OUTLIER EXPOSURE WITH EFFICIENT DIVISION OF POSITIVE AND NEGATIVE EXAMPLES FOR ANOMALOUS SOUND DETECTION

Yuuki Tachioka

Denso IT Laboratory, Tokyo, Japan

ABSTRACT

Unsupervised anomalous sound detection (ASD) is an important and practical task. For machine monitoring, there are target machine type and other types. Recently, the classification of positive and negative examples improved the performance of ASD, where the first is the normal data of the target type, and the latter is the anomalous data of the target type and normal data of other types. This assumes that the distance between normal and anomalous data of the target type is larger than that between anomalous data of the target type and normal data of other types; however, this assumption is not satisfied for some cases. The inclusion of normal data from other machine types in positive examples helps to improve the ASD performance of the target type, but its appropriate division of positive and negative examples is difficult because the number of required model training and testing is $M \cdot 2^{M-1}$ when the number of machine types is M. We propose an efficient division based on the performance change caused by the data mismatch between training and testing, which reduces the number of model training and testing to M and M^2, respectively. Experiments on task2 of the DCASE 2022 challenge show the effectiveness of our proposed approach.

Index Terms— anomalous sound detection, unsupervised training, outlier exposure, feature extraction

1. INTRODUCTION

Anomalous sound detection (ASD) is an important and practical task, and previous challenges in detection and classification of acoustic scenes and events (DCASE)[1] have proposed ASD tasks whose purpose is machine monitoring [1, 2, 3, 4]. For ASD, supervised training is effective [5, 6, 7], but because it is rare to obtain anomalous sound samples, unsupervised training that does not require anomalous sound samples is desirable [2]. There are two types of unsupervised ASD methods: inlier modeling (IM) [2, 5, 6, 7] and outlier exposure (OE) [8, 9, 10]. IM detects anomalies on the basis of anomaly scores by making a probability distribution of normal data. On the other hand, the OE-based model classifies positive examples and negative examples [8, 9, 10] based on the high performance of advanced classifiers, where positive examples are normal sounds and negative examples are anomalous sounds of the target type and normal sounds of other types, which can be discriminated from positive ones. Both methods have been widely used; in fact, recent DCASE challenges have prepared two types of corresponding baselines. For most cases, the OE-based model outperformed IM, but for some cases, classification catastrophically failed [9], that is, IM is more robust than the OE-based model.

To take advantage of both methods, two-stage ASD has been proposed to combine IM and OE [11, 12], which is one of the SOTA methods of the DCASE 2022 challenge. After features are extracted by an OE-based feature extractor, IM detects anomalous sounds. Here, we modify the OE-based feature extractor, which does not fully exploit the data of other machine types. This model assumes that the distance between normal and anomalous sounds of the target type, which is the distance between positive and negative examples, is larger than that between anomalous sound of the target type and normal sounds of other types, which is the distance among negative examples. As shown in the experiments later, this assumption is not satisfied for some cases. In addition, the performance of the multiconditioned models trained on sounds of multiple machine types simultaneously was better than that of models trained on sounds of a single machine type [13]. According to these observations, the inclusion of normal sounds of other machine types in positive examples helps to improve the ASD performance of the target type, but it is difficult to divide positive and negative examples appropriately because the number of total combinations is too large. Thus, we propose an efficient division of positive and negative examples based on the performance change caused by the mismatch of the data between training and testing. The proposed method can be easily applied to other types of OE-based ASD. Experiments on the DCASE 2022 challenge show that the proposed method improved ASD performance.

2. OE-BASED FEATURE EXTRACTOR

The two-stage ASD extracts features using OE-based models and inputs the obtained features into IM to make probabil-

[1] DCASE challenges are one of the biggest shared tasks for acoustic event detection.

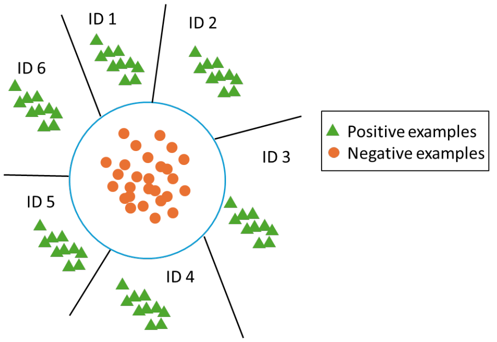

Fig. 1. A schematic diagram of assumptions of embedded vectors obtained by conventional OE-based feature extractor to classify section id.

ity distributions for normal data [11]. We briefly describe the OE-based feature extractor, which is the first stage of the two-stage ASD. To train discriminant feature extractor, classifiers are used to classify positive and negative examples, where positive examples are normal data of the target type and negative examples are anomalous data of the target machine type and normal data of other machine types. In an unsupervised setting, because anomalous data of the target machine type are unobserved, normal data of other machine types are used as negative examples when training classifiers. This is based on the assumption that embedded vectors of negative examples are located near the center of the hypersphere and those of positive data are remote from the center, as shown in Fig. 1 and that to properly classify section ids of normal data, clusters are made in the embedded space, where section id is the recording condition ID of a subset of normal data within the same machine type.

