ENSEMBLE OF MULTIPLE ANOMALY DETECTORS UNDER DOMAIN GENERALIZATION CONDITIONS

Technical Report

Shuxian Wang[1]*, Yajian Wang*[1]*, Diyuan Liu*[2]*, Fan Chu*[2]*, Yunqing Li*[1]*,*
Jia Pan[2]*, Jun Du*[1]*, Tian Gao*[2]*, Qing Wang*[1]

[1] University of Science and Technology of China, Hefei, China
{sxwang21, yajian, lyq123}@mail.ustc.edu.cn, {jundu, qingwang2}@ustc.edu.cn
[2] iFLYTEK, Hefei, China
{dyliu2, fanchu, jiapan, tiangao5}@iflytek.com

ABSTRACT

This technical report outlines our solution to DCASE 2022 Challenge Task 2, Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques. The goal is to detect recordings that contain anomalous machine sounds in the test set using only normal sound data in the training set. Our approaches are based on an ensemble of a self-supervised classifier model, an autoencoder, a binary classification model that utilizes task irrelevant outliers as pseudo-anomalous data and a distance metric based model.

Index Terms— DCASE, unsupervised anomalous sound detection, domain generalization

1. INTRODUCTION

In DCASE challenge 2022 Task 2 "*Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques*" [1], it is required to detect anomalous sounds of machines. In real-world conditions, it is often easier for us to obtain the sound of the machine working normally, while the anomalies are rare and highly diverse. Therefore, we need to use the normal sounds in the training data to detect anomalous sounds in the test data. Furthermore, the acoustic properties of training data and test data are different, i.e. domain shift. In the DCASE2022 task, a new requirement for domain generalization techniques is added, that is, the anomalous sound detection (ASD) system required to be developed does not need to detect domain shifts or adjust models to detect anomalies, and the domain of each sample is not provided in the test data.

Our submission includes an ensemble of four major approaches for anomalous sound detection. First approach is to use a self-supervised classifier to classify the machine's section ID. Our second approach is based on an autoencoder (AE) to detect anomalous sounds, that is, the anomalous score is calculated as the reconstruction error of the observed sound. The third is a method based on binary classification. Since there is no anomalous data in the training set, we first utilize task irrelevant outliers as pseudo-anomalous data , and then train a binary classifier to classify normal and anomalous sounds. The fourth method we use is based on distance metric. Specifically, we first train a classifier (i.e., the first method) and an autoencoder (i.e., the second method), and then extract embeddings from them to compute distances to obtain anomaly scores, respectively.

In the following, we describe each approach and our experimental results in detail. Each recording used in this challenge is a single-channel and 10-second long audio, including seven machines: ToyCar, ToyTrain, fan, gearbox, bearing, slide rail and valve [2, 3].

2. PROPOSED APPROACH

2.1. Self-Supervised Classification

Since the section ID of the machine is known, we can detect anomalous sounds by identifying the machine's section ID. Anomalous sound detection methods based on self-supervised classification have been used before with good results [4, 5]. Moreover, in the DCASE2020 and DCASE2021 challenges, many teams have used this method and achieved satisfactory results [6, 7]. Therefore, we adopt this method to detect anomalous sounds.

2.1.1. Audio Processing

We transformed all audio clip into spectrograms with or without a Mel transformation, and the logarithm was taken for both the STFT and the Mel spectrograms. At the same time, the paper [8] shows that the information in the time domain has a complementary effect on the spectrogram. Therefore, based on the STgram structure [8], we first extracted the feature information based on the raw wave, and then concatenated it to the spectrogram or mel spectrogram as the input of the classifier. In addition, in order to improve the generalization of the model and the representation of the feature vector, we trained Self-Supervised Audio Spectrogram Transformer(SSAST) [9] based on the AudioSet [10], and then extracted the feature vector to connect with the above features.

We found that characteristic parameters such as the number of FFT points, window shift, etc. have a greater impact on the performance, and we have explored a set of optimal parameters for each machine through experiments, see Table 1 for details.

2.1.2. Classifier Architectures, Training and Results

Considering the complementarity between different model structures, we adopt a total of four model structures: TDNN-Xvector,

ResNet34, Resnet-Conformer and FCNN (fully convolutional neural networks), which are all attached with a softmax layer. Cross-entropy loss and batch hard Triplet loss were used. In addition, we also used the attribute information contained in each audio to calculate the auxiliary loss. The softmax classification score of the test sample, measured at the output corresponding to its true machine ID, was used to calculate the anomaly score as follows:

$$A_\theta(X) = \log \frac{1 - p_\theta}{p_\theta} \tag{1}$$

where X is the feature of each audio as input to the model, p_θ is the softmax output of the model for the correct section ID, and A_θ is the anomaly score for each audio.

