

AISTAT LAB SYSTEM FOR DCASE 2025 TASK 2: FIRST-SHOT UNSUPERVISED ANOMALOUS SOUND DETECTION FOR MACHINE CONDITION MONITORING

Technical Report

Hyun Jun Kim¹, Min Jun Kim², Hyeon Gyu Bae², Changwon Lim^{1,2,*}

Department of Smart Cities, Chung-Ang University, Seoul, Korea
Department of Applied Statistics, Chung-Ang University, Seoul, Korea
{hyunjun0615, goodwill1669, amysst11, [clim](mailto:clim@cau.ac.kr)}@cau.ac.kr

ABSTRACT

This report addresses the AISTAT team’s submission to First-Shot Unsupervised Anomalous Sound Detection task in DCASE2025 Task 2. Unlike the previous years’ challenges, the available training data spans from the training dataset of 2020 to 2025. To effectively learn from the given data, we adopt a two-stage training strategy consisting of pretraining followed by transfer learning. During the transfer learning stage, pseudo-labeling is applied to data without attribute information to assign approximate labels and enhance model adaptation. Also, ArcFace loss and Center loss are employed together to directly reduce class-intra variance. Additionally, to extract more informative audio representations, we leverage the multi-layer aggregation. Through these techniques, our single best model achieved a harmonic mean of 66.12, while our best ensemble model achieved a harmonic mean of 66.78.

Index Terms— Anomalous sound detection, two-stage training, pseudo-labeling, class-intra variance, and multi-layer aggregation

1. INTRODUCTION

First-Shot Unsupervised Anomalous Sound Detection is a task in which only normal sound data is available for training, and the key objective is to effectively learn the feature distribution of the normal data. A notable characteristic of DCASE2025 Task 2 [1-4] is that training data from the years 2020 to 2025 can be used, while the evaluation is conducted exclusively on the test data from 2025. Also, similar to DCASE2024 Task 2 [5], a portion of the training data does not contain attribute information. Additionally, clean noise data and clean machine sound data are provided as supplementary data, randomly assigned per machine. In this study, we trained a feature extractor by formulating the problem as a classification task, where the machine attributes were treated as class labels [6,7]. Once the feature extractor was trained, we used a K-Nearest Neighbor (KNN) detector during the test phase to compute anomaly scores based on the distance between the test samples and the training set composed of normal data, thereby identifying abnormal instances.

Although the available training data spans from 2020 to 2025, testing is restricted to 2025 data. Accordingly, we divided the

training procedure into two stages. In the first stage, a pretraining phase was conducted on the 2020-2025 data to learn general representations of machine sounds. In the second stage, transfer learning was performed to adapt the model specifically to the 2025 data. For data without attribute information, we applied pseudo-labeling to supplement the missing labels prior to transfer learning.

Since anomaly scores are calculated based on the distance from the normal training samples during the test phase, abnormal data must be distant from all classes of normal features. Therefore, the feature extractor must be trained to minimize intra-class variance while maximizing inter-class separability. To achieve this, we employed ArcFace loss during pretraining to encourage compact intra-class representations and greater inter-class distances and further applied Center Loss during transfer learning to directly reduce intra-class variance.

In Anomalous Sound Detection, it is crucial to extract rich and informative representations from audio to effectively distinguish between normal and abnormal sounds. To this end, we employed a multi-layer aggregation method that integrates patch embeddings across multiple transformer layers. Our experiments demonstrate that this approach yields superior performance compared to using only the last-layer patch embeddings.

The remainder of this paper is organized as follows. Section 2 describes the proposed methods in detail. Section 3 presents the experimental setup, results and evaluation of our system. Finally, Section 4 summarizes our work.

2. METHOD

This section outlines the methodologies developed for the challenge. Our approach leverages a two-stage framework to learn robust and discriminative audio representations, addressing the challenges of diverse machine types, varying operating conditions, and domain shifts. The overall architecture of our system is illustrated in Figure 1, with subsequent subsections detailing the pretraining and transfer learning phases, feature aggregation strategies, loss functions, pseudo-labeling, and testing procedures.