The classifier λ uses the features $f(\boldsymbol{x}_i)$ extracted from the feature extractor f, where the input acoustic feature is $\boldsymbol{x} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_I\}$ and i is the index of the sound files. For each i, the section id s and the machine type $m \in \mathcal{M}$ are specified. There are two types of loss functions to train classifiers λ. The one of the two loss functions is a categorical cross-entropy of section ids conditioned on the machine type. The classifier λ estimates the posterior probability $\hat{\pi}_\lambda$ of the section id $s \in \mathcal{S}$. For example, $\mathcal{S} = \{1, 2, ...\}$ and $\mathcal{M} = \{\text{bearing}, \text{fan}, ...\}$.

$$\mathcal{L}_s(\boldsymbol{x}; m) = \frac{-1}{|\mathcal{S}| \sum_i \pi_i(m)} \sum_{i,s} \pi_i(m)\pi_i(s) \log(\hat{\pi}_\lambda(s|f(\boldsymbol{x}_i))), \tag{1}$$

where $\pi_i(m)$ is a one-hot distribution of the correct label of machine type m, $\pi_i(s)$ is a one-hot distribution of the correct label of the section id s, and $\hat{\pi}_\lambda(s|f(\boldsymbol{x}_i))$ is the softmax output of the section id s. In addition, the other loss function is a cross-entropy of machine types as

$$\mathcal{L}_m(\boldsymbol{x}) = -\frac{1}{I} \sum_{i,m} \pi_i(m) \log(\hat{\pi}_\lambda(m|f(\boldsymbol{x}_i))), \tag{2}$$

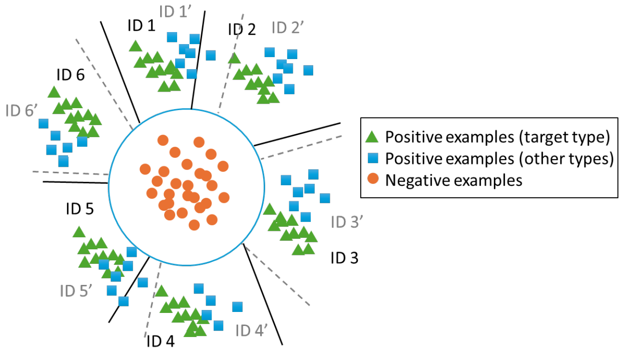

Fig. 2. A schematic diagram of assumption of embedded vectors obtained by the proposed OE-based model trained with two types of positive examples composed of normal data of target type and other types.

where $\hat{\pi}_\lambda(m|f(\boldsymbol{x}_i))$ is the softmax output of the machine type m. The loss function to be optimized for the model of m was a combination of them as

$$\mathcal{L}(m) = \mathcal{L}_m(\boldsymbol{x}) + \gamma\mathcal{L}_s(\boldsymbol{x}; m). \tag{3}$$

3. OE-BASED FEATURE EXTRACTOR TRAINED WITH EXTENDED POSITIVE SETS

3.1. Modified loss function with extended positive sets

As shown in the experiments in Sec. 4, data of some other machine types help improve the ASD performance of the target type. We modify the assumption of Fig. 1 as shown in Fig. 2. There are two types of positive examples in the diagram. One is from the normal data of the target type and the other is from those of other types, which is helpful for the target type classification. These two types of positive examples are far from the negative examples when these two types of positive examples are more similar than the others. Section ids can be different between two positive sets because section ids are arbitrary for each type. The conventional method [11] optimizes a loss function (3) only in terms of the target type m, but our proposed method optimizes a loss function

$$\mathcal{L}(m) = \sum_{m' \in \mathcal{M}_P} \left[\mathcal{L}_{m'}(\boldsymbol{x}) + \gamma\mathcal{L}_s(\boldsymbol{x}; m')\right], \tag{4}$$

for the set of positive machine types $\mathcal{M}_P$. Here, positive machine types $\mathcal{M}_P$ are the target type and helpful machine types, and when $\mathcal{M}_P$ only contains the target type, the proposed method is the same as the conventional method.

