UNSUPERVISED ANOMALOUS SOUND DETECTION USING MULTIPLE TIME-FREQUENCY REPRESENTATIONS

Technical Report

Sergey Verbitskiy[1]*, Milana Shkhanukova*[2]*, Viacheslav Vyshegorodtsev*[3]

Deepsound
[1]s.verbitskii@alumni.nsu.ru [2]milana.shxanukova@mail.ru [3]vyshegorodtsevslava@gmail.com

ABSTRACT

This technical report describes our approach for the DCASE2022 Challenge Task 2. This task aims to continue research on unsupervised anomalous sound detection and develop new high-performing systems for monitoring the condition of machines. In contrast to the DCASE2021 Challenge Task 2, the 2022 task primarily focuses on domain generalization. First and foremost, we propose the idea of using ensembles of 2D CNN-based systems that utilize different time-frequency representations as input features. We use normal sound clips and their section indices to train our anomalous sound detection (ASD) systems for each machine type, and embedding vectors extracted from our CNNs, cosine similarity, and the k-nearest neighbors algorithm (k-NN) to calculate the anomaly scores of test clips. As a result, our method achieves the official score of 0.725 on the development dataset and significantly outperforms the baseline systems.

Index Terms— anomalous sound detection, time-frequency representations, convolutional neural networks, log mel spectrogram, MFCC, GFCC

1. INTRODUCTION

The DCASE2022 Challenge Task 2 (Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques) [1] focuses on the detection of mechanical failure by observing sounds. In unsupervised anomalous sound detection tasks, it is supposed that anomaly detectors are trained exclusively with normal sound clips without anomalies, but after training, are capable of correctly detecting anomalies [1, 2]. Thus, training datasets for these tasks consist of only normal samples [3, 4], and systems are trained in a self-supervised learning manner.

The main difference between the 2022 task and the DCASE2021 Challenge Task 2 [2] is that the source/target domain of each sample from the evaluation dataset is not specified. Thus, it complicates the development of high-performing systems because, in contrast to the 2021 task, researchers are not able to design various algorithms for various domains or use domain adaptation techniques. The 2022 task aims to the development of domain generalization techniques and versatile systems that are mainly trained with the source domain data and are capable of using for the target domain data without domain adaptation.

In previous studies on unsupervised anomalous sound detection, developed systems were based on either autoencoders [5] or classification neural networks [6, 7, 8]. The main idea of autoencoders is to encode and reconstruct audio clips using a low-dimensional space [1]. An autoencoder (AE) is trained with normal sounds, so normal audio clips are generally reconstructed better than abnormal ones. Hence, the anomaly score can be calculated using the reconstruction error. The second approach is based on using classification systems that are trained to distinguish among audio clips by their attributes, which are utilized as classes. Then, the anomaly scores are calculated as the error in the identification of the correct class [1] or through other alternative ways. The most popular alternative way is the use of the average value of cosine or euclidean distance to k-nearest training samples with the same attributes in an embedding space [7, 8].

In this work, we employ the second approach, sections as audio attributes, and ArcFace [9] as a loss function to train our CNN-based systems. Moreover, to calculate the anomaly scores of test clips, we use embeddings extracted from systems and cosine similarity from them to k-nearest embeddings of training clips with appropriate sections.

In addition, we consider the log mel spectrogram (LMS) [10], the Mel-Frequency Cepstral Coefficients (MFCC) [11, 10], and the Gammatone Frequency Cepstral Coefficients [12, 13] as three time-frequency representations, which are used as input features to models. Then, we combine these models to build ensembles that enables to detect anomalies on audio clips better. Since the sounds of different machines have different sound characteristics, it makes sense to use the most appropriate representation for each machine type. Moreover, ensembles of models with various representations as input can contain the advantages of each representation.

For instance, in [14] was shown that models, which utilize multiple feature channels consisting of various audio signal representations as input, achieve higher performance than models with single representations. In contrast to [14], we combine models, which use single feature channels, and use decision-level fusion at the end as in [15] (we use the weighted average of the anomaly scores) to obtain predictions. The main reason is that there are different frequency scales on various representations, and combining them in one multiple channel input can lead to incorrespondence between frequency dimensions, and therefore, to the incorrect extraction of local spatial features with convolutional layers.