We used the AdamW optimizer, however, on bearing machines we found that using SGD worked better. All training data from the development and evaluation datasets were used for training. We fused the results of the four model structures, and the results are shown in Table 1.

Table 1: Self-Supervised Classification Scoring Results(%)

	ToyCar	**ToyTrain**	**bearing**	**fan**	**gearbox**	**slider**	**valve**
Feature	MEL	MEL	MEL	MEL	STFT	STFT	MEL
Num_FFT	2048	1024	4096	8192	2048	4096	2048
Num_Mels	128	64	64	128	N/A	N/A	256
Hop length	256	256	512	512	256	512	512
h-mean AUC (source)	67.95	78.05	78.59	96.57	89.13	96.73	90.19
h-mean AUC (target)	68.88	53.12	77.61	79.53	73.65	86.60	89.50
h-mean pAUC	58.37	54.75	66.12	76.46	63.65	83.22	79.02

2.2. Autoencoder

The autoencoder (AE) is based on the reconstruction error to realize the detection of anomalous sound. That is, the input feature vector is first mapped to a hidden representation with a lower dimensional space by the encoder component, and then, the decoder component attempts to reconstruct the inverse transformation from the hidden representation to the original input signal. The difference between the feature vector of the original input and the output vector of the autoencoder is the reconstruction error. First, we used the normal samples in the training set to train the AE to minimize the reconstruction error. In this way, for the test sample, if it is normal sound, the AE can reproduce it well, but for the anomalous sound that has not been seen during training, the reconstruction error will be bigger. Therefore, the magnitude of the reconstruction error can be used to detect anomalous sounds.

2.2.1. Training and Results

We used a convolutional AE structure similar to [11], and we trained an AE separately for each section of data for each machine. The input of the model was a 128-dimensional logmel spectrogram. To train this model, Adam optimizer is used with the default learning rate of 5×10^4 for 100 epochs, and the results are shown in Table 2.

Table 2: Autoencoder Scoring Results(%)

	ToyCar	**ToyTrain**	**bearing**	**fan**	**gearbox**	**slider**	**valve**
h-mean AUC (source)	86.98	79.55	59.19	76.29	79.25	91.15	61.65
h-mean AUC (target)	60.80	34.03	69.75	40.28	68.21	62.26	55.61
h-mean pAUC	54.76	50.42	52.82	57.27	60.32	62.96	50.40

2.3. Binary Classification

Binary classification models are trained by true normal sounds and pseudo abnormal sounds. The normal sounds of target section of the machine is used as positive data, while the normal sounds of other section and a part of normal data of other machine are used as pseudo negative data to train the binary classification models. In particular, we made the data cleaning of pseudo negative data of ToyCar and Fan by removing the examples whose attributes are completely consistent with the normal sounds. Log Mel-filterbank (LMFB) features are employed as inputs of the system. To generate LMFB, the short-time Fourier transform (STFT) with 1024 FFT points is applied, utilizing a window size of 1024 samples, a hop length of 512 samples and a dimensional Mel basis of 128. Besides, the feature is normalized before being sent to the network.

The network architectures used in binary classification are the same as those used in self-supervised classification including TDNN-Xvector, ResNet34, Resnet-Conformer and FCNN (fully convolutional neural networks). Mixup, which randomly mixes data batches with corresponding labels, is applied to enhance the generalization ability of the model. In addition to Cross Entropy Loss, Focal Loss is also utilized to balance the imbalance number of normal and abnormal samples. The batchsize of training is set as 32 and Adam optimizer is used to train the model with the learning rate of 0.0001. Due to the significant variation during training of binary classification model, we performed the posterior fusion on the scores of the last 10 epochs. The results of binary classification are shown in Table 3.

