INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Variational Classifier for Unsupervised Anomalous Sound Detection under Domain Generalization

Antonio Almudévar, Alfonso Ortega, Luis Vicente, Antonio Miguel, Eduardo Lleida

ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{almudevar, ortega, lvicente, amiguel, lleida}@unizar.es

Abstract

Unsupervised anomalous sound detection typically involves using a classifier with the last layer removed to extract embeddings. After that the cosine distance between train and test embeddings as anomaly score is used. In this paper, we propose a new idea which we call variational classifier that force the embeddings to follow a distribution imposed by design that can depend on the class of the input among other factors. To achieve this goal, in addition to the cross-entropy, we add to the loss function the KL divergence between these distributions and the one followed by the training embeddings. This enhances the ability of the system to differentiate between classes and it allows us to use sampling methods and to calculate the log-likelihood of a test embedding in the train embeddings distributions. We tested this proposal on the DCASE 2022 Task 2 dataset and observed improvements in both classification and unsupervised anomaly detection, which is the primary task.

Index Terms: variational classifier, anomalous sound detection, embeddings extractor

1. Introduction

Unsupervised Anomaly Detection is the task of distinguishing between normal and anomalous data using only normal data for design. The fact of dispensing with anomaly data for the design is key since, as the name suggests, it is difficult to obtain this type of data. The common objective of all anomaly detection systems is to obtain an output value called anomaly score, which determines whether an input is normal or anomalous depending on whether this value is below or above, respectively, a threshold [1, 2]. To design this type of system, a common solution is to design a system capable of obtaining embeddings containing the semantic information of the inputs. Subsequently, the Euclidean or cosine distance between the training and test embeddings are often used as the anomaly score [3]. Thus, since the training embeddings are normal, the distance between the test and training embeddings is expected to be higher when the test data are anomalous than when they are normal. Another alternative is to design a generative model such as a Variational Autoencoder [4] or Normalizing Flow [5] to estimate the distribution that the training embeddings follow so that the anomaly score is the negative likelihood, which is expected to be higher for the anomalous input embeddings [6, 7]. To build the embeddings extractor it is common to train a classifier that distinguishes between different types of normal data and to remove the last layer once this classifier has been trained [2]. In this work we propose a classifier in which the output of the last layer is not a unique embedding per input, but rather the parameters of the distribution followed by the embeddings corresponding to that input. Thus, by knowing the distribution we can combine the two previous solutions, i.e., we can sample embeddings from that distribution and compute cosine distances, and we can also compute the likelihood.

Although the technique we propose in this work is independent of the type of data, in this case we use it to detect anomalies in audios. Specifically, we use the dataset of the DCASE 2022 Task 2 challenge, since it is a de facto standard for designing and evaluating Unsupervised Anomalous Sound Detection systems. This dataset is composed of sounds corresponding to seven different machines and the challenge proposes a metric to evaluate the system. On the one hand, DCASE2021 winner combined three different systems to achieve the best possible performance. The first system obtained x-vectors and calculated the cosine and Mahalanobis distances between the test embedding and the average training embedding as an anomaly score. The second system used WaveNet [8] by adding an x-vector classification head after the dilated convolutions. The last one used a Normalizing Flow to estimate the distribution of an n-bin segment of a spectrogram conditioned to the remaining bins [9]. On the other hand, the DCASE2022 winner used a modified version of MobileFaceNet [10] as embedding extractor. Subsequently, they used the negative logit and the Mahalanobis distance between training and test data as the anomaly score depending on the machine [11]. As can be seen, the winners of the last few years first use embedding extractors and then implement different methods to obtain the anomaly scores from these embeddings. These methods include the cosine and Mahalanobis distances between train and test embeddings or a generative model such as a Normalizing Flow. The objective of this work is to present a monolithic system that allows to calculate the cosine distance and estimate the likelihood of the test embeddings in the train distributions.

This paper is organized as follows. Section 2 presents the problem objetive, dataset and evaluation score. Section 3 describes the proposed approach to solve the problem. The results obtained are shown in section 4. Finally, in section 5, conclusions are summarized.

