Decoupling Detectors for Scalable Anomaly Detection in AIoT Systems with Multiple Machines

Qijun Hou[†], Anbai Jiang[†], Wei-Qiang Zhang[†], Pingyi Fan[*†] and Jia Liu[†‡]
[*]Email: fpy@tsinghua.edu.cn
[†]Department of Electronic Engineering, Tsinghua University, Beijing, China.
[‡]Tsinghua AI Plus, Beijing, China.

Abstract*—The fast-developing Artificial Internet of Things (AIoT) technologies enable the consistent monitoring of multiple machines, by which machine failures can be detected in the early phases, and production efficiency and system management can be greatly promoted, bringing huge significance for anomaly detection. However, in most cases, anomalies are not provided for training, and the lack of direct supervision deprecates the anomaly detection performance. For the application viewpoint, the detector is required to generalize well on multiple machines, except for being computationally efficient. The computational cost is strictly limited, which is a great challenge for mobile and embedded devices. In face of these issues, we propose MobileAnoNet, which decouples an end-to-end detector into a front-end feature extractor and a back-end anomaly detector. The front-end extractor, consuming most computation, is unified for all machine types, while the back-end detector is specialized for each machine type, improving the detection capacity. The model is trained by handy labels of machine types and working conditions, in which multiple classification heads are attached behind the feature extractor during training. The performance of the model is evaluated on two DCASE datasets focusing on machine audio anomaly detection. It's shown that MobileAnoNet achieves a general improvement of 6.9% and 8.8% on two datasets, respectively. The ablation study demonstrates that multi-task learning promotes the general representation capacity. The source code is available at: www.github.com/hqj-les30/MobileAnoNet

***Index Terms*—anomaly detection, machine audio, deep learning**

I. Introduction

With the fast development of Artificial Intelligence (AI) technologies and Internet of Things (IoT) devices, the Artificial Internet of Things (AIoT) emerges as AI-powered IoT systems where powerful deep learning models can analyze enormous amounts of data collected by ubiquitous devices, which could be a revolutionary technology for modern manufacturing, where by setting up sensor networks, parsing the data and digging out hidden patterns, production efficiency can be greatly improved, and risk of failures can be dramatically reduced. Anomaly detection is one of the key technologies for AIoT-based machine systems. In general, it is intractable to collect sufficient data for all kinds of machine failures, and anomalies have to be detected with negligible or even zero prior knowledge, which is extremely challenging in real production scenarios. In order to simulate the worst case, there are no anomaly data in our training dataset. On the other hand, the scalability problem arises when it comes to AIoT systems with multiple machines since different machines may have distinct patterns, and simultaneously deploying multiple machine-specific detectors impose a huge burden on computational resources and device maintenance.

To tackle this problem, we propose MobileAnoNet, which decouples an end-to-end anomaly detector into a front-end feature extractor and a back-end anomaly detector, where the front-end extractor maps the sparse input into a compact feature space, and the back-end detector estimates the anomalous degree of an embedding by calculating an anomaly score. The front-end extractor is unified for all machine types, i.e., the same model structure and the same set of parameters, which means only one front-end extractor is required for deployment. As for the back-end detectors, we select the best-performing back-end detector for each machine type, all of which are distance-based detectors, so as to reduce the computational cost of unshared modules. By reusing the same front-end extractor and simplifying back-end detectors, the proposed scheme significantly reduces the computational cost.

Since only normal data are provided for training, the unified extractor must be either unsupervised or trained by a proxy task that utilizes auxiliary labels such as the inner categories of normal data. In the MobileAnoNet scheme, the extractor is supervised by both the machine type label and the working condition labels, for example, the velocity, the acceleration, and the voltage, which are handy to collect. It is believed that multi-task learning promotes the general representation capacity. During training, multiple classification heads are attached to the feature extractor, each of which corresponds to a specific label. Since different machine types correspond to different types of working conditions, a mask is generated to select the appropriate classification heads for each input. During testing, all classification heads are discarded, and the feature extractor is concatenated with multiple anomaly detectors, each of which is specialized for a machine type.

