DISTANCE-BASED UNSUPERVISED ANOMALOUS SOUND DETECTION WITH ATTENTIVE STATISTICS POOLING AND ARCFACE MULTI-TASK LEARNING

Technical Report

Masayuki Sera[1*], *Takao Kawamura*[1], *Nobutaka Ono*[1],

[1] Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo, Japan,
{sera-masayuki, kawamura-takao}@ed.tmu.ac.jp, onono@tmu.ac.jp

ABSTRACT

In this technical report, we describe our submission to the DCASE 2025 Challenge Task 2, titled "First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring." Our system is a distance-based anomalous sound detection method that determines whether a test input is normal or anomalous based on the Euclidean distance to embeddings of normal data. To obtain effective embeddings, we first apply the pretrained acoustic model BEATs to the input audio clip without any fine-tuning. The resulting patch-level features are then aggregated using Attentive Statistics Pooling to form a fixed-dimensional representation. To further improve the embeddings, we employ AcrFace-based multi-task learning with machine type and attribute classification objectives, which are used only during training. Our system achieved an Ω score of 0.6132 on the official development dataset, corresponding to a 5.3 percentage point improvement over the baseline system (0.5599).

Index Terms— Anomalous Sound Detection, BEATs, ArcFace, Multi-task Learning, Attentive Statistics Pooling

1. INTRODUCTION

As automation and AI-based predictive maintenance become increasingly important in manufacturing, Anomalous Sound Detection (ASD) is gaining attention as a powerful means of early fault detection in machinery. However, practical deployment faces three major challenges:

- Since anomalous sounds are rare and diverse, unsupervised learning using only normal sounds is a realistic approach.
- Domain shifts occur due to differences in machine types and installation environments [1].
- In "first-shot" conditions, only a very limited number of normal sounds are available from unknown machines.

DCASE 2025 Task 2 addresses these issues by requiring anomaly detection for *unknown machines* using only a *single domain* and *a small number of normal samples* [2]. A leading approach in the previous DCASE 2024 Task 2 [3] was the AITHU system [4], which achieved high performance by fine-tuning the pretrained acoustic model BEATs [5] with LoRA [6]. Inspired by this approach, we also utilize BEATs for patch-level acoustic feature extraction. However, to reduce training cost and improve scalability, we adopt a simpler strategy by freezing the BEATs parameters and avoiding any fine-tuning.

*This work was supported by JST SICORP Grant Number JPMJSC2306.

The main ideas behind our method are summarized as follows:

1. We apply Attentive Statistics Pooling [7, 8, 9], an attention-based mechanism that assigns patch-wise weights to extract statistical features emphasizing localized anomalies.
2. We enhance the embedding representation through multi-task learning with ArcFace [10], using machine type and attribute classification as auxiliary tasks [11].
3. During inference, we employ a retraining-free, distance-based detection method that computes anomaly scores without relying on any classifier.
4. Since the BEATs parameters are frozen, we can pre-extract and store patch-level features for all audio clips in advance, which accelerates training and would enable scalability to large datasets.

2. PROPOSED SYSTEM

The system consists of three modules and operates as a single pipeline: "Feature Extraction, Discriminative Space Learning, Distance-Based Inference."

1. Embedding Representation Extraction
2. Embedding Enhancement via Multi-Task Training
3. Distance-Based Anomaly Scoring

2.1. Embedding Representation Extraction

2.1.1. Acoustic Feature Extraction

Given an input waveform $a \in \mathbb{R}^T$ (10 seconds, 16 kHz, $T = 160{,}000$), we apply BEATs [5] to extract a sequence of patch-level features. Specifically, we use the pretrained weights provided as "`BEATs_iter3.pt`," which correspond to the BEATs Base model trained on AudioSet [12]. All parameters of BEATs are kept frozen during training to reduce computational cost.

BEATs first converts the input waveform into a log-Mel spectrogram with 128 frequency bins and approximately 1000 frames (using a 25 ms window and a 10 ms hop size). This spectrogram is divided into 2D patches of size 16 (frequency axis) × 16 (time axis), resulting in a total of $128/16 \times 1000/16 = 8 \times 62 = 496$ patches. Each patch is projected to a D-dimensional embedding vector through a linear transformation, yielding a sequence of patch-level feature vectors.

$$X = f_{\mathrm{BEATs}}(a) = [x_1, x_2, \ldots, x_P], \tag{1}$$

where $x_p \in \mathbb{R}^D$, $P = 496$, and $D = 768$.