2. Problem Description

2.1. Objective

In this paper we present a solution for the DCASE2022 Task 2. The goal of this task is to develop a system capable of distinguishing between normal and anomalous audios. In order to design this system, only normal data are used to train it. Besides, the test data may or may not be in another domain than the training data. Thus, when we have a test data, a priori we do not know if it is from the source or the target domain. Therefore, we must use the same threshold for all data regardless of the domain. This is known as domain generalization [12].

10.21437/Interspeech.2023-1965

2.2. Dataset

The DCASE2022 task 2 dataset is composed of data from ToyADMOS2 [13] and MIMII DUE [14] and contains the sounds emitted by seven machines operating normally and when broken (abnormally). In addition, for each machine there are sounds belonging to six sections. A section is defined as a subset of the dataset for calculating performance metrics. The machines are: ToyCar, ToyTrain, fan, gearbox, bearing, slide rail, and valve and sections 0 to 5. The recordings are 10 seconds long, single channel and sampled at 16 kHz. The dataset is splitted as follows:

Training Dataset: Consists of six sections for each machine type (Sections 0 to 5). For each section, this dataset is composed of 990 clips of normal sounds in the source domain for training, 10 clips of normal sounds in the target domain.

Validation Dataset: It is composed of data from sections 0, 1 and 2 and has 100 normal audios (50 from the source and 50 from the target domain) and 100 anomalous audios (50 from the source and 50 from the target domain) per section.

Evaluation Dataset: It follows the same distribution as the validation dataset, but with data from sections 3, 4 and 5.

2.3. Evaluation

To evaluate this task we use the area under the ROC curve (AUC) and partial AUC (pAUC) [15]. Since this task focuses on domain generalization, the threshold should be the same regardless of the domain of the data, since in the test data we do not have this information. Therefore, we first obtain the AUC_{mnd} and $pAUC_{mn}$, which are the AUC and pAUC with p=10% for machine m, section n and domain d. Finally, the score Ω_m for a machine m and the final score of the task Ω is defined in 2.

$$\Omega_m = h\{AUC_{mnd},\ pAUC_{mn} : n \in \mathcal{S},\ d \in \{\mathfrak{S}, \mathfrak{T}\}\} \quad (1)$$

where $\mathfrak{S}$ and $\mathfrak{T}$ are the source and target domains, repectively.

$$\Omega = h\{\Omega_m : m \in \mathcal{M}\} \quad (2)$$

where h is the harmonic mean, $\mathcal{S} = \{0, 1, 2, 3, 4, 5\}$ and $\mathcal{M}$ are the seven machines

3. Proposed Approach

3.1. Input to Classifier

To obtain the inputs to this network, first, we calculate the spectrogram (or its variations) of the signal $X = \{x_t\}_{t=1}^{T}$, where $x_t \in \mathbb{R}^F$ and F and T are the number of frequency bins and time frames, respectively. As input to the embeddings extractor we take five chunks of L frames. During training phase the initial position of these chunks is chosen randomly and during the test phase it is chosen equispaced among the F frames. The configurations of the inputs that return the best performance for each machine are presented in section 4

3.2. Vanilla Classifier

In the Vanilla Classifier, we distinguish two parts. The first one is f_θ, which is the embeddings extractor and the second one is g_ϕ, which is the last layer and whose output contains the probabilities of each class. Given an input x_i:

$$\boldsymbol{z}_i = f_\theta(x_i) \in \mathbb{R}^k \quad (3)$$

$$\boldsymbol{y}_i = g_\phi(\boldsymbol{z}_i) \in \mathbb{R}^C \quad (4)$$

where $\boldsymbol{z}_i$ is the embedding of the input x_i, k is the embeddings dimension and y_{ij} is the probability that x_i, belongs to class $j \in \{1, 2, \ldots, C\}$.

