ENSEMBLE OF MULTIPLE ANOMALOUS SOUND DETECTORS

Yufeng Deng[1], *Anbai Jiang*[1], *Yuchen Duan*[1], *Jitao Ma*[3], *Xuchu Chen*[1], *Jia Liu*[1,2], *Pingyi Fan*[1], *Cheng Lu*[3], *Wei-Qiang Zhang*[1]

[1] Department of Electronic Engineering, Tsinghua University, Beijing, China
[2] Tsinghua AI Plus, Beijing, China
[3] School of Economics and Management, North China Electric Power University, Beijing, China
dyf20@mails.tsinghua.edu.cn, {liuj, fpy, wqzhang}@tsinghua.edu.cn, lucheng1983@163.com

ABSTRACT

This paper presents our submission to DCASE 2022 Challenge Task 2, which aims to detect anomalous machine status via sounding by using machine learning methods, where the training dataset itself does not contain any examples of anomalies. We build six subsystems, including three self-supervised classification methods, two probabilistic methods and one generative adversarial network (GAN) based method. Our final submissions are four ensemble systems, which are different combinations of the six subsystems. The best official score of the ensemble systems can achieve 86.81% on the development dataset, whereas the corresponding Autoencoder-based baseline and the MobileNetV2-based baseline are with scores of 52.61% and 56.01%, respectively. In addition, our methods rank top on the development dataset and fourth on the evaluation dataset in this challenge.

***Index Terms*—** DCASE, anomaly detection, domain generalization, machine condition monitoring, machine health monitoring

1. INTRODUCTION

The DCASE 2022 Challenge Task 2 is concerned with detecting anomalous state of the target machine using the sounding data. Unlike the acoustic scene classification, the available training data in this task contains samples of only one class — the normal-state class, but the aim is to detect whether a test sample is in another class, refer to as anomaly class, which may include various anomalous situations. A further complication added to this challenge is that the distributions of the training data and of the test data are different. This is called as domain shift. In the literature, there are some works investigating how to solve the domain shift problem by using machine learning methods and reducing the performance gap between the training and test data [1]. Although these techniques achieved impressive performance on image classification, they did not generally gain the expected and comparable results in the machine status detection via sounding up to now.

In this paper, we present six subsystems, the first three are self-supervised classifiers trained by using the supervision information provided by the metadata, similar to the approach taken by several teams at DCASE 2021 [2, 3] and DCASE 2020 [4]. The fourth and fifth models are probabilistic models. For the fourth model, inspired by the probabilistic model in [2], we employ normalizing flows to estimate the conditional density of the feature vectors for each section where the Mel spectrograms are used as the input of the pooling layers. Those output above the defined threshold will be marked as anomaly. The fifth model estimates the conditional density of spectrogram target frames conditioned on remaining frames using an RNN based model and a GMM loss. The sixth model is a GAN based model.

We next present the results of the baseline systems in Section 2, and then describe each of our subsystems in detail in Section 3. For each subsystem, we will describe how it is trained and present its results on the development dataset. After that, we present our ensemble systems and their detection results in Section 4. Finally, conclusions are drawn in Section 5.

2. BASELINE RESULTS

In order to give a clear picture of the Challenge Task 2, we include the baseline scores on Table 1 and Table 2. To present the results succinctly, the results in all tables in this paper present only the harmonic mean of the source AUC, target AUC and pAUC for each machine type on the development dataset. Here the harmonic mean is denoted as h-mean. The data used in this challenge is 16 kHz, single-channel audio. For more details, please see [5, 6, 7].

Table 1: baseline-AE results

bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve	h-mean
54.80%	58.47%	63.07%	57.99%	51.06%	39.61%	50.59%	52.61%

Table 2: baseline-MobileNetV2 results

bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve	h-mean
59.16%	57.21%	59.91%	50.26%	54.23%	51.18%	62.42%	56.01%

3. APPROACHES

The general idea of the first three approaches is to first train a neural network to extract the embeddings of the samples by classifying labels extracted from the metadata, and then use the outlier detection algorithm to score how abnormal the embeddings are. The input to the first two models is a spectrogram with or without a Mel transformation, the difference between the first two models is that the first model uses a single index loss detection, while the second model uses a multiple indices loss detection. For the third model, the inputs are the embeddings extracted by the pre-trained wav2vec [8],