2.1.2. Attentive Statistics Pooling

For each patch x_p, we compute an attention weight as:

$$\alpha_p = \frac{\exp(v_a^\top \tanh(W_a x_p))}{\sum_{j=1}^{P} \exp(v_a^\top \tanh(W_a x_j))}, \tag{2}$$

where $\sum_{p=1}^{P} \alpha_p = 1$. In this formulation, $W_a \in \mathbb{R}^{d_a \times D}$ is a learnable projection matrix that projects the input patch feature x_p into a d_a-dimensional intermediate space, and $v_a \in \mathbb{R}^{d_a}$ is a learnable vector used to compute a scalar attention score from the projected representation. We set $d_a = 128$ in our submission.

Using these attention weights, we calculate the weighted mean vector $\mu = [\mu_1, \cdots, \mu_D]^\top \in \mathbb{R}^D$ and the weighted standard deviation vector $\sigma = [\sigma_1, \cdots, \sigma_D]^\top \in \mathbb{R}^D$ as follows:

$$\mu_d = \sum_{p=1}^{P} \alpha_p x_{p,d} \tag{3}$$

$$\sigma_d = \sqrt{\sum_{p=1}^{P} \alpha_p \left(x_{p,d} - \mu_d\right)^2}, \tag{4}$$

where $x_{p,d}$ denotes the d-th element of the p-th patch feature. These statistics are computed independently for each dimension $d = 1, \ldots, D$. We then concatenate these two vectors along the feature dimension to obtain the final embedding:

$$z_{\text{pool}} = [\mu; \sigma], \tag{5}$$

where $z_{\text{pool}} \in \mathbb{R}^{2D}$.

2.2. Embedding Enhancement via Multi-Task Training

To improve the quality of the embedding z_{pool} obtained from Attentive Statistics Pooling, we apply multi-task learning using ArcFace during training.

First, z_{pool} is projected into a lower-dimensional space as follows:

$$e = \mathrm{ReLU}(\mathrm{BN}(W_e z_{\text{pool}})), \tag{6}$$

where $W_e \in \mathbb{R}^{E \times 2D}$ is a learnable weight matrix, and BN and ReLU denote batch normalization and the rectified linear unit activation function, respectively. We set $E = 512$ in our implementation. This projected embedding e is used only during training.

We then apply ArcFace-based classification to jointly learn machine type and attribute labels. For each classification task, we compute the logit corresponding to label j as follows:

$$\text{logit}_j = \begin{cases} s\cos(\theta_j + m) & \text{if } j = y, \\ s\cos\theta_j & \text{otherwise}, \end{cases} \tag{7}$$

$$\theta_j = \arccos\left(\frac{e^\top c_j}{\|e\|\|c_j\|}\right), \tag{8}$$

where y denotes the index of the ground-truth class, and j refers to a candidate class index ranging over all classes. Here, $c_j \in \mathbb{R}^E$ is a learnable weight vector corresponding to class j. We set $s = 30$ and $m = 0.5$ as scale and margin parameters, respectively. These logits are passed through a softmax function to produce class probabilities. Then, the cross-entropy loss is computed using the ground-truth label y. The resulting losses are denoted as L_{main} for machine type classification and $\ell_{i,n}$ for the n-th sample in attribute classification task i. The total loss is:

$$L = \beta L_{\text{main}} + \sum_{i=1}^{K} \sum_{n=1}^{N_i} \gamma_{i,n}\, \ell_{i,n}, \tag{9}$$

where β is a balancing weight between the main and auxiliary tasks, and $\gamma_{i,n}$ denotes the weighting factor for the n-th sample in task i. In our submission, we set $\beta = 1.0$ and defined $\gamma_{i,n}$ based on the number of attribute sub-tasks K associated with each machine. Specifically, the weight for each sub-task is set to $0.2/K$. For example, if a machine has one sub-task ($K = 1$), then $\gamma_{i,n} = 0.2$; if it has two sub-tasks ($K = 2$), then each is assigned $\gamma_{i,n} = 0.1$.

After training, only z_{pool} is retained and used for inference, while the projection and classification modules are discarded.

2.3. Distance-Based Anomaly Scoring

At inference time, anomalies are detected based on the Euclidean distance between the embedding of a test input and those of normal data. The process consists of two steps:

1. **Normalization:** Each dimension d of the test embedding $z_{\text{pool}}^{\text{test}}$ is normalized using the mean $\mu_{m,d}$ and standard deviation $\sigma_{m,d}$ computed from the embeddings of normal data for machine type m:

$$\hat{z}_{\text{pool},d}^{\text{test}} = \frac{z_{\text{pool},d}^{\text{test}} - \mu_{m,d}}{\sigma_{m,d}}, \tag{10}$$

   for all $d = 1, \ldots, 2D$. Here, $\mu_{m,d}$ and $\sigma_{m,d}$ denote the mean and standard deviation across the training embeddings of machine type m, independently computed for each dimension. Note that these statistics are unrelated to the weighted mean and standard deviation computed in Attentive Statistics Pooling (eqs. (3) and (4)).