3.2. Efficient positive and negative set division

To divide the set of machine types $\mathcal{M}$ into the set of positive ones $\mathcal{M}_P$ and negative ones $\mathcal{M}_N$ ($\mathcal{M}_p \cap \mathcal{M}_N$ = empty and $\mathcal{M}_p \cup \mathcal{M}_N = \mathcal{M}$), for each machine type m, the other $m - 1$

types have an option to belong to $\mathcal{M}_P$ or $\mathcal{M}_N$. To find the best $\mathcal{M}_p$, the total number of training and testing of the model is $|\mathcal{M}| \cdot 2^{|\mathcal{M}|-1}$. In the case of $M = 7$ and $M = 10$, this number is 448 and 5120, respectively. In particular, model training is intractable. It is necessary to reduce this number; thus we propose an efficient division method.

To find the helpful type, the performance improvement or degradation caused by the mismatch of data between training and testing can be used. We compute a performance metric such as AUC [2] in the development set using a model trained on the m_{tr} data to evaluate the m_{te} data as $\varphi(m_{tr}, m_{te})$ where $m_{tr}, m_{te} \in \mathcal{M}$. This is made up of $|\mathcal{M}|$ matched cases and $|\mathcal{M}|(|\mathcal{M}| - 1)$ mismatched cases. Generally, in mismatched cases, the ASD performance degrades from the matched cases, but if the performance improves in the mismatched cases ($m_{tr} \neq m_{te}$), the data m_{tr} are suitable for training models of m_{te}. For each m_{te}, after the calculation of φ, which requires $|\mathcal{M}|$-times evaluations, model training is required once. The total number of necessary model training and that of testing are $|\mathcal{M}|$ and $|\mathcal{M}|^2$, respectively, which are much smaller than $|\mathcal{M}| \cdot 2^{|\mathcal{M}|-1}$.

The metric φ is normalized for each model because the performance of the matched case is different from machine type to machine type. Normalized performance metric is the difference of φ between matched and mismatched cases as

$$\bar{\varphi}(m_{tr}, m_{te}) = \varphi(m_{tr}, m_{te}) - \varphi(m_{tr}, m_{tr}). \tag{5}$$

For each m_{te}, if mismatched cases are better than the matched case, the data m_{tr} are helpful to improve the performance of m_{te}. This judgement is based on the normalized metric $\bar{\varphi}$ as

$$\bar{\varphi}(m_{tr}, m_{te}) \begin{cases} > 0 & (m_{tr} \in \mathcal{M}_P) \\ \leq 0 & (m_{tr} \in \mathcal{M}_N) \end{cases}. \tag{6}$$

4. EXPERIMENT

4.1. Experimental setups

We conducted an experiment using the DCASE challenge 2022 task2 dataset[3] [4] for unsupervised machine condition monitoring, which focuses mainly on the domain-shift scenario. It consists of two types of machines ("ToyCar" and "ToyTrain") from ToyADMOS2 [14] and of five types of machines ("bearing," "fan," "gearbox," "slider," and "valve") from MIMII DG [15]. Evaluation was performed on the development set and the evaluation set. There are three sections per machine type (sections 00, 01, and 02) for the development set and other three sections per machine type (sections 03, 04, and 05) for evaluation set. This task prepared two domains (source and target), but domain information cannot be used for evaluation. For training, each section provided 990 normal clips from the source domain and 10 normal clips from the target domain. Thus, the training condition for the target domain was fewshot. For testing, each section provided 100 normal clips and 100 anomalous clips from both domains.

Table 1. All combinations of harmonic mean of AUC and pAUC ($p = 0.1$)[%] for source domain (development set). m_{tr} is machine type for training data and m_{te} is machine type for test data.

$m_{tr} \setminus m_{te}$	bearing	fan	gearbox	slider	valve	ToyCar	ToyTrain
bearing	47.56	58.22	**66.06**	63.76	55.20	64.85	53.16
fan	57.76	58.89	58.63	66.01	**69.95**	63.06	52.01
gearbox	44.67	62.79	**69.01**	66.11	60.37	64.21	49.76
slider	46.64	62.87	66.82	**92.49**	61.82	60.56	53.12
valve	53.59	59.05	65.08	76.94	**88.83**	58.25	54.65
ToyCar	50.29	66.80	55.85	63.41	54.54	**70.43**	49.73
ToyTrain	54.51	54.82	62.01	64.60	54.25	**65.43**	56.96

Table 2. All combinations of harmonic mean of AUC and pAUC ($p = 0.1$)[%] for target domain (development set).