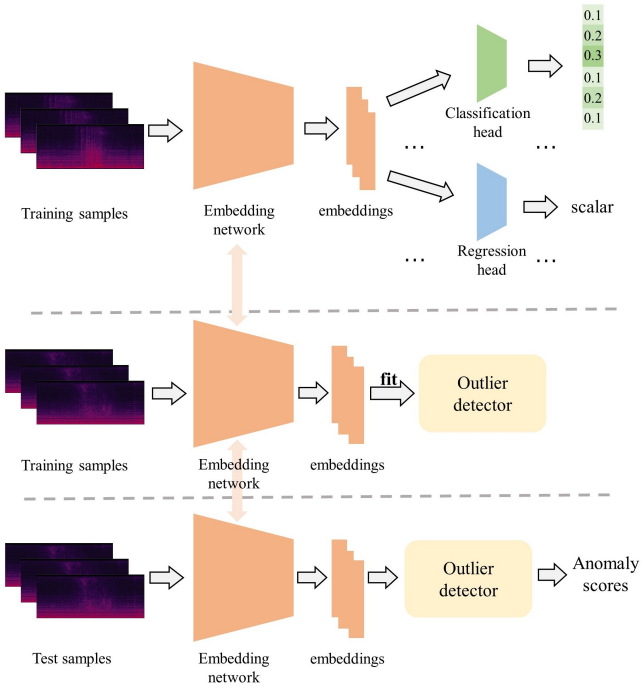

Figure 1: Overview of the classification based method.

which is trained with a partial Audioset [9]. The loss function for the first three models is ArcFace [10]. The fourth and fifth models differ from the first three in that they are not trained using any metadata information, which makes them completely unsupervised. The fourth model attempts to learn several distributions of some feature vector bins conditioned on the remaining bins. The fifth model estimates the distribution of target frames for the spectrogram conditioned on the remaining frames. The sixth model is a GAN based model. After describing these six subsystems, we present four ensembles of these subsystems, which are our final submission.

3.1. Classification Based Methods

In this subsection, we describe the first three subsystems, which we call SC (**S**ection **C**lassification), MHCR (**M**ulti-**h**ead **C**lassification & **R**egression) and SC-wav2vec. All of the three subsystems follow the overview shown in Figure 1.

The overview shown in Figure 1 is divided into three processes. First, we use the training samples to train the embedding network, and then use the trained embedding network to extract the embeddings of the training samples. Later on, we use these embeddings to train the outlier detector. Finally, the trained embedding network is used to extract the embeddings of the test samples, and the trained outlier detector is used to score the abnormality of these embeddings. In Figure 1, the embedding is extracted from the output of the last or penultimate layer of the embedding network. Since the ArcFace [10] loss function increases the inter-class distance and decreases the intra-class distance so that the network learns a better representation of the data, an ArcFace [10] layer is usually chosen for the classification head. The regression header is a fully connected layer.

3.1.1. Features & Training

The input feature of the embedding network used in the SC and MHCR is STFT spectrogram with or without a Mel transformation. For the SC-wav2vec, the input feature is the embedding extracted from a pre-trained wav2vec [8], which is trained using a partial AudioSet [9]. The logarithm is taken for the Mel spectrogram but not taken for the STFT spectrogram. We select the optimal STFT frame length, hop length, and number of frames based on the results on the development dataset and decide whether to use the Mel transform for each machine type. The specific feature parameters are shown in Table 3.

For the SC, the embedding network is trained to predict the section IDs using ArcFace [10] loss function. Since only the section ID metadata is used, there is only one classification head in the SC. For the MHCR, we additionally use other tags in the filenames to design classification (factory noise, microphone position, etc.) or regression (speed, weight, etc.) tasks. In order to achieve multi-label classification or regression, multiple parallel classification or regression heads are used. Different tasks are trained simultaneously. The hyperparameter λ is used to balance multiple losses, as shown in (1).

$$\mathcal{L} = \sum_i \lambda_i \mathcal{L}_i \tag{1}$$

where $\lambda_i > 0$ and $\mathcal{L}_i$ is the loss of label i. For the SC-wav2vec, we first train wav2vec [8] model using Fairseq [11] on the balanced AudioSet [9], while the features extracted using the pre-trained wav2vec [8] are input to the embedding network. The supervised label for the SC-wav2vec is the section IDs, hence, only one classification head is used in the SC-wav2vec.

Based on various experiments using the training dataset provided in section 0-2, we observe that the architecture of the embedding network has a significant impact on the performance and the optimal network architecture is different for different machine types, so we select the best performing network from MobileFaceNet (MFN) [12], MFNSE, Ecapa-tdnn [13] and Cnn6 for each machine type. The MFNSE is an improved network from the MFN, the difference between the MFNSE and MFN is that we add a squeeze-excitation [14] block to the bone block of the MFN. The Cnn6 is a 6-layer convolutional network used in [15].

