and p_θ is the softmax output by MobileNetV2 for the correct section. The anomaly detection threshold is determined as the same as the AE-based baseline.

3.3.3. Results

Table 1 and 2 show the AUC and pAUC scores for the two baselines. Because the results produced with a GPU are generally non-deterministic, the average and standard deviations from these five independent trials (training and testing) are shown in the following table.

4. CHALLENGE RESULTS

Challenge results and analysis of the submitted systems will be added for the official submission of the paper to the DCASE 2021 Workshop.

5. CONCLUSION

This paper presented an overview of the task and analysis of the solutions submitted to DCASE 2021 Challenge Task 2. We will add challenge results and analysis of the submitted systems for the official submission of the paper to the DCASE 2021 Workshop.

6. REFERENCES

[1] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing acoustic feature extractor for anomalous sound detection based on Neyman-Pearson lemma," in *Proc. 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 698–702.

Table 1: Results of the AE-based baseline

Section			AUC [%]		pAUC [%]	
			Source	Target	Source	Target
ToyCar	Dev.	00	67.63 ± 1.21	54.50 ± 0.89	51.87 ± 0.50	50.52 ± 0.20
		01	61.97 ± 1.50	64.12 ± 1.07	51.82 ± 0.87	52.14 ± 0.80
		02	74.36 ± 0.82	56.57 ± 1.53	55.56 ± 0.83	52.61 ± 1.20
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
ToyTrain	Dev.	00	72.67 ± 1.19	56.07 ± 0.80	69.38 ± 1.06	50.62 ± 0.68
		01	72.65 ± 0.32	51.13 ± 0.53	62.52 ± 0.88	48.60 ± 0.13
		02	69.91 ± 0.33	55.57 ± 1.07	47.48 ± 0.02	50.79 ± 0.93
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Fan	Dev.	00	66.69 ± 0.81	69.70 ± 0.32	57.08 ± 0.15	55.13 ± 0.34
		01	67.43 ± 1.12	49.99 ± 0.48	50.72 ± 0.42	48.49 ± 0.38
		02	64.21 ± 1.27	66.19 ± 1.23	53.12 ± 0.78	56.93 ± 1.37
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Gearbox	Dev.	00	56.03 ± 0.53	74.29 ± 0.51	51.59 ± 0.16	55.67 ± 0.97
		01	72.77 ± 0.72	72.12 ± 1.06	52.30 ± 0.18	51.78 ± 0.15
		02	58.96 ± 0.53	66.41 ± 0.72	51.82 ± 0.29	53.66 ± 0.57
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Pump	Dev.	00	67.48 ± 0.58	58.01 ± 0.57	61.83 ± 0.41	51.53 ± 0.27
		01	82.38 ± 0.27	47.35 ± 0.53	58.29 ± 0.77	49.65 ± 1.46
		02	63.93 ± 0.45	62.78 ± 0.70	55.44 ± 0.52	51.67 ± 0.35
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Slide rail	Dev.	00	74.09 ± 0.48	67.22 ± 0.45	52.45 ± 0.63	57.32 ± 0.52
		01	82.16 ± 0.35	66.94 ± 0.39	60.29 ± 0.30	53.08 ± 0.39
		02	78.34 ± 0.16	46.20 ± 0.77	65.16 ± 0.55	50.10 ± 0.31
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Valve	Dev.	00	50.34 ± 0.27	47.12 ± 0.18	50.82 ± 0.16	48.68 ± 0.09
		01	53.52 ± 0.33	56.39 ± 1.42	49.33 ± 0.10	53.88 ± 0.61
		02	59.91 ± 0.34	55.16 ± 0.22	51.96 ± 0.52	48.97 ± 0.04
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–