The experiment is conducted on the datasets of both DCASE 2022 Challenge TASK 2 and DCASE 2023 Challenge TASK 2 [1], [2]. MobileAnoNet achieves a general improvement of 6.9% on the DCASE22 dataset and a general improvement of 8.8% on the DCASE23 dataset. The performance on the same machine type across two datasets is consistent: MobileFaceNet generally performs better onmachines with periodic

This work was supported by the National Key Research and Development Program of China (Grant NO.2021YFA1000504) and the National Natural Science Foundation of China under Grant No. 62276153.

and stationary sound such as gearbox, slider, and valve, while the performance degrades on unstationary machines such as ToyCar and ToyTrain. To validate the effectiveness of multi-task learning, we compare MobileAnoNet with two models trained only by one kind of label. It is observed that multi-task learning generalizes better on distinct machine types.

The main contributions are summarized as follows:

1) We propose MobileAnoNet, which can be decoupled into a front-end feature extractor and a back-end anomaly detector. For AIoT systems with multiple machines, the front-end extractor is reused, and the back-end detectors are specialized for each machine type, which yields great scalability while reducing the computational cost.
2) MobileAnoNet is trained by multi-task learning, leveraging the machine type label and the working condition labels, which are handy to collect. It is observed that multi-tasking enforces the model to generalize well on distinct machines.

The rest of this paper is organized as follows. Section II introduces previous works on anomaly detection in industrial scenarios. Section III elaborates on the proposed MobileAnoNet in detail. Section IV presents experimental results on two datasets and the ablation study. Finally, section V concludes the paper.

II. Related Work

As a vital technology for production efficiency and reliability, anomaly detection for industrial manufacturing has been widely explored in multiple modals, ranging from industrial images [3], machine audio [4], [5] and time series signals of sensors [6]. Most anomaly detection models can be decomposed into front-end feature extractors and back-end anomaly detectors, which is consistent with our framework, though not specifically emphasized in the original research.

A. Back-end Anomaly Detector

Categorized by detection mechanism, back-end detectors can be intuitively divided into distance-based methods, reconstruction-based methods, and probabilistic methods. Distance-based detectors are most commonly used due to their simplicity and effectiveness. K-nearest neighbor (KNN) [7] leverages the average distance from the query embedding to the k-nearest embeddings of normal samples as the anomaly score, while the distance to the average embedding can also be utilized for detection [4], [5]. Local outlier factor (LOF) [8] improves KNN by estimating the neighbor density of both normal and query embeddings. Support vector data description (SVDD) [9] and Deep SVDD [10] map the embeddings into a compact hyper-sphere where the distance to the center is utilized as the anomaly score. Commonly used reconstruction-based models are Principal Component Analysis (PCA) and Autoencoder [4], [11], where they are only trained to reconstruct normal samples, and it is likely that anomalies will have larger reconstruction error. Probabilistic methods attempt to model the distribution of normal embeddings. DAGMM [12] models the concatenation of the latent space and the reconstruction error of an autoencoder by the Gaussian Mixture Model (GMM). NF-CDEE [5] utilizes normal flow to model the distribution of some mel-spectrogram filters conditioned on the rest.

B. Front-end Feature Extractor

Feature extractors aim to extract compact and semantic embeddings from sparse input and decouple the understanding task from the detection task for back-end detectors. Since no anomalies are provided for training, the feature extractors are either pre-trained models or trained by a proxy task. This scheme has demonstrated the efficiency on multiple anomaly detection tasks in industrial scenarios [12]. PatchCore [3] obtains the embeddings of industrial product images via a WideResNet pre-trained on ImageNet, before sending them to a KNN detector. Lopez et al. [5] processed machine audio by an XVector and a distance-based detector.