As a backbone architecture, we employ ERANNs (Efficient Residual Audio Neural Networks) [16]. ERANNs are CNN-based systems for audio pattern recognition tasks that utilize time-frequency representations as input (the log mel spectrogram in the original paper). ERANNs are based on ResNets [17] and WideResNets [18]. These systems achieve high performance on diverse audio pattern recognition tasks while having low computational complexity compared with other CNN-based systems.

The remainder of this technical report is organized as follows:

Section 2 describes the feature extraction process, data augmentation techniques, the proposed architecture of CNN-based systems, and the method of obtaining the anomaly scores. Section 3 provides experimental results on the development dataset, Section 4 desribes our submissions, and Section 5 concludes the technical report.

2. ANOMALOUS SOUND DETECTION SYSTEM

2.1. Feature extraction

In this study, we use multiple time-frequency representations (LMS, MFCC, and GFCC) as input features to our models.

We apply the short-time Fourier transform (STFT) with the Hann window of size 1024 and the hop size of 256 to extract all representations. The number of frequency bins for each time-frequency representation is 256. We also use a sampling rate (sr) of 16 kHz for all audio clips (we do not apply resampling methods). Moreover, we adopt the lower cut-off frequency $f_{min} = 50$ Hz and the upper cut-off frequency $f_{max} = 7500$ Hz. Thus, the shape of each time-frequency representation for 10-second audio clips equals 626×256.

Instead of the logarithm, we use the cube root of the values of the mel spectrogram to extract MFCC as in [13]. We have conducted experiments and figured out that MFCC with the cube root outperforms default MFCC.

2.2. Data augmentation techniques

We apply two data augmentation techniques to prevent models from overfitting during training:

- **temporal cropping**: during training models, we use 3-second sections of audio clips that are cut from random places. During evaluating models, we utilize full 10-second audio clips without the temporal cropping to obtain embeddings.
- **SpecAgment** [19]: we also use SpecAugment for frequency and time masking on time-frequency representations of training audio clips. SpecAugment is applied with two time masks with a maximum length of 32 frames and two frequency masks with maximum length of 32 bins.

2.3. Neural network architecture

In this work, we use ERANN-2-0 ($W = 2$ and $s_m = 0$) [16] as a backbone architecture to build our systems. The architecture of ERANN-2-0 is described in Table 1.

Table 1: Architecture of ERANN-2-0

Blocks/Layers	Stride	Kernel	Output size
batchnorm	–	–	$626 \times 256 \times 1$
residual block $\times$ 4	1×1	3×3	$626 \times 256 \times 16$
residual block $\times$ 4	2×2	3×3	$313 \times 128 \times 32$
residual block $\times$ 4	2×2	3×3	$156 \times 64 \times 64$
residual block $\times$ 4	2×2	3×3	$78 \times 32 \times 128$
residual block $\times$ 4	2×2	3×3	$39 \times 16 \times 256$
global pooling	–	–	$1 \times 1 \times 256$
flatten	–	–	256
fully connected	–	–	256

Residual blocks, which based on the basic blocks of ResNet-V2 [20], contains two or three 2D convolutional layers, two batch-norm layers [21], and two non-linear activation functions (Leaky ReLU with $\alpha = 0.01$). The first batch-norm layer is used for the frequency-wise normalization. In global pooling, we apply a combination of average and max pooling as in [22].

Thus, ERANN-2-0 is used to derive embeddings of size 256 that represent audio clips. These embeddings are utilized as input features to ArcFace, as well as for calculating the anomaly scores.

2.4. Additive Angular Margin Loss

To train our systems we employ Additive Angular Margin Loss (ArcFace) [9] as a loss function. This loss function was applied in many approaches for previous unsupervised anomalous sound detection tasks [6, 8], and ADS systems trained with ArcFace demonstrated better performance than systems trained with standard softmax losses. This loss ensures a margin between classes that enhances the intra-class compactness and the inter-class separability of extracted embeddings. Moreover, in contrast to other standard losses, this loss contains an additional fully connected layer with trainable parameters.