For the training of embedding network, we adopt the AdamW optimizer with the default learning rate of 1×10^{-3}, weight decay 1×10^{-4}, and 25 epochs for training, where all the training data is drawn from the development and evaluation datasets. When training the wav2vec [8] model on the balanced AudioSet [9], we use the initial learning rate of 1×10^{-7} and linearly increase the learning rate to 5×10^{-3} in the first 500 updates, then decay to 1×10^{-6} along the cosine curve.

3.1.2. Anomaly Detection Algorithms

Because the training samples only contain the normal ones, outlier detection algorithms are used to detect anomalous samples. Once the training is completed, the embedding network is used to extract the embeddings of the input samples, which are used to fit the outlier detectors. For the outlier detection, we employ four well known algorithms, k-NN [16], LOF [17], cosine distance and Mahalanobis distance. For the cosine distance and Mahalanobis distance, we compute the average embedding using the embeddings of the training samples, and additionally compute the covariance matrix for the Mahalanobis distance. In the test phase, the average

embedding and covariance matrix are used to compute the cosine and Mahalanobis distances of the test embedding, and use them to present the anomaly scores.

3.1.3. Results

Table 3 shows the results of the SC, Table 4 shows the results of the MHCR, while Table 5 shows the results of the SC-wav2vec.

Table 3: The results of SC

	bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve
feature	STFT	STFT	STFT	STFT	logmel	STFT	STFT
nMels	-	-	-	-	64	-	-
nffts	2048	2048	2048	2048	2048	2048	2048
nFrames	192	192	192	192	192	192	192
network	MFN	MFNSE	MFN	MFNSE	Ecapa-tdnn	Cnn6	MFNSE
h-mean	82.20%	80.64%	83.20%	88.27%	77.50%	75.92%	96.84%

Table 4: The results of MHCR

bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve	h-mean
72.07%	72.16%	84.01%	83.18%	71.10%	68.14%	95.28%	77.01%

3.2. Probabilistic Models

In this subsection, we describe the fourth and fifth subsystems, which we call WSP-NFCDEE and IMDN (**I**nterpolation **M**ixture **D**ensity **N**etwork).

The WSP-NFCDEE is built on the NFCDEE proposed in [2]. The difference between the NFCDEE and the WSP-NFCDEE is that we add a **W**eighted **S**tatistic **P**ooling (WSP) layer before the normalizing flows, which improves the performance on most machine types, especially on the slider. Hence, we call this method WSP-NFCDEE. Let $\boldsymbol{X} \in \boldsymbol{R}^{M \times T}$ is a Mel spectrogram, where M is the number of Mel bins and T is the number of frames. The WSP computes the mean and standard deviation of $\boldsymbol{X}$ along time axis to obtain the mean vector $\boldsymbol{y} \in \boldsymbol{R}^{M}$ and the standard deviation vector $\boldsymbol{z} \in \boldsymbol{R}^{M}$. The output of the WSP is $\alpha \cdot \boldsymbol{y} + \beta \cdot \boldsymbol{z}$, where the α and β are two trainable parameters satisfying the constraints (2).

$$\alpha + \beta = 1, \alpha, \beta > 0 \tag{2}$$

For the IMDN, we adopt three network structures mainly based on CNN and GRU, and a special density estimation loss function that combines the IDNN structure and Gaussian mixture model. We employ three networks named IGNN, LRCGNN, and a simplified DeepFilterNet [18]. Sufficient experiments had shown that networks based on RNN structure perform well on time series inputs. Hence, we select the 3D input with this form, (batch size, nFrames, nMels), which will be later propagated forward in the time dimension by the GRU, while CNN is mainly in the frequency dimension. We first select two lightweight network architectures, IGNN and LRCGNN. These two networks are with relatively low computation complexity, but achieve significant improvements over the Autoencoder-based baseline. The architecture of IGNN is shown in Table 6. The LRCGNN additionally add a Conv1d layer after the first fully connected layer of IGNN.