Since only normal data are provided for training, the front-end extractor and the back-end detector are not optimized end-to-end, which does not impose a restriction on the combination of different extractors and detectors, resulting in great scalability for multiple anomaly detection tasks. In the proposed scheme, for AIoT systems with multiple and heterogeneous machines, only one unified feature extractor is utilized for data representation, and we can select the best-performing back-end detector for each machine.

III. Proposed Method

The proposed MobileAnoNet is elaborated in this section.

A. Audio Representation

Short Time Fourier Transform (STFT) is adopted as the input of MobileAnoNet, which better preserves semantic information than mel-spectrogram. The input clips are 10s or 12s in length with a sampling rate of 16kHz, and we use a 2048-point Fast Fourier Transform (FFT) with a hop length of 512 points, resulting in a 1025×313 spectrogram for a 10s clip or a 1025×376 spectrogram for a 12s clip (both with padding on two edges when calculating STFT). The clip spectrogram is then sliced into segments of 1025×64 to reduce the size of the neural network. During training, we randomly sample 20 segments from each clip spectrogram, while during inference, segments are extracted compactly with a hop length of 1 frame, which is believed to promote detection performance.

B. Multi-task Learning for Training

It is believed that multi-task learning promotes overall representation capacity and induces the model to be more generalized for multiple tasks. Thus, MobileAnoNet is supervised by both the machine type label and the working condition labels (velocity, weight, etc) simultaneously, all of which are handy to collect for AIoT systems. The framework of MobileAnoNet is illustrated in Fig 1, which comprises a unified feature extractor and multiple classification heads, each of which is dedicated to a specific label. The audio

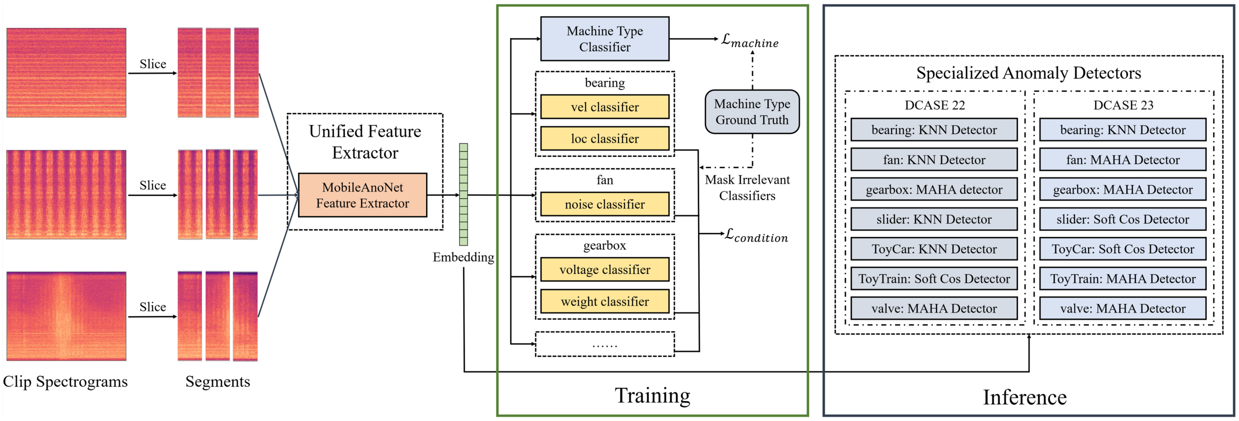


Fig. 1: General Framework of MobileAnoNet

spectrograms of different machine types are randomly batched and transformed into their respective embeddings, which are the output of the unified feature extractor, then sent into the corresponding classification heads respectively, since different machine types are assigned to different working condition labels.