The loss function for this classifier is the cross-entropy between the labels $\boldsymbol{t}_i$ corresponding to the input x_i and the output of the classifier $\boldsymbol{y}_i$, so that:

$$\mathcal{L}_C(\boldsymbol{t}_i, \boldsymbol{y}_i) = CE(\boldsymbol{t}_i, \boldsymbol{y}_i) \quad (5)$$

3.3. Variational Classifier

In the variational classifier, the aim is that the embeddings are close to a set of known distributions, so that we can draw samples from this distribution and also calculate the likelihood of an embedding in these distributions. Since the objective is to discriminate between inputs of different classes, we define as many distributions as classes. In this way, the embedding of each input will try to follow a distribution depending on the class. In particular, we choose these distributions to be Gaussian, although the use of different families of distributions could be studied. In addition, the means of the different distributions are orthogonal to each other and have magnitude μ. Enforcing that embeddings of different classes tend to be orthogonal to each other has been found to improve classification performance [16]. In addition, the covariance matrix is $\sigma^2 I$ in the C distributions. Therefore, the embedding corresponding to input x_i should be distributed close to a distribution $\mathcal{N}\left(\boldsymbol{\mu}_{t_i}, \sigma^2 I\right)$ and it is also satisfied that $\langle \boldsymbol{\mu}_{t_j}, \boldsymbol{\mu}_{t_k} \rangle = 0$ if $t_j \neq t_k$.

To model this, in the variational classifier two other parts are added with respect to the Vanilla Classifier, which are h_{θ_μ} and h_{θ_σ} and that are two affine transformations $R^k \rightarrow R^k$. Thus, we have that:

$$\boldsymbol{\mu}_{z_i} = h_{\theta_\mu}(f_\theta(x_i)) \in \mathbb{R}^k \quad (6)$$

$$\boldsymbol{\sigma}_{z_i} = h_{\theta_\sigma}(f_\theta(x_i)) \in \mathbb{R}^k \quad (7)$$

We must note that these transformations add a number of parameters that, for common values of k, is quite small with respect to the total number of parameters of the whole system. Specifically, the total number of trainable parameters added is $2k(k+1)$. Once $\boldsymbol{\mu}_{z_i}$ and $\boldsymbol{\sigma}_{z_i}$ are obtained, during training we take one sample from this distribution, according to expression 8. The number of samples could be larger, but for simplicity, we have left it at one in this work. On the other hand, during the test phase, we take $\boldsymbol{z}_i = \boldsymbol{\mu}_{z_i}$.

$$z_i \sim \mathcal{N}\left(\boldsymbol{\mu}_{z_i}, \operatorname{diag}(\boldsymbol{\sigma}_{z_i}^2)\right) \quad (8)$$

Finally, to obtain the predicted class of the input x_i, we have that:

$$y_i = g_\phi\left(\boldsymbol{z}_i\right) \in \mathbb{R}^C \quad (9)$$

Figure 1 shows the variational classifier schematic. In this case, the loss function is defined as:

$$\begin{aligned}\mathcal{L}_{VC}(\boldsymbol{t}_i, \boldsymbol{y}_i) = {} & \beta_{class} \cdot CE(t_i, \boldsymbol{y}_i) \\ & + \frac{1}{k} D_{KL}\left(\mathcal{N}(\boldsymbol{\mu}_{z_i}, \operatorname{diag}(\boldsymbol{\sigma}_{z_i}^2)) \| \mathcal{N}(\boldsymbol{\mu}_{t_i}, \sigma^2 I)\right)\end{aligned} \quad (10)$$

The first term is the cross-entropy, commonly used in vanilla classifiers. The coefficient β_{class} serves to determine the importance given to the classification task and in the section 4 we study the influence of its value. The second term is the novelty of this work and establishes that the embedding space is distributed around distributions depending on the class of the data. In particular, given an embedding $\boldsymbol{z}_i$, this second term is:

$$\begin{aligned} & D_{KL}\left(\mathcal{N}(\boldsymbol{\mu}_{z_i}, \operatorname{diag}(\boldsymbol{\sigma}_{z_i}^2)) \| \mathcal{N}(\boldsymbol{\mu}_{t_i}, \sigma^2 I)\right) \\ & = \frac{1}{2}\left[k \log \sigma^2 - \sum_{n=1}^{k} \log(\boldsymbol{\sigma}_{z_i})_n^2 - k\right. \\ & \left. + \frac{1}{\sigma^2}\sum_{n=1}^{k}((\boldsymbol{\mu}_{z_i})_n - (\boldsymbol{\mu}_{t_i})_n)^2 + \frac{1}{\sigma^2}\sum_{n=1}^{k}(\boldsymbol{\sigma}_{z_i})_n^2\right] \end{aligned} \tag{11}$$

Finally, it should be noted that to use backpropagation algorithm it is necessary to use the reparameterization trick, which is proposed in [4].

