COMBINE MAHALANOBIS DISTANCE, INTERPOLATION AUTO ENCODER AND CLASSIFICATION APPROACH FOR ANOMALY DETECTION

Technical Report

Yuya Sakamoto and Naoya Miyamoto

Fixstars Corporation
3-1-1 Shibaura, Minato-ku, Tokyo, 108-0023 Japan
{yuya.sakamoto, naoya.miyamoto}@fixstars.com

ABSTRACT

This paper is a technical report of the method we submitted to DCASE 2021 Challenge Task 2. In our method, one sample is converted into a time-series log-mel-spectrogram similar to the Autoencoder-based baseline. For the feature vector obtained from this log-mel-spectrogram, 3 types of anomaly detection models, section ID classification, interpolation deep neural network and mahalanobis distance are constructed, and the final degree of anomaly is calculated as an ensemble of 3 models. In this task, it is necessary to deal with the domain shift problem, which has different characteristics between training data and test data. We addressed this problem by absorbing the difference in the mean of log-mel-spectrogram features between domains.

***Index Terms*—** Unsupervised anomaly detection, Interpolation deep neural network, Section ID classification

1. INTRODUCTION

Anomalous sound detection is a task to judge the normality / abnormality of the machine from the recorded sound of the machine. DCASE2021 Challenge Task 2 is a competition for the accuracy of anomalous sound detection [1] [2] [3]. The two main challenges in this task are:

1. Unsupervised detection to find anomalous data in test data sets in the situation where only normal data is given as training data
2. Domain shift. Acoustic characteristics such as machine part number and recording conditions differ between training data and test data.

1. is the same as the previous challenge [4]. 2. is a new challenge added this time, the dataset is divided into source domain and target domain, and there is enough training data of source domain, but there is very little target domain data. The task of domain shift is to detect abnormalities in the test data of the target domain with such training data. We have built the following three types of anomaly detection models for this task.

1. Section ID classification following the method of last year's top prizewinners [5] [6].
2. Interpolation deep neural network [7].
3. Anomaly detection using the mahalanobis distance of mean vector of log-mel-spectrogram.

The final anomaly score is calculated by ensemble the anomaly score obtained by these models.

In order to deal with domain shift, we assumed main shifts between domains is mean of log-mel-spectrogram features and there is no other shifts. Therefore, the problem of domain shift is dealt with by absorbing the mean value gap due to domain shift for each model.

2. METHOD

2.1. Common preprocessing

Before explaining the three types of models, the common preprocessing will be described. First, each sample given as a time series signal is converted into a log-mel-spectrogram in the same way as the Autoencoder-based baseline. Expressed mathematically, one sample is transformed into a set of vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$. Where, $\mathbf{x} \in \mathbb{R}^F$ is the log-mel-spectrgram in one time-frame, F is the number of mel-filters, and T is the number of time-frames. The number of mel-filters is $F = 128$, which is the same as the baseline.

2.2. Section ID classification

The Section ID classification model is the method adopted by many top prizewinners in the previous challenge [4], and this time we also adopted this idea. There are various networks used for inference, such as [5] and [6]. The basic idea common to all of them is to transform the anomaly detection problem into a classification problem that predicts section ID. This can be regarded as an approach to treat data from different sections as simulated anomaly data.

We concatenate each vector of log-mel-spectrogram by P frames as well as Autoencoder-based baseline PF dimensional vector $\mathbf{x}' \in \mathbb{R}^{PF}$ and built a neural network model that predicts the section ID for it. Specifically, the input layer is $P = 5, F = 128$, so it is 640 dimensions, the intermediate layer is 128-dimensional output Dense + Batch Normalization + Relu are 4 layers, and the output layer is Dense + softmax of 3 dimensions output to predict section ID. Since the section ID of the test data is known, the degree of anomaly is calculated based on whether or not the correct section ID could be predicted for each $\mathbf{x}'_i$. Specifically, it follows the anomaly calculation method of MobileNet V2-based baseline, and if $f(\cdot)$ is the softmax output by neural network for the correct section ID, the anomaly of each $\mathbf{x}'_i$ Is expressed as follows.

$$a_c(\mathbf{x}'_i) = \log \frac{f(\mathbf{x}'_i)}{1 - f(\mathbf{x}'_i)} \tag{1}$$

Also, since one sample contains $T-P+1$ vectors, the average of only part of (1) is used as the anomaly of Section ID classification. In the case of data with non-stationarity such as valve, it was effective to calculate the degree of anomaly by averaging only a part in this way.