The loss function of MobileAnoNet is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{machine} + \mathcal{L}_{condition} \tag{1}$$

where $\mathcal{L}_{machine}$ and $\mathcal{L}_{condition}$ are the loss given by the machine type label and the working condition labels. $\mathcal{L}_{machine}$ is implemented by Center Loss [13]:

$$\begin{aligned}\mathcal{L}_{machine} &= \mathcal{L}_s + \lambda \mathcal{L}_c \\ &= \frac{1}{n}\sum_{i=1}^{n} CE(W_m^T x_i, y_i^m) + \lambda \cdot \frac{1}{n}\sum_{i=1}^{n} \|x_i - c_{y_i^m}\|_2^2\end{aligned} \tag{2}$$

where the first term is the cross entropy loss between each predicted machine type $W_m^T x_i$ and the ground truth y_i^m, where W_m^T is the weight of the classification head. The second term is the Mean Square Loss (MSE) between each embedding x_i and the center of the corresponding class $c_{y_i^m}$, where λ is used for balancing two terms.

$\mathcal{L}_{condition}$ is the cross entropy loss of working conditions, which is formulated as follows:

$$\mathcal{L}_{condition} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k_i} CE(W_{c_j}^T x_i, y_i^{c_j}) \tag{3}$$

where k_i is the number of working condition labels x_i possesses, c_j is the j th working conditions of x_i and $W_{c_j}^T x_i$ is the predicted working condition of c_j. Since different machine types correspond to different types of working condition, a mask is generated to select the appropriate classification heads for each input.

TABLE I: Architecture of Front-end Feature Extractor

Input	Operator	t	c	n	s
1×1025×64	conv3×3	-	16	1	2
16×513×32	depthwise conv3×3	-	16	1	1
16×513×32	bottleneck	2	16	5	2
16×257×16	bottleneck	4	32	1	2
32×129×8	bottleneck	2	32	6	1
32×129×8	bottleneck	4	32	1	2
32×65×4	bottleneck	2	32	2	1
32×65×4	conv1×1	-	128	1	1
128×65×4	GDConv65×12	-	128	1	1

We use the same notations as MobileNetV2 [14]: t is the expansion factor, c is the output channel, n is the repetition time, and s is the stride. GDConv [15] is a depth-wise convolution which has the same kernel size as the input feature map.

C. *Network Architecture*

Much effort has been made to adapt neural networks to mobile and embedded devices, where computational and storage capabilities are extremely limited. Among them, the MobileNet family [14]–[16] massively adopt depth-wise separable convolution layers and bottlenecks to significantly reduce the number of parameters while achieving good classification performance on multiple datasets.

MobileAnoNet adopts a similar architecture as the feature extractor, yet proportionally reduces the output channel of each convolution layer, since the input spectrograms (1025×64) are bigger than images (224×224). Table I depicts the detailed architecture of the front-end feature extractor. Each classification head consists of a linear layer and a batch normalization (BN) layer [17], which maps the output of the feature extractor to the predicted logits.

D. *Back-end Detectors*

In MobileAnoNet, six types of back-end detectors are utilized: KNN, LOF, COS, MAHA, Soft COS, and Soft MAHA. The KNN detector calculates the average distance between a query embedding and the top k nearest embeddings of normal samples. The LOF detector extends the KNN detector

TABLE II: Results on DCASE22

	Official-AE [18]				Official-MBN [18]				Guan et al. [19]				MobileAnoNet			
	AUCs	AUCt	pAUC	hmean	AUCs	AUCt	pAUC	hmean	AUCs	AUCt	pAUC	hmean	AUCs	AUCt	pAUC	hmean
bearing	54.4	58.4	52.0	54.8	60.6	59.9	57.1	59.2	55.6	**74.7**	52.6	59.5	**71.5**	63.5	**64.5**	**66.3**
fan	78.6	47.2	57.5	58.5	70.8	48.2	56.9	57.2	69.9	62.7	58.5	63.4	**84.5**	**66.6**	**66.0**	**71.4**
gearbox	68.9	62.6	55.8	62.0	69.2	56.2	56.0	59.9	71.4	62.7	56.0	62.7	**87.3**	**81.2**	**70.1**	**78.8**
slider	78.0	47.7	55.8	58.0	65.2	38.2	54.7	50.2	89.5	**75.5**	61.7	73.8	**95.5**	70.8	**70.6**	**77.4**
ToyCar	90.4	34.8	52.7	51.1	59.1	52.0	52.3	54.3	**66.3**	**82.7**	**59.9**	**68.4**	65.4	73.5	58.7	65.3
ToyTrain	76.3	23.4	50.5	39.6	57.3	45.9	51.5	51.1	**85.5**	**52.1**	**57.8**	**62.3**	76.0	49.5	56.9	58.9
valve	52.0	49.5	50.4	50.6	67.1	57.2	62.4	62.0	71.2	52.6	60.3	60.4	**88.1**	**89.4**	**77.2**	**84.6**
all_hmean	68.7	41.9	53.4	52.5	63.8	50.0	55.7	55.9	71.2	64.3	58.0	64.0	**80.0**	**68.5**	**65.6**	**70.9**