$m_{tr} \setminus m_{te}$	bearing	fan	gearbox	slider	valve	ToyCar	ToyTrain
bearing	57.12	56.18	**58.65**	57.95	49.65	53.59	53.56
fan	58.29	62.57	50.37	58.50	**63.29**	53.44	53.26
gearbox	57.44	56.36	**62.30**	58.64	55.00	53.58	50.70
slider	56.23	56.73	54.97	**61.82**	54.96	50.83	54.24
valve	54.12	54.37	50.39	60.77	**84.60**	49.15	55.31
ToyCar	59.33	**61.25**	50.16	59.20	49.02	54.25	46.00
ToyTrain	**56.93**	54.75	52.95	56.43	51.44	52.73	51.60

The Mel spectrogram was used for the input, where the window size was 128 ms and the hop size was 16 ms. The classification models were composed of efficientnetV2B0 [16], Transformer [17], and Conformer [18]. The number of epochs was 50, and the batch size was 128. AdamW optimizer was used with a learning rate of 10^{-3}. The weight γ in Eqs. (3) and (4) was 10. We modified the source codes provided by the authors of the two-stage ASD[4]. For the performance metric φ, we used the harmonic mean of AUC and pAUC ($p = 0.1$).

IM was used to calculate the anomaly score for the i-th file as $a_i = A(h(f(\boldsymbol{x}_i)))$, where IM h was Gaussian mixture models (16 mixtures) [19] and the aggregator A was a max pooling. Based on the score a_i, the performance of ASD was evaluated. To clarify the effectiveness of the proposed method, the system combination approach [12] was not used in this experiment.

[2] AUC score cannot be used when no anomalous data are provided. For that case, the classification accuracy of the section id of normal data can be used.

[3] We used DCASE challenge 2022 dataset because in DCASE challenge 2023, the settings were totally different and it is difficult to apply OE-based models.

Table 3. Harmonic mean of AUC and pAUC ($p = 0.1$)[%] for source domain (development set). ‘d(target)’ and ‘d(source)’ are proposed division on target and source domain of the development set, respectively.

Method	bearing	fan	gearbox	slider	valve	ToyCar	ToyTrain	hmean
baseline	47.56	58.89	69.01	92.49	88.83	**70.43**	56.96	65.84
d(target)	46.40	66.23	77.73	92.76	81.50	69.15	**66.22**	68.54
d(source)	**51.45**	62.94	**81.48**	**94.69**	88.00	69.95	63.20	**70.25**

Table 4. Harmonic mean of AUC and pAUC ($p = 0.1$)[%] for target domain (development set).

Method	bearing	fan	gearbox	slider	valve	ToyCar	ToyTrain	hmean
baseline	57.12	62.57	62.30	61.82	84.60	54.25	**51.60**	60.68
d(target)	60.02	64.19	65.44	**62.83**	82.53	53.88	46.51	60.54
d(source)	**60.64**	**66.64**	**66.41**	60.05	**87.51**	**56.63**	50.39	**62.42**

4.2. Result and discussion

Table 1 shows the harmonic mean of AUC and pAUC ($p = 0.1$) in the source domain of the development set. The row shows the machine type of the training data m_{tr} and the column shows the machine type of the test data m_{te}. For example, the element in the first row and in the second column ($\varphi(\mathrm{bearing}, \mathrm{fan}) = 58.22\%$) shows the harmonic mean of AUC and pAUC in the fan data evaluated with the model trained on the bearing data. The diagonal elements were the matched cases and the off-diagonal ones were the mismatched cases. If the assumption of Fig. 1 is satisfied, the diagonal elements should always be better than the off-diagonal ones, because data of other types were concentrated near the hypersphere and in mismatched cases it is hard to discriminate the difference between normal and anomalous data of the target type. This difference is negligible for the model trained on other types, because both are negative examples for the model. However, for three types out of seven machine types (bearing, fan, and ToyTrain), off-diagonal elements were better than diagonal ones. This shows that some machine types are helpful to improve the performance of the target type as shown in Fig. 2. Table 2 shows the harmonic mean in the target domain of the development set. The trends were similar to those of the cases in Table 1. For four of the seven types of machines (bearing, fan, ToyCar, and ToyTrain), the off-diagonal elements were better than the diagonal ones and the types of machines whose off-diagonal elements were better than the diagonal ones (bearing, fan, and ToyTrain) were common in Table 1 and Table 2. Based on these results, positive and negative examples were divided.

Table 3 shows the harmonic mean in the source domain of the development set for the matched cases, which compares the performance of the proposed methods using positive and negative division with the baseline. Except for ToyCar, the ASD performance was improved by the proposed method.