All train and test sample are applied preprocessing to subtract mean of concatenated log-mel-spectrogram vector, $\mathbf{m}'$ ($= \frac{1}{T-P+1}\sum_{i=1}^{T-P+1}\mathbf{x}'_i$), from $\mathbf{x}'_i$. By applying this treatment to each sample, it can be expected that the difference in the average between the samples will disappear, and as a result, the difference between the domains will be absorbed. On the other hand, depending on the machine type and Section ID, the section cannot be predicted correctly because important information for classification is lost by subtracting the mean vector. Therefore, the training data is inferred by the trained network, and it is used in the final ensemble only when the score of the prediction result is below the threshold value. We used the threshold 0.0 for source domain and 0.5 for target domain.

2.3. Interpolation deep neural network

2.3.1. Anormaly Score

Interpolation deep neural network (IDNN) is proposed method in [7]. IDNN detect anomalies by using $1, \cdots, \frac{P+1}{2}-1, \frac{P+1}{2}+1, \cdots, P$ from P frames as inputs and predicting $\frac{P}{2}$ th frame. This time, $P = 5, F = 128$ as in 2.2. Therefore a model was constructed in which a 512-dimensional vector concatenated at the $1, 2, 4, 5$ frame was input, and a 128-dimensional vector at the 3 frame was output. The middle layer consists of 6 layers of 128-dimensional output Dense + Batch Normalization + Relu and finally connect to the output layer through Dense (no activation).

The output of this IDNN is used to calculate the anomaly, which follows the GMM-based scoring approach shown in [8]. However, in this method, a single Gaussian distribution is considered instead of the Gaussian Mixture Model. Assuming that IDNN$(\cdot)$ is a function that obtains the output of the above model, the error vector $\mathbf{e}_i$ for each frame can be obtained as follows.

$$\mathbf{e}_i = \mathrm{abs}(\mathbf{x}_{i+2} - \mathrm{IDNN}(\{\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+3}, \mathbf{x}_{i+4}\})) \tag{2}$$

Using this (2), calculate the error vectors of training data. Then, the average error vector $\boldsymbol{\mu}_e$ and its covariance $\boldsymbol{\Sigma}_e$ are obtained from the obtained error vectors set. For the anomaly of the test data sample, obtain $\mathbf{e}_i$ ($i = 1, \cdots, T-P+1$) by (2) and calculate the following.

$$d_i = \sqrt{(\mathbf{e}_i - \boldsymbol{\mu}_e)^T \boldsymbol{\Sigma}_e^{-1} (\mathbf{e}_i - \boldsymbol{\mu}_e)} \tag{3}$$

The final degree of anomaly is calculated by averaging this d_i, but like 2.2, only a part of d_i is extracted and averaged.

2.3.2. Bayesian mean estimation for domain shift

Before learning and inferring IDNN, the mean vector is subtracted from each concatinated input vector and output vector (i.e. $\mathbf{x}'$ described in 2.2) for normalizing to mean $\mathbf{0}$. However, because there is a gap in the mean vector between domains, the mean is calculated for each source domain and target domain. The mean vector $\boldsymbol{\mu}^{(s)}$ of the source domain is maximum likelihood estimated as from the vector of the source domain. On the other hand, the number of target domain training data is small. Therefore, it is estimated by utilizing the mean vector $\boldsymbol{\mu}^{(s)}$ and the covariance matrix $\boldsymbol{\Sigma}^{(s)}$ of the source domain. To achieve this, we used bayesian estimation [9]. First, it is assumed that each vector $\mathbf{x}'^{(t)}$ of the target domain is generated from the following equation using the target mean vector $\boldsymbol{\mu}^{(t)}$ and the covariance matrix $\boldsymbol{\Sigma}^{(s)}$ estimated from the data of the source domain.

$$\mathbf{x}'^{(t)} \sim p\left(\mathbf{x}'^{(t)}|\boldsymbol{\mu}^{(t)}\right) = \mathcal{N}\left(\mathbf{x}'^{(t)}|\boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(s)}\right) \tag{4}$$

In other words, the mean vector is peculiar to the target domain, but the covariance is considered to be common to the source domain. Then, suppose that the prior distribution of $\boldsymbol{\mu}^{(t)}$ follows the Gaussian distribution of mean $\boldsymbol{\mu}^{(s)}$ as follows.