Table 5: The results of SC-wav2vec

	bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve
Scoring	maha	maha	k-NN	k-NN	LOF	k-NN	cosine
h-mean	58.54%	57.16%	58.01%	66.77%	71.90%	65.03%	68.89%

Table 6: The architecture of IGNN

layer_name	parameters
Fully Connected	(nMels, 128)
3 × GRU	(128, 128)
Fully Connected	(128, 32)
Fully Connected	(32, 128)
3 × GRU	(128, 128)
Fully Connected	((nF-1)×128, 2×nC×nMels)

To achieve a better performance on more complex data, we employ DeepFilterNet (DFnet) [18], a more complex Unet-like network proposed in speech enhancement. The computational complexity of the original DeepFilterNet [18] is high, so we use two fully connected layers in place of the original two convolutional layers, which greatly reduces the amount of computation. Additionally, we add a fully connected layer to match the number of Mel bins.

IDNN [19] was demonstrated to achieve significant improvement on non-stationary signals, which predicted the center frame of the input Mel spectrogram. We also adopt this idea. Another modification is that the Gaussian mixture model is adopted as the loss function. Let $\boldsymbol{x}_p$ is the p-th frame to be predicted. By mapping input features $\boldsymbol{X}$ to the parameters of the GMM, we obtain the frame predictions in probabilistic form (In the test system, the component weights are selected the same value):

$$p\left(\boldsymbol{x}_p|\boldsymbol{X}, D, E\right) = \sum_{m=1}^{C} p_m\left(\boldsymbol{x}_p|\boldsymbol{X}, D, E\right) \tag{3}$$

where C is the number of Gaussian components and p_m is the density of the m-th Gaussian component. Since the aim is to find the maximum probability density of the predicted frame, the LSE function is as follows.

$$\mathcal{L}_{LSE} = -\log \sum_{i=1}^{M} \exp\left(\sum_{m=1}^{C} \log p_m\left(x_p|\boldsymbol{x}, D, E\right)\right) \tag{4}$$

where M is the number of Mel bins. When the covariance matrix of GMM is taken as diagonal matrix, this loss function is proved to be equivalent to maximization of the log likelihood of network with the given data.

3.2.1. Features & Training

The input feature of the WSP-NFCDEE and the IMDN is STFT spectrogram with Mel transformation and the logarithm is taken for the Mel spectrogram. For different machine types we tune the values of nMels, nffts, nFrames.

3.2.2. Results

Table 7 shows the results of the WSP-NFCDEE, while Table 8 shows the results of the IMDN.

Table 7: The results of WSP-NFCDEE

	bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve
nMels	128	128	64	64	64	64	64
nffts	2048	2048	2048	2048	2048	2048	1024
nFrames	100	60	100	100	100	100	30
network	LRCGNN	LRCGNN	DFnet	LRCGNN	LRCGNN	DFnet	IGNN
h-mean	68.12%	68.54%	79.96%	78.52%	65.23%	60.29%	68.27%

Table 8: The results of IMDN

	bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve
nMels	128	128	64	64	64	64	64
nffts	2048	2048	2048	2048	2048	2048	1024
nFrames	100	60	100	100	100	100	30
network	LRCGNN	LRCGNN	DFnet	LRCGNN	LRCGNN	DFnet	IGNN
h-mean	60.92%	62.24%	67.73%	73.17%	68.29%	65.07%	86.19%

3.3. AEGAN-AD

In this subsection, we describe the sixth subsystem, which we call AEGAN-AD.

We design an autoencoder which reconstructs Mel spectrograms and complement it with a discriminator, resulting in a GAN [20] model. Inspired by [21], we adopt a DCGAN [22]-like autoencoder, with the discriminator being our encoder and the generator being our decoder. As deconvolution suffers from checker-board effect, yet this effect is somehow resulting from the periodicity of the spectrogram, which makes reconstruction better than an "upsample-conv" structure. BNs are substituted by LNs in order to promote the detection in the target domain. Since most samples in a batch are from the source domain, it is likely that the network is misled by the biased statistics and only learns the distribution of source domain, resulting in a poor performance for the target domain. LN, which normalizes each sample independently, can learn to transform spectrograms into domain-invariant features. As for ToyCar and gearbox, we pass the latent variable through an adaptive LN which does different affine transformations for different sections. This observation indicates that it could help to transform reconstructed samples to their respective styles as in [23] so that features of different sections can be better represented. The loss function is selected as the MSE. Anomaly detection is conducted not only in input space, but also in the latent space, which is done by sending the reconstructed samples back to the encoder to obtain their latent representations. L1 norm, L2 norm and cosine are utilized to measure the difference of each spectrogram and the overall anomaly score is the mean/min/max of these. We select the best performing metric among these metrics.