2.1. Two-stage framework

In the **pretraining phase**, we utilized the entire dataset spanning 2020 to 2025 to learn general features of machine sounds. This dataset includes a diverse set of machines, such as ToyCar, Fan, Gearbox, and others, some of which are not present in the 2025

* Corresponding author

data, thereby facilitating the learning of broad acoustic characteristics across various machine types and operating conditions. To

ensure data integrity, we removed any duplicate data across different years. Each unique combination of machine type and

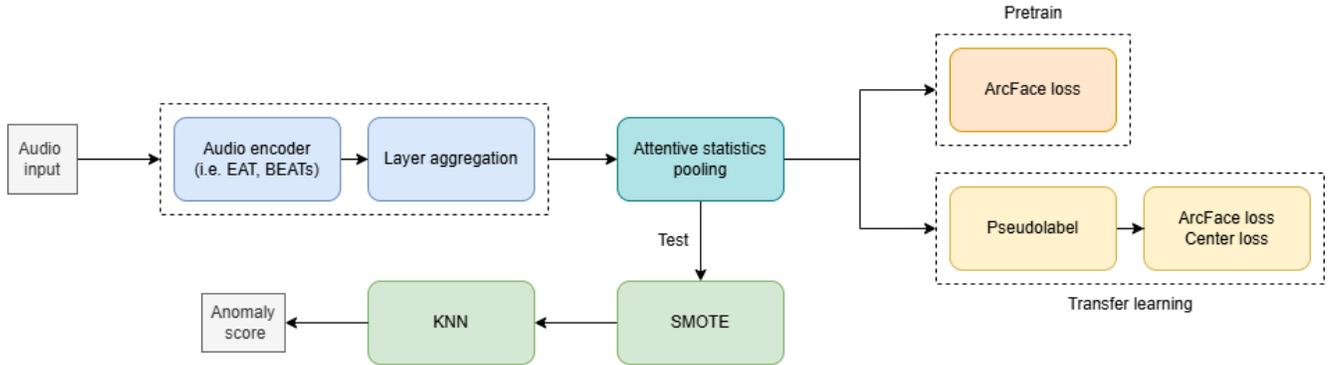


Figure 1: Illustration of the proposed system. Audio features are derived through layer aggregation across the outputs of all transformer layers in the audio encoder. These outputs are processed by attentive statistics pooling to produce the final embedding. During training, this embedding is optimized using classification-based loss functions to distinguish machine types and attribute information. In the testing phase, the final anomaly score is computed via K-Nearest Neighbors (KNN) between the embeddings of training and test data. Note that the pretraining stage employs only ArcFace loss, while the transfer learning stage additionally incorporates Center loss.

attribute information (e.g., operating speed, environmental noise) was treated as a distinct class, aligning with our classification-based feature learning approach. This strategy encourages the audio encoder to extract discriminative features that capture the unique acoustic signatures of each machine attribute pair.

However, including target domain in the pretraining phase would introduce severe class imbalance, as these datasets may contain varying distributions or fewer samples for certain machines or conditions. Such an imbalance is detrimental to our classification-based approach, as it could bias the model toward overrepresented classes, reducing their ability to generalize effectively. To mitigate this, we excluded all the target domain data from the pre-training phase, focusing solely on source domain data from 2020 to 2025. This ensured a balanced class distribution, allowing the model to learn robust and generalizable features without being skewed by the specific characteristics of target domains.

The pre-training was conducted using the ArcFace loss function, which enhances class separability by introducing an angular margin in the feature space, as detailed in Section 2.4. The resulting pre-trained model serves as a strong foundation for capturing general machine sound characteristics, which are critical for the subsequent transfer learning phase.

In the **transfer learning phase**, we finetuned the pre-trained model using the entire 2025 training dataset, which includes both source and target domain data. Consistent with the pre-training phase, each combination of machine type and attribute information was treated as a single class for classification purposes. This approach ensures that the model continues to leverage the discriminative feature space established during pretraining while adapting to the specific characteristics of the 2025 dataset, including potential domain shifts.

To enhance model performance, we explored various loss function combinations, incorporating the Center loss alongside the widely adopted ArcFace loss. The ArcFace loss maintains class separability, while the Center loss encourages intra-class compactness by pulling features toward their respective class centers, as detailed in Section 2.4.

Additionally, for machines in the 2025 dataset lacking attribute information, we applied a pseudo-labeling strategy using the pretrained weights to assign temporary labels. This process, which involves extracting feature embeddings, visualizing them with UMAP, and clustering to determine pseudo-label classes, is elaborated in Section 2.5. The pseudo-labeled data were integrated into the transfer learning phase, augmenting the training set and improving the model’s ability to generalize to machines with incomplete attribute information.