$$\boldsymbol{\mu}^{(t)} \sim p\left(\boldsymbol{\mu}^{(t)}\right) = \mathcal{N}\left(\boldsymbol{\mu}^{(t)}|\boldsymbol{\mu}^{(s)}, \alpha\mathbf{I}\right) \tag{5}$$

α of (5) is estimated from the data, which will be described later. From (4) and (5), the posterior distribution given the vector set $\mathbf{X}'^{(t)}$ of the target domain $p(\boldsymbol{\mu}^{(t)}|\mathbf{X}'^{(t)})$. Specifically, it is given in the following form.

$$p\left(\boldsymbol{\mu}^{(t)}|\mathbf{X}'^{(t)}\right) = \mathcal{N}\left(\boldsymbol{\mu}^{(t)}|\hat{\mathbf{m}}^{(t)}, \hat{\boldsymbol{\Lambda}}^{-1}\right) \tag{6}$$

Here, the following formula is obtained.

$$\hat{\boldsymbol{\Lambda}} = N^{(t)}\left(\boldsymbol{\Sigma}^{(s)}\right)^{-1} + \frac{1}{\alpha}\mathbf{I} \tag{7}$$

$$\mathbf{m}^{(t)} = \hat{\boldsymbol{\Lambda}}^{-1}\left\{\left(\boldsymbol{\Sigma}^{(s)}\right)^{-1}\sum_{i=1}^{N^{(t)}}\mathbf{x}'^{(t)}_i + \frac{1}{\alpha}\boldsymbol{\mu}^{(s)}\right\} \tag{8}$$

$N^{(t)}$ is the total number of vectors obtained from the training data of the target domain. Normalize with $\mathbf{m}^{(t)}$ obtained by this (8) as the mean vector of the target domain. Finally, α can be regarded as a coefficient that indicates the validity of the prior distribution (5). Therefore, where $\bar{\mathbf{m}} = \frac{1}{N^{(t)}}\sum_{i=1}^{N^{(t)}}\mathbf{x}'^{(t)}_i$, It is estimated as follows.

$$\alpha = \left(\bar{\mathbf{m}} - \boldsymbol{\mu}^{(s)}\right)^T\left(\boldsymbol{\Sigma}^{(s)}\right)^{-1}\left(\bar{\mathbf{m}} - \boldsymbol{\mu}^{(s)}\right) \tag{9}$$

Note that this model is built separately for each machine type and each section.

2.4. Mahalanobis distance of mean vector

In the previous challenge, we calculated the mean vector of log-mel-spectrogram for each sample as following, and found that it is effective to use the degree of anomaly according to the mahalanobis distance [10].

$$\mathbf{m} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{x}_i \tag{10}$$

Note that, $\mathbf{x}_i$ is not a vector in which frames are concatenated. Thus, the dimension of $\mathbf{x}_i$ is 128.

On the other hand, this challenge is different from the previous one in the following two points.

1. Domain shift.
2. Attribute information such as operating speed level and machine part number is given only to each sample of training data.

As mentioned in 2., training data is given attribute information. Therefore, training data can be grouped so that samples with the same attribute information are in the same group. Thus, we adopted a method to calculate the degree of abnormality based on the mahalanobis distance for each group. However, since attribute information is not given to test data, the mahalanobis distance is calculated for all groups to which it can belong, and the smallest one is calculated as the degree of anomaly. The mahalanobis distance given a test sample is calculated as follows:

$$a_m(\mathbf{m}_*) = \min_{g \in G} \sqrt{(\mathbf{m}_* - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}^{-1} (\mathbf{m}_* - \boldsymbol{\mu}_g)} \tag{11}$$

Where $\mathbf{m}_*$ is the feature vector of the test sample calculated by (10), and G is the set of groups to which the test data can belong, Let $\boldsymbol{\mu}_g$ be the mean vector for each group and $\boldsymbol{\Sigma}$ be the covariance matrix.

Also, due to the problem of domain shift, the training data given as the target domain is very small, so the estimation of the covariance matrix of the data belonging to the target domain does not work well. Therefore, the covariance matrix is common to all groups, and only the mean vector is estimated for each group. The mean vector in the group of target domain is estimated using the bayesian estimation described in 2.3.2. In this case, the mean vector of the prior distribution is the average of all data in the source domain.

Note that this model is built separately for each machine type and each section same as IDNN model described in 2.3. In addition, this method of averaging the log-mel-spectrogram is disadvantageous because it erases information in non-stationary data such as valve, so this method is not used in the valve dataset.