2. **Anomaly scoring:** The anomaly score is computed as the minimum Euclidean distance from the normalized test embedding $\hat{z}_{\text{pool}}^{\text{test}}$ to the set of normalized embeddings $\hat{E}_{\text{normal}}^{m}$:

$$\text{anomaly\_score}(\hat{z}_{\text{pool}}^{\text{test}}) = \min_{\hat{z} \in \hat{E}_{\text{normal}}^{m}} \left\| \hat{z}_{\text{pool}}^{\text{test}} - \hat{z} \right\|, \tag{11}$$

   where $\hat{E}_{\text{normal}}^{m}$ denotes the set of normalized embeddings obtained from normal training inputs of machine type m. Based on the computed anomaly score, a test input is classified as anomalous if the score exceeds a predefined threshold. In our submission, this threshold is determined automatically for each machine type by analyzing the distribution of anomaly scores derived from normal training samples. Specifically, we fit a Gamma distribution to the scores of these normal samples and define the threshold as the 90th percentile of the fitted distribution. This data-driven approach enables adaptive thresholding without requiring any labeled anomalous data.

This scoring method allows for anomaly detection without additional retraining or complex post-processing.

Table 1: Overview of Compared Methods

Method	BEATs	Pooling	Training Task	Inference
Baseline (FS-AE)	–	–	–	Reconstruction Error
BEATs + Average Pooling	Fixed	Simple Average	Machine Type	Distance
BEATs + Attentive Statistics Pooling	Fixed	Weighted Stats	Machine Type	Distance
Proposed	Fixed	Weighted Stats	Machine Type + Attributes	Distance

Table 2: Detailed Scores by Machine Type (AUC src / AUC tgt / pAUC)

Method	Metric	bearing	fan	gearbox	slider	ToyCar	ToyTrain	valve
Baseline	AUC(src)	**0.748**	**0.656**	0.712	0.759	**0.833**	0.500	0.554
	AUC(tgt)	0.562	0.358	0.603	0.526	0.654	0.524	0.566
	pAUC	0.547	0.514	0.515	**0.536**	**0.666**	0.497	0.519
Average Pooling	AUC(src)	0.593	0.526	0.708	0.824	0.685	**0.753**	0.728
	AUC(tgt)	0.589	0.500	0.650	0.554	0.642	0.744	0.627
	pAUC	0.566	0.511	0.585	0.497	0.516	0.582	0.576
Attentive Statistics Pooling	AUC(src)	0.626	0.571	0.691	0.805	0.614	0.749	**0.765**
	AUC(tgt)	0.722	0.540	0.658	**0.560**	0.683	**0.762**	0.592
	pAUC	**0.610**	0.507	0.597	0.501	0.535	**0.610**	**0.586**
Proposed	AUC(src)	0.628	0.568	**0.744**	**0.806**	0.702	0.719	0.682
	AUC(tgt)	0.728	**0.553**	**0.686**	0.541	**0.723**	0.743	**0.707**
	pAUC	**0.614**	**0.518**	**0.624**	0.494	0.546	0.574	0.566

Table 3: Overall Ω Score (Harmonic mean of AUC and pAUC)

Method	Ω Score
Baseline (FS-AE)	0.5599
BEATs + Average Pooling	0.5912
BEATs + Attentive Statistics Pooling	0.6076
Proposed	**0.6132**

3. EXPERIMENTS

3.1. Setup

We conducted our experiments using the DCASE 2025 Task 2 development dataset [13, 14], which includes seven machine types: *bearing*, *fan*, *gearbox*, *slider*, *ToyCar*, *ToyTrain*, and *valve*. During training, only normal sound data were used in accordance with the task definition, which assumes no access to anomalous samples. For evaluation, we computed anomaly scores based on the Euclidean distance between embeddings, and applied this scoring method to both source and target domain samples. As the evaluation metric, we calculate AUC and pAUC (with FPR ≤ 0.1) for each machine type and domain (source/target), and use their harmonic mean:

$$\Omega = \mathrm{Hmean}(\mathrm{AUC},\, \mathrm{pAUC}), \tag{12}$$

as the final performance score. Here, $\mathrm{Hmean}(b_1, b_2) = \dfrac{2b_1b_2}{b_1 + b_2}$ denotes the harmonic mean of two values b_1 and b_2.