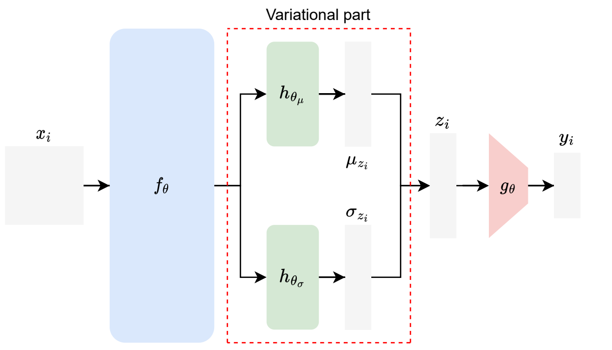

Figure 1: *Schematic of a Variational Classifier. By eliminating the blocks inside the red dashed line we would obtain a vanilla classifier.*

4. Results

In this section we describe the implementation and training details and present and analyze the results obtained by using it.

4.1. Implementation and Training Details

To measure the performance of this proposal, we use as our embeddings extractor g_ϕ a very similar architecture to the one proposed in [11] and that is shown in Table 1, which is a slight modification of MobileFaceNet. For this model, since $k = 128$, the number of trainable parameters added is 33024, an increase of 3.89% over the total, despite being a model with around 850k parameters. Although other works train first with data from all machines and then finetuning for each machine obtaining a classifier for each one [11], for simplicity, we have directly trained a classifier for each machine from scratch to classify the section to which each machine belongs. In order to achieve the best possible performance we have conducted a grid search over validation dataset to find the best spectrogram configurations for each machine, which are presented in Table 2. To train the models we use Adam optimizer [17] with a learning rate of 0.001 during 50 epochs with multistep scheduler with a decay rate equal to 0.2 in epochs 25 and 40 and warmup during 10 epochs. The batch size used is 64 and we also introduce mixup with $\alpha = 0.8$ and SpecAugment for data augmentation. To implement the mixup in this proposal we have had to add a new equation since in addition to mixing the inputs x_i, x_j, the labels $\boldsymbol{t}_i$, $\boldsymbol{t}_j$, we also mix the means corresponding to the class of each input in the form:

$$\tilde{\mu} = \lambda \cdot \boldsymbol{\mu}_{t_i} + (1-\lambda) \cdot \boldsymbol{\mu}_{t_j} \tag{12}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$.

Table 1: *Embeddings extractor architecture, where t is the expansion factor, c is the output channels, n is the number of inverted residuals blocks, and s is the stride.*

Operator	t	c	n	s
Conv2d 3x3	-	64	-	2
Blockneck	2	128	2	2
Blockneck	4	128	2	2
Blockneck	4	128	2	2
Conv2d 1x1	-	512	-	1
Linear GDConv2d	-	512	-	1
Linear Conv2d 1x1	-	128	-	1

Table 2: *Best Spectrogram for each machine*

	FFT Length	Chunk size	Mel filters	db
ToyCar	256	512	256	Yes
ToyTrain	128	1024	128	No
Bearing	1024	128	512	Yes
Fan	512	256	512	Yes
Gearbox	400	400	nan	Yes
Slider	128	1024	128	Yes
Valve	512	256	512	No

4.2. Anomaly Detection Performance

The main goal of this work is to study whether variational classifiers improve performance in the anomaly detection task. For this purpose, we study how the different hyperparameters of these systems influence. In addition, we compare the results obtained using a variational classifier with respect to using a vanilla classifier trained with cross-entropy or ArcFace [18], which are two of the most used loss functions to obtain embeddings used to detect anomalies [19, 20]. In the case of variational classifier we define two anomaly scores. The first one is the cosine distance of each test embedding with all the train embeddings and choose the minimum. To obtain the second one, we take each test embedding and calculate its negative likelihood in each distribution of the train embeddings, choosing the minimum of all as the anomaly score. In the vanilla classifier we take only the first anomaly score, since it is not possible to calculate the second one because we do not know the distribution of the embeddings.