We use all the six sections from the development dataset combined with the additional training dataset to train our ASD systems using ArcFace. We adopt $m = 0.09$ (the angular margin penalty) and $s = 40$ (the re-scale factor) for all experiments.

2.5. Calculating anomaly scores

To calculate the anomaly scores of test clips we use cosine similarity and the k-nearest neighbors algorithm [23]. The underlying idea is to consider the value of the anomaly score of an audio clip as the average cosine similarity (with a minus sign) from the embedding of this audio clip to the k-closest embeddings of training clips from the same section. We adopt $k = 1$ for all experiments.

In addition, we use the logarithm to have a greater difference in absolute value between the anomaly scores S:

$$S = \log(1 - C), \tag{1}$$

where $C \in [-1, 1]$ is the cosine similarity between two embeddings.

2.6. Ensemble strategy

Ensemble techniques are proved to be effective in the previous DCASE challenges [2]. In previous works, ensemble techniques were mainly used for the combining of models that have different hyperparameters or architectures [6, 7].

We propose another ensemble strategy that is supposed to combine models with the same architecture but with different time-frequency representations (LMS, MFCC, GFCC) as their input.

To build ensembles and calculate the final anomaly score of an audio clip we employ the weighted average:

$$S = \sum_{i=1}^{n} w_i \cdot S_i, \qquad \sum_{i=1}^{n} w_i = 1, \tag{2}$$

where S is the final anomaly score of the audio clip, S_i are the anomaly scores of the audio clip calculated using the $i-$th model with the $i-$th time-frequency representation as input, w_i are weights, and n is the number of models (we use $n \leq 3$).

3. EXPERIMENTS AND RESULTS

3.1. Training setup

Parameters of models are optimized with the Adam optimizer [24] with the learning rate of 0.0002 and with a mini-batch size of 32. We also use an exponential moving average (EMA) of model parameters with a decay rate of 0.999. We do not apply any schedulers. We train all models for 100 epochs and use early stopping to obtain final models with the best validation scores (the harmonic mean of the AUC and pAUC scores over all sections and domains) for each machine type.

3.2. Dataset

To train and evaluate ASD systems development (including the additional training dataset) and evaluation datasets are provided [1, 3, 4]. The development dataset contains training and test data. The training data consists of seven types of machines ("ToyCar", "ToyTrain", "bearing", 'fan", "gearbox", "slider", and "valve"), and each type of machine consists of six sections (Sections 00, 01, 02, 03, 04, and 05). In the training data, for each section and machine type, there are 990 clips of normal sounds in the source domain and only ten clips of normal sounds in the target domain.

The test data (the part of the development dataset) comprises the same machine types, the first three sections, 50 clips of normal and 50 clips of abnormal sounds for each section, domain, and machine type.

The evaluation dataset consists of the last three sections (Sections 03, 04, and 05), has the same number of audio clips as the test data, and, in contrast to the development dataset, does not contain any information about the domain.

All audio recordings have a length of 10 seconds and a sampling rate of 16 kHz.

3.3. Results

The results of our ASD systems on the development dataset are demonstrated in Table 2, Table 3, Table 4, and Table 5. In the tables, we compare single models ERANN-S, the ensemble ERANN-M containing two models, and the ensemble ERANN-L containing three models with various time-frequency representations as input. We also compare our systems with the baseline systems [1]. All systems consist of ERANN-2-0 models.

Table 2, Table 3, and Table 4 contain the values of the harmonic mean of the AUC and pAUC ($p = 0.1$) scores over all the sections for each machine type and domain. Table 5 shows the values of the official score Ω that is equal to the harmonic mean of the AUC and pAUC scores over all the machine types, sections, and domains. In addition, in Table 6, we detail the values of the weights w_i for our ensembles ERANN-M and ERANN-L.

Moreover, Table 5 contains results of the single model ERANN-S-B that use the most performed time-frequency representation for each machine type: the log mel spectrogram for ToyCar, bearing, fan, and valve, MFCC for gearbox and slider, and GFCC for ToyTrain.