3.2. Compared Methods

Table 1 summarizes the four methods compared in this study. Each method differs in its pooling strategy and task configuration. Note that the baseline method (FS-AE) [15] does not use BEATs, while the other methods use BEATs with frozen weights.

To evaluate the effectiveness of the proposed method, we compare it with a baseline and two ablation variants, that differ in their pooling strategy and training objectives. All methods use fixed BEATs weights; the differences

3.3. Results and Discussion

As shown in Table 3, all methods utilizing BEATs embeddings outperform the baseline FS-AE system in terms of the overall Ω score, indicating the benefit of using pretrained representations. Among them, the proposed method achieves the highest score of 0.6132, followed by Attentive Statistics Pooling (0.6076) and Average Pooling (0.5912). This trend suggests that both attentive pooling and multi-task learning are effective components contributing to performance improvements.

Table 2 provides more detailed scores by machine type and domain. The proposed method achieves the best or comparable performance in most cases. Notably, it records the highest pAUC for *bearing*, *fan*, and *gearbox*, and the highest AUC in the target domain for *fan*, *gearbox*, *ToyCar*, and *valve*. Meanwhile, for *slider* and *ToyTrain*, the Attentive Statistics Pooling method slightly outperforms the proposed method in AUC for the source domain, indicating that attention-based pooling alone is particularly effective in these cases.

Overall, these results demonstrate that the combination of BEATs embeddings, Attentive Statistics Pooling, and multi-task classification yields robust and consistent performance across various machine types and domain conditions. The proposed method provides a strong and generalizable approach to anomaly detection in the DCASE 2025 Task 2 setting.

4. CONCLUSION

In this paper, we proposed a unsupervised anomalous sound detection system that combines Attentive Statistics Pooling and ArcFace-based Multi-Task Learning, while keeping the BEATs model frozen. The system employs distance-based inference without requiring any classifier during inference.

Our method achieved an Ω score of 0.6132 on the official dataset, representing a 5.3 percentage point improvement over the baseline, while significantly reducing training cost by avoiding fine-tuning of the acoustic model.

In future work, we plan to extend the system to machine types with limited attribute information by leveraging metadata and exploring self-supervised adaptation techniques, aiming for practical deployment in real-world industrial environments.

5. REFERENCES

[1] K. Wilkinghoff, T. Fujimura, K. Imoto, J. L. Roux, Z.-H. Tan, and T. Toda, "Handling domain shifts for anomalous sound detection: A review of DCASE-related work," *arXiv e-prints: 2503.10435*, 2025.

[2] T. Nishida, N. Harada, D. Niizumi, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, K. Imoto, K. Dohi, H. Purohit, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2025 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," *arXiv e-prints: 2506.10097*, 2025.

[3] ——, "Description and discussion on DCASE 2024 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2024, pp. 111–115.

[4] Z. Lv, A. Jiang, B. Han, Y. Liang, Y. Qian, X. Chen, J. Liu, and P. Fan, "AITHU system for first-shot unsupervised anomalous sound detection," Detection and Classification of Acoustic Scenes and Events Challenge (DCASE), Tech. Rep., 2024.

[5] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. International Conference on Machine Learning (ICML)*, vol. 202, 2023, pp. 5178–5193.

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. International Conference on Learning Representations (ICLR)*, 2022.

[7] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. INTERSPEECH*, 2018, pp. 2252–2256.

[8] H. J. Kim, C. Lim, J. Lee, H. K. Bae, M. J. Kim, and Y. S. Lee, "Colligate embeddings from pretrained models based on different preprocessing methods," Detection and Classification of Acoustic Scenes and Events Challenge (DCASE), Tech. Rep., 2024.

[9] F. Chu, Y. Zhou, and M. Qian, "Unified anomaly detection for machine condition monitoring: Handling attribute-rich and attribute-free scenarios," Detection and Classification of Acoustic Scenes and Events Challenge (DCASE), Tech. Rep., 2024.

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4685–4694.

[11] S. Venkatesh, G. Wichern, A. Subramanian, and J. Le Roux, "Improved domain generalization via disentangled multi-task learning in unsupervised anomalous sound detection," in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2022.

[12] J. F. Gemmeke, D. P. W. 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 *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[13] 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 *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2021, pp. 1–5.

[14] 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 *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2022.

[15] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "First-shot anomaly detection for machine condition monitoring: A domain generalization baseline," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2023, pp. 191–195.