Let's first analyze how the μ and σ parameters influence performance at each anomaly score. To do so, we set $\beta_{class} = 1$. In Table 3 we see that, when the anomaly score used is the cosine distance, lower means magnitude perform better in general. In addition, it is convenient to reduce the variances when the means are lower. On the other hand, we see that, when negative likelihood is used as anomaly score, lower means also work better and that, in general terms, lower variances work better.

In order to analyze the influence of β_{class}, we take the best μ and σ found for each anomaly score in Table 3. The results are shown in Figure 2. In this case we can see that the case that works best for both scores is $\beta_{class} = 1$. On the other hand, the extreme values perform better than 0.1 in the case of the cosine distance, while in the case where the anomaly score is the negative likelihood the opposite happens.

Table 3: Ω *depending on* μ *and* σ *when cosine distance*$|$*negative likelihood is used as the anomaly score.*

		μ 5	10	20	30
σ^2	0.02	66.3$|$65.9	65.6$\|$**66.0**	66.1$|$64.6	63.7$|$64.2
	0.05	65.5$|$64.4	65.8$|$65.4	65.9$|$64.5	64.8$|$64.5
	0.1	65.9$|$64.8	**66.4**$|$64.5	66.3$|$63.2	64.7$|$63.7
	0.2	64.5$|$63.7	66.3$|$65.3	65.2$|$63.4	63.5$|$63.0

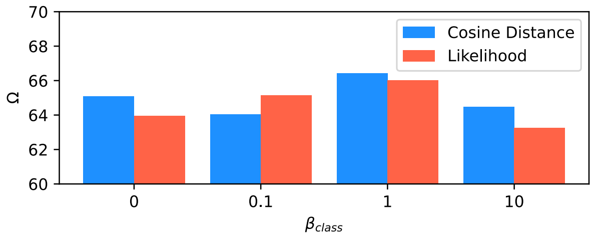

Figure 2: *Comparison of* β_{class}.

Finally, we have observed that there are hyperparameter configurations that work better on some machines than on others. Thus, in order to optimize as much as possible the result in the proposed dataset, we have used the validation data to find the best hyperparameter configurations and the best anomaly score for each machine. Table 4 shows the best hyperparameter configuration for each machine. In addition, in Figure 3 we show the results compared with a vanilla classifier of identical architecture with cross-entropy and ArcFace as loss functions. As we can see, both the best on average and the best per machine significantly outperform the vanilla classifier. We also compare in Figure 3 our variational classifier with the three best systems of DCASE 2022 Task 2. We can see, that the best per machine outperforms [21] and [22]. However, it does not outperform [11]. Nevertheless, it must be emphasized that the system proposed in that paper is an ensemble of several systems for each machine, while ours is a single system per machine. This has the advantage that for each input audio a single prediction has to be made, which is crucial for this problem, which in industry is intended to work in real time. In addition, the number of parameters of our system, not being an ensemble of several, is much smaller.

Table 4: *Best hyperparameters of the variational classifier per machine for Anomaly Detection.*

	μ	σ^2	β_{class}	Anomaly Score
ToyCar	5	0.5	1	Neg. Likelihood
ToyTrain	10	0.05	1	Neg. Likelihood
Bearing	10	0.2	1	Cos. Distance
Fan	10	0.02	10	Neg. Likelihood
Gearbox	5	0.02	0.1	Neg. Likelihood
Slider	10	0.02	1	Cos. Distance
Valve	10	0.1	1	Cos. Distance

4.3. Classification Performance

Although it is not the main objective of this work, we have found that the variational classifier, in addition to presenting better performance in anomaly detection, is also a better classifier. This may be due to the fact that the embeddings of the same class are more concentrated among them. In addition, as embeddings from different classes are trained to be orthogonal to each other, there is more separation between them in space. Specifically, we performed a search similar to the one described in the previous section, achieving the best on average a mean accuracy of 71.1% when $\mu = 5$, $\sigma^2 = 0.02$, $\beta_{class} = 1$. Furthermore, if we choose the best combination of hyperparameters for each machine, the mean accuracy is 72.4%. In this case, we do not show these configurations, since classification is not the task of this challenge. The two previous results greatly outperform the average accuracy of the vanilla classifier trained with cross-entropy loss, which is 68.8%.