For gearbox and slider, a discriminator is introduced to promote the reconstruction, while the autoencoder becomes the generator. The discriminator has the similar architecture with the encoder. It is trained to do a feature level discrimination on the reconstructed samples as a complement for MSE loss. Loss function for the discriminator is WGAN-GP [24] and loss function for the generator is a combination of MSE and feature matching loss [25]. Both of them are shown as (5) and (6), respectively.

$$\begin{aligned}\mathcal{L}_D =&\mathbb{E}_{\hat{x}\sim P_g}[D(\hat{x})] - \mathbb{E}_{x\sim P_r}[D(x)]+ \\ &\lambda\mathbb{E}_{\tilde{x}\sim P_{\tilde{x}}}[(\|\nabla_{\tilde{x}}D(\tilde{x})\|_2 - 1)^2]\end{aligned} \tag{5}$$

$$\mathcal{L}_G = \mathbb{E}_{x\sim P_r}[\|x - G(x)\|_2^2] + \mu\|\mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{\hat{x}\sim P_g}[f(\hat{x})]\|_2^2 \tag{6}$$

Table 9: The results of AEGAN-AD

bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve	h-mean
75.78%	65.83%	71.50%	75.02%	79.16%	58.71%	52.52%	67.04%

where P_r and P_g denote the real distribution and the reconstructed distribution respectively. $P_{\tilde{x}}$ is the linear combination of P_r and P_g. $f(x)$ is the output of the last convolution layer in D. This embedding is also extracted during test time and it is compared with average embedding using k-NN [16], LOF [17], cosine and Mahalanobis distances. We simply choose the best performing metric from both G-based and D-based metrics.

All input for the model is 128×128 Mel spectrogram computed with 2048-point FFT and 512 hop-length. Logarithm is taken first and a MinMaxScaler then scales spectrograms to [-1, 1]. We use an Adam optimizer with a learning rate of 2×10^{-4}. The batch size is set to 512. The model is trained on both development set and evaluation set. The performance is shown in Table 9.

4. SUBMISSION RESULTS

In this subsection, we present the results of ensembles. For the ensembles, we combine the six subsystems by first standardizing the training data scores and then searching over a grid of convex combinations, similar to [2].

The difference between submission-1 and submission-2 is that for submission-1, we additionally train domain classifiers with Cnn6 for sections with obvious domain differences to predict whether the test samples belong to the source domain or the target domain. The test scores for both domains are normalized respectively. For the submission-3, we combine the top performing two or three subsystems for each machine type. For the submission-4, we only combine the SC and WSP-NFCDEE for each machine type. Table 10 shows the results.

Table 10: The results of ensembles

method	bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve	h-mean
submission-1	87.75%	84.98%	87.55%	89.35%	83.52%	78.81%	98.10%	86.81%
submission-2	87.72%	84.34%	87.55%	88.77%	83.48%	78.15%	98.06%	86.51%
submission-3	87.62%	84.73%	87.54%	89.33%	82.25%	77.94%	98.10%	86.40%
submission-4	82.76%	83.88%	84.09%	89.25%	78.81%	76.57%	97.02%	84.18%

5. CONCLUSION

We have outlined our submission to the DCASE 2022 Challenge Task 2, which features a domain shift between the training and test distributions.

In this challenge, we build six subsystems and four ensemble systems in which four new unsupervised models, namely WSP-NFCDEE, IGNN, LRCGNN, and AEGAN-AD are integrated. All these four new models are employed with unsupervised training. Our methods are expected to be promising because they clearly match the data sampling characteristics of practical application scenarios for machine working condition detection in Industry 4.0. Moreover, our best official score of ensembles can achieve 86.81% on the development dataset, which is 30.80% higher than the best baseline. Finally, our method ranks fourth on the evaluation dataset.

6. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219 and No. 62276153.

7. REFERENCES

[1] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 5, pp. 1–46, 2020.

[2] 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.

[3] K. Morita, T. Yano, and K. Tran, "Anomalous sound detection using cnn-based features by self supervised learning," DCASE2021 Challenge, Tech. Rep., July 2021.

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

[5] 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.

[6] 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.

[7] 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.

[8] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[9] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[10] 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.

[11] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[12] 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.

[13] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[15] 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.

[16] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[18] H. Schroter, A. N. Escalante-B, T. Rosenkranz, and A. Maier, "Deepfilternet: A low complexity speech enhancement framework for full-band audio based on deep filtering," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7407–7411.

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

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[21] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian conference on computer vision*. Springer, 2018, pp. 622–637.

[22] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[23] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.