We show that ERANN-S with a specific time-frequency representation as input performs better for a specific machine type. For instance, the log mel spectrogram (LMS) is the best representation for bearing, fan, and valve. On the other hand, MFCC is better than LMS for gearbox and slider. Thus, analysis of multiple time-frequency representations and choosing the best one for each machine type can be effective for anomalous sound defections tasks. From table 5, we can also see that ensembles of models outperform single models by a large margin.

As can be seen from the tables, our ASD systems significantly outperform the baseline systems. The best ensembles ERANN-L, which consist of three models, achieve the official score of 0.725.

4. SUBMISSIONS

In total, we submit results of four ASD systems on the evaluation dataset:

- **submission 1**: the results of single models ERANN-S with the log mel spectrogram as input;
- **submission 2**: the results of single models ERANN-S-B with the most performed time-frequency representation as input;
- **submission 3**: the results of ensembles ERANN-M (two models);
- **submission 4**: the results of ensembles ERANN-L (three models).

5. CONCLUSION

In this technical report, we have proposed the technique of using an ensemble of 2D CNN-based systems with various time-frequency representations as input features for the DCASE2022 Challenge Task 2. We have proven by the experiments that this technique is effective for anomalous sound detection tasks.

We have shown that our anomalous sound detection systems have significantly better performance than the baseline systems. Our best ensembles achieve the official score of 0.725 on the development dataset.

6. REFERENCES

[1] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and Discussion on DCASE 2022 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques," *In arXiv e-prints: 2206.05876*, 2022.

[2] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and Discussion on DCASE 2021 Challenge Task 2: Unsupervised Anomalous Detection for Machine Condition Monitoring Under Domain Shifted Conditions," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 186–190.

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

[4] 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.

[5] I. Kuroyanagi, T. Hayashi, Y. Adachi, T. Yoshimura, K. Takeda, and T. Toda, "An Ensemble Approach to Anomalous Sound Detection Based on Conformer-Based Autoencoder and Binary Classifier Incorporated with Metric Learning," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 110–114.

[6] J. A. Lopez, G. Stemmer, P. Lopez Meyer, P. Singh, J. Del Hoyo Ontiveros, and H. Cordourier, "Ensemble Of Complementary Anomaly Detectors Under Domain Shifted Conditions," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 11–15.

[7] K. Wilkinghoff, "Combining Multiple Distributions based on Sub-Cluster AdaCos for Anomalous Sound Detection under Domain Shifted Conditions," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 55–59.

[8] K. Morita, T. Yano, and K. Tran, "Anomalous Sound Detection Using CNN-based Features by Self Supervised Learning," DCASE2021 Challenge, Tech. Rep., July 2021.

[9] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[10] S. K. Kopparapu and M. Laxminarayana, "Choice of Mel filter bank in computing MFCC of a resampled speech," 05 2010, pp. 121–124.

[11] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," *Proc. 1st Int. Symposium Music Information Retrieval*, 11 2000.

[12] B. Ayoub, K. Jamal, and Z. Arsalane, "Gammatone frequency cepstral coefficients for speaker identification over VoIP networks," in *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, 2016, pp. 1–5.

[13] X. Zhao and D. Wang, "Analyzing noise robustness of MFCC and GFCC features in speaker identification," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7204–7208.

[14] J. Sharma, O.-C. Granmo, and M. Goodwin, "Environment Sound Classification Using Multiple Feature Channels and Attention Based Deep Convolutional Neural Network." in *Interspeech*, 2020, pp. 1186–1190.

[15] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion," *Sensors*, vol. 19, no. 7, 2019.

[16] S. Verbitskiy, V. Berikov, and V. Vyshegorodtsev, "ERANNs: Efficient Residual Audio Neural Networks for Audio Pattern Recognition," *In arXiv e-prints: 2106.01621*, 2021.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[18] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," 2017.

[19] W. C. Daniel S. Park, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH*, 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Computer Vision – ECCV 2016*, 2016, pp. 630–645.

[21] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 07–09 Jul 2015, pp. 448–456.

[22] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[23] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient Algorithms for Mining Outliers from Large Data Sets," *SIGMOD Rec.*, vol. 29, no. 2, p. 427–438, may 2000.