2.5. Ensemble

Normalize each anomaly obtained by the method shown in 2.2, 2.3, 2.4 to mean 0 and standard deviation 1. Average them to calculate the final anomaly. However, some models are not used depending on the machine type and domain. These variations vary by submission (see 3 for details). Finally, we set decision threshold is 0.0.

3. SUBMITTED SYSTEM VARIATION

In this task, submission of up to 4 systems is allowed, so we submitted 4 types of results by changing the combination of models used and hyperparameters. In submit1 and submit3, the models used depends on the machine type. Details are shown in Table 1. The difference between submit1 and submit3 is whether the IDNN scores are calculated by averaging part of d_i or all of them.

In submit2, all models are used for all machine type, except for the mahalanobis distance for valve.

In submit4, the Section ID classification model is not used, and other is the same as submit1.

4. RESULT

Training data for each machine type, 3 sections and validation data with correct answers are given for development. We used this data to evaluate AUC and pAUC. The evaluation results are shown in Table 2 and 3. These tables show the arithmetic mean of AUC (pAUC) calculated for each section and each domain. The baseline results are excerpted from [11].

5. CONCLUSION

In this article, we have shown the three models we built for DCASE 2021 Challenge Task 2 and their ensembles. In addition, the problem of domain shift was dealt with by absorbing the mean value gap between source and target. With these measures, we were able to obtain an arithmetic mean AUC of about 80 % in validation data, which is higher than that of the two baseline systems. The method described in this article is still weak in dealing with domain shifts. This point are expected improvement, for example, by combining the bayesian estimation as used in this paper with the Adaptive Batch Normalization proposed by [12].

6. REFERENCES

[1] 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 conditions," *In arXiv e-prints: 2106.04492, 1 – 5*, 2021.

[2] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, "MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions," *In arXiv e-prints: 2006.05822, 1 – 4*, 2021.

[3] 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," *arXiv preprint arXiv:2106.02369*, 2021.

[4] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, November 2020, pp. 81–85. [Online]. Available: http://dcase.community/documents/workshop2020/proceedings/DCASE2020Workshop_Koizumi_3.pdf

[5] R. Giri, S. V. Tenneti, K. Helwani, F. Cheng, U. Isik, and A. Krishnaswamy, "Unsupervised anomalous sound detection using self-supervised classification and group masked autoencoder for density estimation," DCASE2020 Challenge, Tech. Rep., July 2020.

[6] P. Primus, "Reframing unsupervised machine condition monitoring as a supervised classification task with outlier-exposed classifiers," DCASE2020 Challenge, Tech. Rep., July 2020.

[7] 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)*, 2020, pp. 271–275.

[8] T. Hayashi, T. Yoshimura, and Y. Adachi, "Conformer-based id-aware autoencoder for unsupervised anomalous sound detection," DCASE2020 Challenge, Tech. Rep., July 2020.

Table 1: use model list of submit 1 and 3

Algorithm	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
Classification	target only	source only	yes	yes	yes	yes	yes
IDNN	no	yes	no	yes	no	yes	yes
Mahalanobis	yes	yes	yes	yes	yes	yes	no

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[10] Y. Sakamoto and N. Miyamoto, "Anomaly calculation for each components of sound data and its integration for dcase 2020 challenge task2," DCASE2020 Challenge, Tech. Rep., July 2020.

[11] http://dcase.community/challenge2021/task-unsupervised-detection-of-anomalous-sounds/.

[12] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=Hk6dkJQFx

Table 2: Arithmetic mean of AUC

submits	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
baseline (AE)	63.19	63.00	64.03	66.76	63.66	69.16	53.74
baseline (MobileNetV2)	59.58	59.16	64.66	68.24	64.20	62.62	57.07
submit1	84.77	83.45	72.12	79.26	73.22	76.61	86.17
submit2	80.74	80.33	71.05	79.26	73.26	76.61	86.17
submit3	84.77	81.71	72.12	78.76	73.22	75.50	77.55
submit4	83.11	82.46	70.14	78.05	71.43	74.61	78.00

Table 3: Arithmetic mean of pAUC

submits	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
baseline (AE)	52.42	54.90	53.58	52.80	54.74	56.40	50.61
baseline (MobileNetV2)	57.64	51.74	64.84	60.03	58.06	56.86	52.83
submit1	65.90	70.10	59.14	57.90	61.07	63.21	71.77
submit2	62.71	68.49	57.05	57.90	60.28	63.21	71.77
submit3	65.90	65.41	59.14	56.93	61.07	61.42	57.41
submit4	63.02	68.64	55.14	60.05	59.08	60.89	61.78