2.2. Multi-layer aggregation

The multi-layer aggregation strategy, first introduced in the field of speaker recognition [8], is a methodology that leverages the outputs from all layers of a transformer model to construct a comprehensive feature representation. In speaker recognition, this approach is employed to capture local features, such as pitch, intonation style, and pronunciation patterns, while simultaneously modeling global context to account for long-range dependencies inherent in variable-length speech sequences [9]. By integrating these diverse features, the model achieves a robust representation that encapsulates both fine-grained and overarching audio characteristics.

In our work, we adapted this approach, aggregating features from lower and higher layers to capture both local and global audio characteristics. Lower layers extract fine-grained patterns, such as specific frequency components or short-term temporal variations, while higher layers encode long-term operational characteristics, such as a machine’s periodic hum. To implement this, we concatenated the outputs of all transformer layers along the feature dimension, followed by a sequence of layer normalization, a linear layer, a GELU activation, and another linear layer. This process integrates multi-scale features into a unified representation, enhancing the model’s ability to handle diverse audio patterns.

By combining features sensitive to different temporal scales, multi-layer aggregation enhances the model’s ability to detect diverse anomalies, from short-term events like sudden high-

frequency noise to long-term deviations like disruptions in rhythmic patterns. Additionally, this approach improves generalization across domains. Lower layers provide domain-invariant features, such as basic frequency patterns, while higher layers capture domain-specific information, such as environmental noise characteristics. Our experiments demonstrate that multi-layer aggregation significantly enhances model performance, with particularly notable improvements in the target domain, where data availability is limited.

2.3. Attentive statistics pooling

To effectively aggregate the sequence outputs of the audio encoder, we employed attentive statistics pooling, a technique widely used in audio processing to capture both local and global characteristics of variable-length sequences [10]. This method enhances the model’s ability to generate a fixed length representation from the transformer’s time-distributed outputs, which is critical for downstream classification and anomaly detection tasks.

In our implementation, attentive statistics pooling was applied to the output of the transformer encoder, following the multi-layer aggregation step described in Section 2.2. The pooled representation was processed through two linear layers with a ReLU activation function between them, using a dropout of 0.2, where the intermediate dimension was set to three times the input dimension, and finally projected to a 768-dimensional space for input to the loss function.

2.4. Loss function

ArcFace loss, or Additive Angular Margin Loss, was first introduced in face recognition and has since been widely adopted in various classification tasks for its ability to enhance discriminative power [11]. This loss function modifies the standard softmax loss by adding an angular margin, which enforces a larger angular distance between different classes in the feature space, thereby improving class separability.

ArcFace loss is particularly effective for distinguishing between different machine types and their attributes, such as operating speeds or environmental noise conditions. By treating each unique combination of machine type and attribute information as a distinct class, the ArcFace loss ensures that the audio encoder learns to extract features that are highly discriminative. This is crucial for the DCASE2025 Task 2, where the model must generalize across domains, and clear class separations in the feature space facilitate robust anomaly detection.

In the transfer learning stage, we aimed to adapt the pre-trained model to the training dataset for this year. To achieve this, we introduced the **Center loss** in conjunction with the ArcFace loss. Center loss is designed to learn a center for each class and to pull the feature vectors of the same class closer to their respective centers, thereby reducing intra-class variations while maintaining the inter-class separations enforced by ArcFace loss [12]. The center loss is mathematically defined as:

$$L_C = \frac{1}{2} \sum_{i=1}^m \|x_i - c_{y_i}\|_2^2, \quad (1)$$

where m is the number of samples in a mini-batch, x_i represents the feature embedding of the i -th sample, c_{y_i} denotes the learnable class center corresponding to the class label y_i , and $\|\cdot\|_2^2$ is the squared L2 norm. This formulation encourages feature

embeddings of the same class to cluster tightly around their respective centers, enhancing the model’s ability to generalize to the target dataset. The class centers c_{y_i} are updated during training to reflect the evolving features distributions, complementing the angular margin-based separation provided by ArcFace loss.