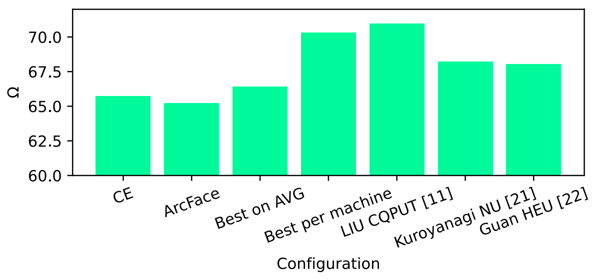

Figure 3: *Comparison of the variational classifier with the vanilla classifier and the three best performing systems of DCASE2022 Task 2.*

5. Conclusions

In this work we have presented the variational classifier, which allows to obtain the distribution of the embeddings for each input instead of calculating only one embedding. This allows the negative likelihood of the test embeddings in the distributions of the train embeddings to be used as an anomaly score. In addition, we can generate embeddings from these distributions and calculate the cosine distance between the train and test embeddings. This combines the two most commonly used methods for anomaly detection into one. In addition we have seen that the performance is better in the variational classifier than in a vanilla classifier when its loss function is the cross-entropy or the ArcFace. In particular, we have presented the hyperparameter configuration that provides the best results on average in this dataset, which is made up of audios from diverse origins. Therefore, we conjecture that these results can be extrapolated to work with audios from other types of machines. Finally, we have imposed that embeddings of different classes tend to be orthogonal to each other, which also improves the performance in the classification task with respect to a vanilla classifier trained with the cross-entropy as loss function.

6. Acknowledgements

This work was supported in part by MCIN/AEI/10.13039/501100011033, and the European Union NextGenerationEU/PRTR under Grant PDC2021-120846-C41 and Grant PID2021-126061OB-C44, and in part by the Government of Aragon under Grant T36_23R.

7. References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[2] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[3] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70–91, 2014.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[5] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[6] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[7] P. Kirichenko, P. Izmailov, and A. G. Wilson, "Why normalizing flows fail to detect out-of-distribution data," *Advances in neural information processing systems*, vol. 33, pp. 20 578–20 589, 2020.

[8] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

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

[10] 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.

[11] Y. Zeng, H. Liu, L. Xu, Y. Zhou, and L. Gan, "Robust anomaly sound detection framework for machine condition monitoring," DCASE2022 Challenge, Tech. Rep., July 2022.

[12] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International conference on machine learning*. PMLR, 2013, pp. 10–18.

[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," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 1–5.

[14] 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," *In arXiv e-prints: 2205.13879*, 2022.

[15] P. Gimeno, V. Mingote, A. Ortega, A. Miguel, and E. Lleida, "Partial auc optimisation using recurrent neural networks for music detection with limited training data," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, vol. 2020-October, 2020, pp. 3067–3071.

[16] K. Ranasinghe, M. Naseer, M. Hayat, S. Khan, and F. S. Khan, "Orthogonal projection loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 333–12 343.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] 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.

[19] R. Giri, S. V. Tenneti, K. Helwani, F. Cheng, U. Isik, and A. Krishnaswamy, "Unsupervised anomalous sound detection using self-supervised classification and group masked autoencoder for density estimation," *Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2020 Challenge), Tech. Rep*, vol. 23, 2020.

[20] 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)*. IEEE, 2022, pp. 816–820.

[21] I. Kuroyanagi, T. Hayashi, K. Takeda, and T. Toda, "Two-stage anomalous sound detection systems using domain generalization and specialization techniques," DCASE2022 Challenge, Tech. Rep., July 2022.

[22] F. Xiao, Y. Liu, Y. Wei, J. Guan, Q. Zhu, T. Zheng, and J. Han, "The dcase2022 challenge task 2 system: Anomalous sound detection with self-supervised attribute classification and gmm-based clustering," DCASE2022 Challenge, Tech. Rep., July 2022.