Table 2: Results of the MobileNetV2-based baseline

Section			AUC [%]		pAUC [%]	
			Source	Target	Source	Target
ToyCar	Dev.	00	66.56 ± 2.68	61.32 ± 5.94	66.47 ± 5.67	52.61 ± 2.41
		01	71.58 ± 5.54	72.48 ± 3.68	66.44 ± 2.84	63.99 ± 2.60
		02	40.37 ± 7.19	45.17 ± 3.36	47.48 ± 0.23	48.85 ± 0.94
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
ToyTrain	Dev.	00	69.84 ± 4.39	46.28 ± 3.85	54.43 ± 1.65	51.27 ± 0.73
		01	64.79 ± 3.65	53.38 ± 2.47	54.09 ± 1.15	49.60 ± 0.88
		02	69.28 ± 6.73	51.42 ± 2.64	47.66 ± 0.40	53.40 ± 1.12
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Fan	Dev.	00	43.62 ± 2.35	53.34 ± 2.03	50.45 ± 1.15	56.01 ± 1.38
		01	78.33 ± 1.52	78.12 ± 4.25	78.37 ± 2.26	66.41 ± 7.16
		02	74.21 ± 3.85	60.35 ± 3.79	76.80 ± 0.78	60.97 ± 6.55
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Gearbox	Dev.	00	81.35 ± 1.59	75.02 ± 2.92	70.46 ± 3.67	64.77 ± 2.52
		01	60.74 ± 5.11	56.27 ± 8.27	53.88 ± 2.82	53.30 ± 2.97
		02	71.58 ± 7.16	64.45 ± 9.67	62.23 ± 6.67	55.58 ± 7.90
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Pump	Dev.	00	64.09 ± 4.34	59.09 ± 3.08	62.40 ± 1.90	53.96 ± 0.93
		01	86.27 ± 3.18	71.86 ± 5.97	66.66 ± 5.23	62.69 ± 2.33
		02	53.70 ± 4.99	50.16 ± 3.78	50.98 ± 1.23	51.69 ± 1.03
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Slide rail	Dev.	00	61.51 ± 4.92	51.96 ± 3.17	53.97 ± 2.03	51.96 ± 2.96
		01	79.97 ± 3.70	46.83 ± 10.65	55.62 ± 1.57	52.02 ± 4.17
		02	79.86 ± 1.41	55.61 ± 5.48	71.88 ± 4.64	55.71 ± 2.84
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–
Valve	Dev.	00	58.34 ± 4.01	52.19 ± 3.33	54.97 ± 4.43	51.54 ± 1.88
		01	53.57 ± 2.26	68.59 ± 2.84	50.09 ± 0.45	57.83 ± 2.49
		02	56.13 ± 1.96	53.58 ± 0.55	51.69 ± 0.32	50.86 ± 0.84
	Eval.	03	–	–	–	–
		04	–	–	–	–
		05	–	–	–	–

[2] Y. Kawaguchi and T. Endo, "How can we detect anomalies from subsampled audio signals?" in *Proc. 27th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017.

[3] 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 Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 212–224, Jan. 2019.

[4] Y. Kawaguchi, R. Tanabe, T. Endo, K. Ichige, and K. Hamada, "Anomaly detection based on an ensemble of dereverberation and anomalous sound extraction," in *Proc. 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 865–869.

[5] Y. Koizumi, S. Saito, M. Yamaguchi, S. Murata, and N. Harada, "Batch uniformization for minimizing maximum anomaly score of DNN-based anomaly detection in sounds," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 6–10.

[6] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *Proc. 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 271–275.

[7] H. Purohit, R. Tanabe, T. Endo, K. Suefusa, Y. Nikaido, and Y. Kawaguchi, "Deep autoencoding GMM-based unsupervised anomaly detection in acoustic signals and its hyperparameter optimization," in *Proc. 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 175–179.

[8] 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. 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 81–85.

[9] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proc. 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 46–50.

[10] S. Kapka, "ID-conditioned auto-encoder for unsupervised anomaly detection," in *Proc. 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 71–75.

[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. 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 170–174.

[12] 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. 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 66–70.

[13] 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.

[14] 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 preprint arXiv:2105.02702*, 2021.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.