During transfer learning, we explored four loss configurations for transfer learning. First, we used ArcFace loss alone, as in pre-training, to maintain strong class separability for machine types and attributes. Second, following the original implementation of Center loss, we used a combination of Cross Entropy (CE) loss and Center loss as the loss function. Third, we combined Focal loss with Center loss. Focal loss is a loss function that addresses class imbalance by focusing on hard-to-classify examples, down-weighting well-classified ones [13]. Its formula is given by:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t). \quad (2)$$

We set γ to 5 in all our experiments. Last, we combined ArcFace loss with Center loss, weighing Center loss at 0.9 and ArcFace loss at 0.1 to compute the final loss. The combination of a discriminative loss (CE, Focal or ArcFace) with Center loss ensures that class separability is preserved while Center loss enhances intra-class compactness by encouraging features to converge toward their class centers, significantly improving performance.

Table 1: the number of classes after pseudo-labeling based on the pretrained weights.

Machine	EAT	BEATs
ToyTrain	4	4
Slider	5	5
Bearing	5	4
AutoTrash	2	2
Polisher	4	4
ScrewFeeder	5	3
ToyPet	4	3
Total	29	25

2.5. Pseudo-labeling

To leverage data from machines lacking attribute information, we implemented a pseudo-labeling strategy to enhance model performance. This approach assigns temporary labels to unlabeled data enabling their use in transfer learning. The pseudo-labeling process begins with extracting feature embeddings from the unlabeled machines using a pre-trained model. To analyze the structure of these embeddings, we applied Uniform Manifold Approximation and Projection (UMAP) [14] to reduce their dimensionality and visualize them in two dimensions. UMAP preserves both local and global data structures, making it effective for identifying clusters that may correspond to different operational conditions or machine states.

Based on the UMAP visualization, we adopted a dual approach to assign pseudo-labels. For machines exhibiting distinct clusters in the feature space, the number of pseudo-label classes was determined directly by the number of observed clusters, reflecting natural groupings in the data. For machines with ambiguous or overlapping clusters, we employed agglomerative hierarchical clustering to generate a dendrogram, which represents the hierarchical structure of the data. Using a top-down approach, we interpreted the dendrogram and set a distance threshold between 20 and 40 to identify candidate pseudo-label classes. This

threshold was chosen empirically to balance the granularity of clustering, ensuring meaningful and distinct class assignments.

To address potential data imbalance, we calculated the number of samples per candidate pseudo-label class and selected the

configuration with the least imbalance for each machine type. Data imbalance, where some classes have significantly more

Model	Stage	Agg	Loss	AUC source	AUC target	pAUC	hmean	System ID
EAT	1	X	ArcFace	68.44	66.64	56.62	63.45	
		O		67.24	71.58	57.41	64.85	
	2	O	ArcFace	67.80	72.70	57.73	65.52	1
			CE + Center	69.33	72.71	58.07	66.09	2
			Focal + Center	69.41	72.74	58.07	66.12	
ArcFace + Center	68.55	73.31	57.61	65.81	3			
BEATs	1	X	ArcFace	69.70	61.94	56.06	62.07	
		O		69.74	70.41	56.71	64.97	
	2	O	ArcFace	70.18	70.77	57.60	65.59	4
			CE + Center	69.84	70.83	57.62	65.52	5
			Focal + Center	70.01	70.74	57.55	65.55	
			ArcFace + Center	69.90	70.92	57.95	65.66	6
			Ensemble					
	Ensemble 1		69.49	72.82	59.51	66.78	1	
	Ensemble 2		69.39	72.88	59.49	66.75	2	
	Ensemble 3		69.63	72.86	59.21	66.70	3	
	Ensemble 4		69.61	72.80	59.25	66.70	4	

Table 2: Performance of single systems and ensemble systems. The best performance for each model is highlighted. Note that the system ID corresponds to the system index used in the ensemble described in Table 3, while the submission ID refers to the index used for the final submissions.

samples than others, can bias model training, so this step ensures a balanced representation of pseudo-labels. The process was applied individually to each machine type, recognizing their unique acoustic characteristics and clustering behaviors. The number of classes resulting from pseudo-labeling is shown in Table 1.