Table 3: Binary Classification Scoring Results(%)

	ToyCar	**ToyTrain**	**bearing**	**fan**	**gearbox**	**slider**	**valve**
h-mean AUC (source)	78.56	79.62	82.76	80.26	84.41	98.62	87.50
h-mean AUC (target)	70.85	45.29	86.76	59.04	67.62	79.11	90.59
h-mean pAUC	64.69	52.10	66.31	65.54	62.65	68.43	84.36

2.4. Distance Metric

The basic idea of the distance metric-based anomalous sound detection method is that the feature vector extracted from the anomalous sound will be quite different from the feature vector extracted from the normal sound. Specifically, we extracted feature vectors from classifiers and autoencoders trained in a self-supervised manner, respectively. Since there are only normal sounds during training, the distance between anomalous sounds in the test set and the normal samples in the training set will be relatively large. On the contrary,

the distance between the normal sounds in the test set and the normal samples in the training set will be relatively small.

For each sample in the test set, first, we calculated the cosine distance between it and all samples in the training data that belong to the same section ID as this sample, and sorted them from near to far. Then, we tried two distance measures: 1) the closest distance; 2) the average of the top 5 closest distances. We used one of these two distances to calculate the anomaly score for each machine. Obviously, the anomaly samples in the test set are farther away from the normal samples in the training set, so it has a higher anomaly score. Table 4 and Table 5 show the results of using distance metric based on self-supervised classifier and autoencoder.

Table 4: Distance Metric Scoring Results Based on Self-Supervised Classification(%)

	ToyCar	**ToyTrain**	**bearing**	**fan**	**gearbox**	**slider**	**valve**
Distance metric	Top1	Top1	Top1	Average of the top 5	Top1	Average of the top 5	Average of the top 5
h-mean score	56.60	54.50	66.00	80.90	77.50	87.40	79.40

Table 5: Distance Metric Scoring Results Based on Autoencoder(%)

	ToyCar	**ToyTrain**	**bearing**	**fan**	**gearbox**	**slider**	**valve**
Distance metric	Top1	Top1	Average of the top 5	Top1	Top1	Top1	Top1
h-mean score	64.30	49.50	57.30	50.20	63.10	63.90	51.10

where "h-mean score" is the harmonic mean of h-mean AUC (source), h-mean AUC (target) and h-mean pAUC for each machine. "Top1" refers to taking the nearest cosine distance to the test sample, and "Average of the top 5" refers to taking the average of the top 5 nearest cosine distances.

2.5. Ensemble

Obviously, the results of the above four anomalous sound detection methods are complementary, so we can ensemble them. We combined these four models by searching a convex combining grid, similar to [12]. We explored four different sets of weights as the final four systems submitted, and Table 6 shows the results.

3. CONCLUSIONS

This paper presents an ensemble approach for anomalous sound detection based on self-supervised classification, autoencoder, binary classification and distance metric. Experimental results show that by integrating our different methods, we can achieve better results than the baseline. We believe this is because each method focuses on different features to detect anomalous sounds, and they have strong complementarity, so it is important to integrate models that focus on different features. Future work includes developing more "internal modeling" (IM) based [13] anomalous sound detection methods and more effective approaches to deal with domain generalization.

4. REFERENCES

[1] Kota Dohi, Keisuke Imoto, Noboru Harada, Daisuke Niizumi, Yuma Koizumi, Tomoya Nishida, Harsh Purohit, Takashi Endo, Masaaki Yamamoto, and Yohei 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.

[2] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "Toyadmos2: Another dataset of miniaturemachine operating sounds for anomalous sound detection under domain shift conditions," in *Proc. 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021, pp. 1–5.

[3] 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.

[4] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 9758–9769.

[5] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, 2018.

[6] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Unsupervised Anomalous Sound Detection using Self-Supervised Classification and Group Masked Autoencoder for Density Estimation," *Tech. report in DCASE2020 Challenge Task 2*, 2020.

[7] J. Lopez, G. Stemmer, and P. Lopez-Meyer, "Ensemble of complementary anomaly detectors under domain shifted conditions," *Tech. report in DCASE2021 Challenge Task 2*, 2021.

[8] Y. Liu, J. Guan, Q. Zhu and W. Wang, "Anomalous Sound Detection Using Spectral-Temporal Information Fusion," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 816-820.

[9] Gong, Yuan and Lai, Cheng-I Jeff and Chung, Yu-An and Glass, James, "SSAST: Self-Supervised Audio Spectrogram Transformer." *arXiv preprint arXiv:2110.09784*, 2021.

[10] J. F. Gemmeke et al., "Audio Set: An ontology and human-labeled dataset for audio events," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776-780.

[11] Ribeiro, Alexandrine, et al. "Deep dense and convolutional autoencoders for unsupervised anomaly detection in machine condition sounds." *arXiv preprint arXiv:2006.10417*, 2020.