Table 5. Harmonic mean of AUC and pAUC ($p = 0.1$)[%] for target domain (evaluation set).

Method	bearing	fan	gearbox	slider	valve	ToyCar	ToyTrain	hmean
baseline	60.61	56.10	**57.81**	60.68	69.30	39.17	**52.33**	55.04
d(target)	**62.51**	58.55	55.41	**65.08**	76.74	40.27	48.82	56.07
d(source)	61.50	**60.80**	56.09	62.63	**79.70**	**45.22**	45.65	**56.88**

The division based on the source domain (Table 1) was better than that based on the target domain (Table 2). Table 4 shows the harmonic mean in the target domain of the development set. Except for ToyTrain, ASD performance improved and trends were similar to those of Table 3 despite the domain-shift scenario. The division based on the source domain was the best on average. Table 5 shows the harmonic mean in the target domain of the evaluation set. The proposed method improved the ASD performance of the baseline.

5. CONCLUSION

To improve ASD performance by including normal data from other machine types in positive examples appropriately, we propose an efficient division of positive and negative examples based on the performance change caused by the mismatch of data between training and testing. This method reduced the total number of training from $M \cdot 2^{M-1}$to M and that of testing from $M \cdot 2^{M-1}$to M^2, where M is the number of machine types. Experiments on task2 of the DCASE 2022 challenge show the effectiveness of our proposed method.

6. REFERENCES

[1] A Mesaros, T Heittola, A Diment, B Elizalde, A Shah, E Vincent, B Raj, and T Virtanen, “DCASE 2017 challenge setup: Tasks, datasets and baseline system,” in *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017, pp. 85–92.

[2] Y Koizumi, Y Kawaguchi, K Imoto, T Nakamura, Y Nishikaido, R Tanabe, H Purohit, K Suefusa, T Endo, M Yasuda, and N Harada, “Description and discussion on DCASE 2020 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring,” in *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 81–85.

[3] 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 sound detection for machine condition monitoring under domain shifted consitions,” in *Proceedings of Workshop on Detection and*

[4] `https://github.com/ibkuroyagi/dcase2022_task2_challenge_recipe`

Classification of Acoustic Scenes and Events (DCASE), 2021, pp. 186–190.

[4] Kota Dohi, Keisuke Imoto, Noboru Harada, Daisuke Niizumi, Yuma Koizumi, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, 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 *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2022.

[5] Erik Marchi, Fabio Vesperini, Florian Eyben, Stefano Squartini, and Björn Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 1996–2000.

[6] Dong Yul Oh and Il Dong Yun, "Residual error based anomaly detection using auto-encoder in SMD machine sound," *Sensors*, vol. 18, no. 5, 2018.

[7] Yuta Kawachi, Yuma Koizumi, and Noboru Harada, "Complementary set variational autoencoder for supervised anomaly detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2366–2370.

[8] Ritwik Giri, Srikanth V Tenneti, Fangzhou Cheng, Karim Helwani, Umut Isik, and Arvindh Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 46–50.

[9] Paul Primus, Verena Haunschmid, Patrick Praher, and Gerhard Widmer, "Anomalous sound detection as a simple binary classification problem with careful selection of proxy outlier examples," in *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 170–174.

[10] Tadanobu Inoue, Phongtharin Vinayavekhin, Shu Morikuni, Shiqiang Wang, Tuan Hoang Trong, David Wood, Michiaki Tatsubori, and Ryuki Tachibana, "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 66–70.

[11] Ibuki Kuroyanagi, Tomoki Hayashi, Kazuya Takeda, and Tomoki Toda, "Improvement of serial approach to anomalous sound detection by incorporating two binary cross-entropies for outlier exposure," in *Proceedings of 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 294–298.

[12] Ibuki Kuroyanagi, Tomoki Hayashi, Kazuya Takeda, and Tomoki Toda, "Two-stage anomalous sound detection systems using domain generalization and specialization techniques," Tech. Rep., DCASE2022 Challenge, 2022.

[13] Yuuki Tachioka, "Conditioning of autoencoder for various types of anomaly sound detection by using single model," The 11th International Conference on Computer and Communications Management (ICCCM), 2023.

[14] Noboru Harada, Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Masahiro Yasuda, and Shoichiro Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021, pp. 1–5.

[15] 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.

[16] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le, "Self-training with noisy student improves ImageNet classification," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10684–10695.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.

[18] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proceedings of Interspeech*, 2020, pp. 5036–5040.

[19] Wenbo Liu, Delong Cui, Zhiping Peng, and Jihai Zhong, "Outlier detection algorithm based on Gaussian mixture model," in *Proceedings of IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2019, pp. 488–492.