2.6. Testing phase

In the testing phase, the trained audio encoder generates embeddings for both training and test audio samples. KNN algorithm is employed to compute the distances between the embeddings of test samples and those of normal training samples. Anomaly scores are derived from these distances, typically as the average distance to the k nearest neighbors, where larger distances indicate a higher likelihood of an anomaly. We used $k=1$ in all our experiments. Also, to address the class imbalance, SMOTE was used to generate synthetic embeddings for the underrepresented normal samples in the target domain, balancing the dataset used for KNN-based anomaly detection. Furthermore, to account for variations across machines, we calculated per-machine statistics from the training dataset and normalized the embeddings of test samples using the corresponding mean and standard deviation for each machine.

3. EXPERIMENT

This section outlines the experimental setup and results. Regarding the supplementary data, a distinguishing feature of this year's competition, we attempted a data augmentation technique using clean noise but could not achieve meaningful results.

3.1. Experimental setup

We utilized EAT [15] and BEATs [16] models for our experiments. The input audio was preprocessed in the same manner as each model's pretraining approach. For efficient training, we applied Low-Rank Adaptation (LoRA) [17] with a rank of 64 to the query, value, and projection layers of both BEATs and EAT models. All systems were trained for 40 epochs, with the harmonic mean (hmean) of AUC source, AUC target, and pAUC on the development test dataset monitored to select the weights from the epoch showing the best performance.

When maintaining the same number of trainable parameters as in pretraining, we observed severe overfitting. To address this, we froze most model parameters and applied LoRA specifically to the adaptor layers used for layer aggregation. Additionally, during the transfer learning phase, we froze the existing model parameters and updated only the LoRA parameters of the adaptor. However, when updating the EAT model using center loss, we found that setting all model parameters as learnable, without using adaptor LoRA, yielded better performance.

The learning rate was updated using a cosine warmup restart schedule. For pretraining, the initial learning rate was set to $1e-5$, the minimum learning rate to $1e-7$, with a warmup of 1 epoch and restarts every 5 epoch. Weight decay was set to 0.0001, batch size to 32, and accumulation steps to 8.

For transfer learning, the initial learning rate was set to $1e-7$, the minimum learning rate to $5e-10$, weight decay to 0.01, batch size to 16, and accumulation steps to 16. Other settings remained the same as in pre-training.

3.2. Experimental results

Table 2 summarizes the performance of our systems. Specifically, the introduction of layer aggregation significantly enhanced the AUC performance in the target domain for both models. Notably,

System ID	1	2	3	4	5	6
Ensemble 1	0.18	0	0.42	0	0.04	0.36
Ensemble 2	0.18	0.18	0.24	0.12	0.12	0.16
Ensemble 3	0	0.6	0	0	0.4	0
Ensemble 4	0	0	0.6	0	0	0.4

Table 3: Combination coefficients of four submitted systems

source domain. The best system based on EAT and BEATs achieved a harmonic mean of 66.12 and 65.66 respectively.

To determine the final submitted system, we conducted a grid search to explore various combinations of single systems, using the weights presented in Table 3. During this exploration, we observed that systems utilizing Focal loss combined with Center loss consistently exhibited relatively lower performance. Consequently, these systems were excluded from the final submission. Submissions 1 and 2 were generated by identifying optimal weights for ensembles of multiple single systems. In contrast, Submissions 3 and 4 were created by performing model-level ensembling of systems using Center loss and a combination of ArcFace loss and Center loss configuration, respectively. The results of the ensemble systems for each machine are presented in Table 4.

4. CONCLUSION AND FUTURE WORKS

This paper described the AISTAT Lab’s system for first-shot unsupervised anomalous sound detection. In response to the expanded training data available compared to previous challenges, we adopted a two-stage training framework to maximize data utilization. Additionally, we employed layer aggregation to integrate multi-scale audio representations, capturing both fine-grained acoustic patterns and long-term contextual dependencies. In the transfer learning phase, we enhanced the standard ArcFace loss with Center loss, maintaining class separability while promoting intra-class cohesion, resulting in notable performance gains. As future work, we plan to explore the integration of clean noise and clean machine sound data, which were not utilized in this study, to further enhance model robustness and performance.

5. ACKNOWLEDGMENT

This research was supported by the Chung-Ang University Research Scholarship Grants in 2024.

6. REFERENCES

[1] 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,” *In arXiv e-prints: 2506.10097*, 2025.

the EAT model improved its target domain AUC from 66.64 to 71.58, while the BEATs model saw an increase from 61.94 to 70.41. The difference in harmonic mean between the models was marginal. However, EAT generally outperformed BEATs in the target domain, while BEATs showed better performance in the

[2] 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 Detection and Classification of Acoustic Scenes and Events Workshop (DCASE), 1–5*. Barcelona, Spain, November, 2021.