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Table 2: Harmonic Mean of AUC for the source domain on Development Dataset

	Input Features			ToyCar	ToyTrain	bearing	fan	gearbox	slider	valve	h-mean
	LMS	MFCC	GFCC								
baseline (AE) [1]				**0.904**	0.763	0.544	0.786	0.689	0.780	0.520	0.687
baseline (CNN) [1]	✓			0.591	0.573	0.606	0.708	0.692	0.652	0.671	0.638
ERANN-S	✓			0.807	0.768	0.783	0.698	0.715	**0.973**	0.933	0.800
		✓		0.713	0.767	0.674	0.705	0.884	0.959	0.903	0.787
			✓	0.728	**0.863**	0.627	0.686	0.859	0.930	0.895	0.783
ERANN-M	✓	✓		0.790	0.783	0.785	0.823	0.884	0.961	0.950	0.848
ERANN-L	✓	✓	✓	0.797	0.849	**0.806**	**0.823**	**0.884**	0.961	**0.951**	**0.863**

Table 3: Harmonic Mean of AUC for the target domain on Development Dataset

	Input Features			ToyCar	ToyTrain	bearing	fan	gearbox	slider	valve	h-mean
	LMS	MFCC	GFCC								
baseline (AE) [1]				0.348	0.234	0.584	0.472	0.626	0.477	0.495	0.419
baseline (CNN) [1]	✓			0.520	0.459	0.599	0.482	0.562	0.382	0.572	0.500
ERANN-S	✓			0.693	**0.462**	0.835	0.589	0.684	0.663	0.859	0.658
		✓		0.749	0.445	0.714	0.543	0.813	**0.703**	0.815	0.654
			✓	0.667	0.422	0.697	0.543	0.745	0.623	0.844	0.621
ERANN-M	✓	✓		**0.763**	0.458	**0.836**	0.594	0.813	0.702	0.868	**0.687**
ERANN-L	✓	✓	✓	0.762	0.436	0.835	**0.594**	**0.813**	0.702	**0.876**	0.681

Table 4: Harmonic Mean of pAUC on Development Dataset

	Input Features			ToyCar	ToyTrain	bearing	fan	gearbox	slider	valve	h-mean
	LMS	MFCC	GFCC								
baseline (AE) [1]				0.527	0.505	0.520	0.575	0.585	0.558	0.504	0.537
baseline (CNN) [1]	✓			0.523	0.515	0.571	0.569	0.560	0.547	0.624	0.557
ERANN-S	✓			0.598	0.520	**0.694**	**0.669**	0.599	0.636	0.793	0.634
		✓		0.616	0.539	0.603	0.555	0.728	**0.664**	0.753	0.628
			✓	0.595	0.532	0.598	0.605	0.677	0.543	0.791	0.611
ERANN-M	✓	✓		0.620	0.533	0.693	0.637	0.728	0.663	0.806	0.659
ERANN-L	✓	✓	✓	**0.621**	**0.540**	0.688	0.637	**0.730**	0.663	**0.831**	**0.662**

Table 5: Official score Ω on Development Dataset

	Input Features			Official Score
	LMS	MFCC	GFCC	Ω
baseline (AE) [1]				0.527
baseline (CNN) [1]	✓			0.566
ERANN-S	✓			**0.690**
		✓		0.684
			✓	0.663
ERANN-S-B	✓	✓	✓	**0.712**
ERANN-M	✓	✓		0.722
ERANN-L	✓	✓	✓	**0.725**

Table 6: Weights for ensembles

Machine Type	ERANN-M		ERANN-L		
	LMS	MFCC	LMS	MFCC	GFCC
ToyCar	0.56	0.44	0.51	0.36	0.13
ToyTrain	0.59	0.41	0.03	0.42	0.55
bearing	0.99	0.01	0.89	0.01	0.10
fan	0.20	0.80	0.21	0.78	0.01
gearbox	0.01	0.99	0.01	0.97	0.02
slider	0.01	0.99	0.01	0.98	0.01
valve	0.51	0.49	0.45	0.32	0.23