[12] P. Daniluk, M. Gozdziewski, S. Kapka, and M. Kosmider, "Ensemble of auto-encoder based systems for anomaly detection," *Tech. report in DCASE2020 Challenge Task 2*, 2020.

[13] 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.

Table 6: *DCASE 2022 Task 2 experimental results on development dataset(%). The value in the row "Total score" represents the harmonic mean of the AUC and pAUC scores over all the machine types, sections, and domains.*

		Baseline(MNv2)	Baseline(AE)	System1	System2	System3	System4
ToyCar	Self-supervised classification weight	-	-	0.1	0.2	0.2	0.3
	Autoencoder weight	-	-	0.4	0.3	0.3	0.2
	Binary classification weight	-	-	0.5	0.5	0.5	0.5
	Distance metric weight	-	-	0	0	0	0
	h-mean AUC(source)	59.17	91.41	76.42	73.69	73.69	71.49
	h-mean AUC(target)	52.28	35.01	83.76	82.68	82.68	81.14
	h-mean pAUC	52.31	52.70	63.52	62.00	62.00	61.25
ToyTrain	Self-supervised classification weight	-	-	0.8	0.7	0.8	0.7
	Autoencoder weight	-	-	0	0	0	0
	Binary classification weight	-	-	0.1	0.2	0.2	0.3
	Distance metric weight	-	-	0.1	0.1	0	0
	h-mean AUC(source)	58.32	76.32	73.50	74.63	73.18	73.96
	h-mean AUC(target)	46.19	23.51	64.61	63.97	64.22	63.22
	h-mean pAUC	51.56	50.50	58.73	59.41	58.52	58.94
bearing	Self-supervised classification weight	-	-	0.2	0.2	0.2	0.3
	Autoencoder weight	-	-	0	0	0	0
	Binary classification weight	-	-	0.8	0.8	0.8	0.7
	Distance metric weight	-	-	0	0	0	0
	h-mean AUC(source)	62.89	54.45	85.20	85.20	85.20	84.83
	h-mean AUC(target)	61.72	58.66	88.11	88.11	88.11	87.19
	h-mean pAUC	57.57	52.03	67.75	67.75	67.75	67.88
fan	Self-supervised classification weight	-	-	0.4	0.4	0.3	0.5
	Autoencoder weight	-	-	0	0	0	0
	Binary classification weight	-	-	0.2	0.2	0.2	0.3
	Distance metric weight	-	-	0.4	0.4	0.5	0.2
	h-mean AUC(source)	71.35	78.59	96.18	96.18	96.01	96.42
	h-mean AUC(target)	48.54	47.23	84.49	84.49	84.49	84.45
	h-mean pAUC	57.05	57.53	80.25	80.25	79.99	80.15
gearbox	Self-supervised classification weight	-	-	0	0.2	0.2	0.3
	Autoencoder weight	-	-	0.4	0.3	0.3	0.3
	Binary classification weight	-	-	0.1	0.2	0.2	0.3
	Distance metric weight	-	-	0.5	0.3	0.3	0.1
	h-mean AUC(source)	69.60	68.94	93.34	93.25	93.25	92.52
	h-mean AUC(target)	56.57	62.64	80.06	77.26	77.26	75.65
	h-mean pAUC	56.13	58.50	66.71	67.03	67.03	66.65
slider	Self-supervised classification weight	-	-	0.6	0.7	0.7	0.3
	Autoencoder weight	-	-	0.3	0.1	0.1	0
	Binary classification weight	-	-	0.1	0.2	0.2	0.3
	Distance metric weight	-	-	0	0	0	0.4
	h-mean AUC(source)	66.13	77.95	97.64	98.06	98.06	98.61
	h-mean AUC(target)	40.58	47.70	86.69	86.30	86.30	85.32
	h-mean pAUC	54.70	55.78	83.38	82.76	82.76	80.86
valve	Self-supervised classification weight	-	-	0.7	0.4	0.4	0.5
	Autoencoder weight	-	-	0	0	0	0
	Binary classification weight	-	-	0.1	0.2	0.2	0.3
	Distance metric weight	-	-	0.2	0.4	0.4	0.2
	h-mean AUC(source)	67.18	52.04	90.38	90.06	90.06	89.36
	h-mean AUC(target)	57.49	49.47	90.86	90.91	90.91	90.75
	h-mean pAUC	62.49	50.36	83.29	82.95	82.95	82.94
Total score		56.58	52.71	79.11	78.71	78.55	78.02