[3] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, “MIMII DG: sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task,” in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*. Nancy, France, November, 2022.

[4] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, “First-shot anomaly detection for machine condition monitoring: a domain generalization baseline,” in *Proceedings of 31st European Signal Processing Conference (EUSIPCO), pages 191–195*, 2023.

[5] 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 2024 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring,” *In arXiv e-prints: 2406.07250*, 2024.

[6] Z. Lv, A. Jiang, B. Han, Y. Liang, Y. Qian, X. Chen, J. Liu, P. Fan, “AITHU system for first-shot unsupervised anomalous sound detection,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Tokyo, Japan, October 2024.

[7] A. Jiang, X. Zheng, Y. Qiu, W. Zhang, B. Chen, P. Fan, W. Q. Zhang, C. Lu, J. Liu, “THUEE system for first-shot unsupervised anomalous sound detection,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE), Tokyo, Japan, October 2024*.

[8] J.-w. Jung, Y. J. Kim, H.-S. Heo, B.-J. Lee, et al., “Pushing the limits of raw waveform speaker recognition,” in *Proc. Interspeech*, 2022.

[9] Y. Zhang, Z. Lv, H. Wu, S. Zhang, et al., “MFA-Conformer: Multi-scale feature aggregation conformer for automatic speaker verification,” in *Proc. Interspeech*, 2022.

[10] B. Desplanques, J. Thienpondt, and K. Demuyne, “Ecapatdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” *arXiv preprint arXiv:2005.07143*, 2020.

[11] 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.

- [12] Y. Wen, K. Zhang, Z. Li, Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Computer vision—ECCV 2016: 14th European conference, amsterdam, the netherlands, proceedings, part VII 14* (pp. 499-515). Springer International Publishing, October 11–14, 2016.
- [13] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988), 2017.
- [14] L. McInnes, J. Healy, J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.
- [15] W. Chen, Y. yang, Z. Ma, Z. Zheng, and X. Chen, “EAT: Self-supervised pre-training with efficient audio transformer,” *arXiv preprint arXiv:2401.03497*, 2024.
- [16] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, “Beats: Audio pre-training with acoustic tokenizers,” *arXiv preprint arXiv:2212.09058*, 2022.
- [17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, ... and W. Chen, “Lora: Low-rank adaptation of large language models,” *ICLR, 1(2)*, 3, 2022.

Table 4: Detection results of four submitted systems on the development set

Machine	Metric	Ensemble 1	Ensemble 2	Ensemble 3	Ensemble 4
ToyCar	AUC_s	62.52	62.84	62.48	62.48
	AUC_t	69.72	69.76	68.96	68.96
	pAUC	51.00	51.63	50.68	50.42
	hmean	60.07	60.46	59.72	59.88
ToyTrain	AUC_s	73.72	73.36	74.24	74.40
	AUC_t	77.08	77.12	77.36	76.96
	pAUC	60.42	60.21	61.37	61.68
	hmean	69.62	69.43	70.27	70.35
Bearing	AUC_s	57.20	57.28	57.44	57.52
	AUC_t	74.84	75.32	75.08	74.56
	pAUC	56.16	56.21	55.84	55.58
	hmean	61.66	61.82	61.68	61.49
Fan	AUC_s	62.24	62.60	62.44	62.08
	AUC_t	62.04	61.52	62.48	62.08
	pAUC	52.00	51.79	51.84	51.84
	hmean	58.35	58.21	58.47	58.25
Gearbox	AUC_s	68.76	68.72	70.16	70.76
	AUC_t	79.16	79.36	79.40	78.84
	pAUC	66.53	66.79	66.16	66.89
	hmean	71.08	71.22	71.49	71.83
Slider	AUC_s	90.32	90.32	90.48	90.36
	AUC_t	74.88	74.68	74.76	74.04
	pAUC	64.63	64.05	63.84	63.79
	hmean	75.19	74.86	74.83	74.53
valve	AUC_s	82.24	82.40	81.16	80.44
	AUC_t	75.40	75.20	74.88	75.80
	pAUC	71.58	71.63	70.26	70.42
	hmean	76.16	76.15	75.17	75.33