by estimating the local density in the feature space. Cosine distance is selected as the distance metric for both the KNN and the LOF detector, while the neighbor number k is selected as 2 and 4 for KNN and LOF, respectively. The COS detector and the MAHA detector calculate the cosine distance and Mahalanobis distance between a query embedding and the average embedding of all normal samples of the same class as the anomaly score, respectively.

Both datasets have a domain shift problem, where most clips (99:1) for training are from the source domain, while equal amounts of source and target clips are presented for testing. This problem mirrors the variations of working conditions through time in reality, and how to distinguish between normal target clips and anomalous source clips is a challenging problem. In face of this problem, we also introduce a soft decision mechanism for the COS and MAHA detectors, referred to as Soft Cos and Soft MAHA respectively, in which the average embeddings are saved for both the source domain and the target domain and the respective distances to two embeddings are calculated. The smaller distance is chosen as the final anomaly score. Among six detectors, the best-performing one is selected for each machine type to promote performance, which yields great scalability across distinct machines.

The general inference procedure is also illustrated in Fig 1. The STFT of a query clip is calculated at first (1025×313 for 10s, 1025×376 for 12s), then sliced into compactly overlapped segments (1025×64), which are further processed by the feature extractor into 128 dimension embeddings. Embeddings from the same clip are averaged into a clip embedding, then sent into the corresponding anomaly detector for estimation.

IV. Experiments

A. Dataset

The proposed MobileAnoNet model is tested on the DCASE22 and DCASE23 datasets. The DCASE datasets consist of audio clips recorded from various machines with labels of the working conditions of the machines. The training set contains only normal data, while the test set has an equal number of normal and anomalous data instances. For each machine type, the DCASE22 dataset contains three sections, while the DCASE23 dataset contains one section. Within each section, the data is divided into the source domain and target domain, with a ratio of 99:1 for the source domain and target domain data in the training set and an equal number of data instances from both domains in the test set.

B. Evaluation Method

The performance is measured by the Area under the Receiver Operating Characteristic (ROC) Curve (AUC) and partial AUC (pAUC). AUC is calculated on both the source domain and the target domain, and it is worth to note that the calculation of AUC on each domain takes all anomalous samples into account, which can be formulated as follows:

$$AUC_{m,n,d} = \frac{1}{N_d^- N_n^+} \sum_{i=1}^{N_d^-} \sum_{j=1}^{N_n^+} \mathcal{H}(\mathcal{A}(\mathbf{x}_j^+) - \mathcal{A}(\mathbf{x}_i^-)) \tag{4}$$

where x_i^- and x_j^+ are the normal and anomalous samples of machine m, section n and domain d, respectively. $\mathcal{A}(\cdot)$ is the anomaly detector and $\mathcal{H}(\cdot)$ is the heaviside function.

The pAUC is an AUC calculated from a portion of the ROC curve over the pre-specified range of interest. Its calculation formula is as Equation (5).

$$pAUC_{m,n,d} = \frac{1}{\lfloor pN_d^- \rfloor N_n^+} \sum_{i=1}^{\lfloor pN_n^- \rfloor} \sum_{j=1}^{N_n^+} \mathcal{H}(\mathcal{A}(\mathbf{x}_j^+) - \mathcal{A}(\mathbf{x}_i^-)) \tag{5}$$

where p is chosen as 0.1.

Let AUCs, AUCt denote the harmonic mean of the AUC of the source domain and the target domain across all sections, respectively. The general performance on each machine is measured by the harmonic mean of AUCs, AUCt, and pAUC. A general harmonic mean calculated over all machine types is also presented to demonstrate the scalability over different machines.

C. Experiment Settings

MobileAnoNet is trained on all machine types on two datasets respectively. The weight parameter λ for Center Loss is set to 0.2, and the batch size is set to 160. The models are trained 15 epochs using an Adam [21] optimizer.

On the DCASE22 dataset, MobileAnoNet is compared with two official baselines (Official-AE and Official-MBN) [18] and the work of Guan et al. [19]. Official-AE is a dense autoencoder that reconstructs the mel-spectrogram, and

TABLE III: Results on DCASE23

	Official-AE [20]				Official-AEMA [20]				MobileAnoNet			
	AUCs	AUCt	pAUC	hmean	AUCs	AUCt	pAUC	hmean	AUCs	AUCt	pAUC	hmean
bearing	57.9	57.0	48.6	54.2	65.2	**55.3**	51.4	56.7	**70.1**	51.7	**56.2**	**58.4**
fan	80.2	36.2	59.0	52.6	**87.1**	46.0	59.3	59.9	83.8	**78.1**	**73.3**	**78.2**
gearbox	60.3	60.7	53.2	57.9	71.9	70.8	54.3	64.6	**84.6**	**80.3**	**67.2**	**76.6**
slider	70.3	48.8	56.4	57.2	84.0	73.3	54.7	68.5	**99.0**	**99.4**	**95.9**	**98.1**
ToyCar	70.1	**46.9**	**52.5**	**54.9**	**74.5**	43.4	49.2	52.8	64.7	41.6	48.9	50.0
ToyTrain	57.9	**57.0**	**48.6**	**54.2**	56.0	42.5	48.1	48.2	**60.1**	46.0	47.9	50.6
valve	55.4	50.7	51.2	52.3	56.3	51.4	51.1	52.8	**88.9**	**79.7**	**59.2**	**73.7**
hmean	63.6	49.7	52.5	54.7	68.8	52.4	52.4	56.9	76.5	62.0	60.9	**65.7**

TABLE IV: Ablation Study on DCASE23

	Only Machine Type				Only Working Conditions				MobileAnoNet			
	AUCs	AUCt	pAUC	hmean	AUCs	AUCt	pAUC	hmean	AUCs	AUCt	pAUC	hmean
bearing	65.5	61.1	**58.1**	61.4	69.1	**66.3**	54.8	**62.8**	**70.1**	51.7	56.2	58.4
fan	74.3	66.6	55.6	64.6	87.7	69.2	68.4	74.1	**83.8**	**78.1**	**73.3**	**78.2**
gearbox	80.2	74.0	60.8	70.7	82.3	78.5	60.7	72.5	**84.6**	**80.3**	**67.2**	**76.6**
slider	99.4	98.5	94.5	97.4	93.9	95.8	79.6	89.2	**99.0**	**99.4**	**95.9**	**98.1**
ToyCar	64.9	**45.4**	**51.7**	**52.8**	**72.3**	42.3	48.8	51.8	64.7	41.6	48.9	50.0
ToyTrain	60.3	43.9	48.4	50.0	**63.3**	**53.3**	**49.2**	**54.7**	60.1	46.0	47.9	50.6
valve	74.3	70.6	52.5	64.3	83.8	77.3	55.4	69.9	**88.9**	**79.7**	**59.2**	**73.7**
all_hmean	72.4	61.4	57.7	63.3	**77.6**	**64.8**	58.0	**65.8**	76.5	62.0	**60.9**	65.7

Official-MBN employs a MobileNetV2 to classify the section label.

On the DCASE23 dataset, MobileAnoNet is compared with two official baselines (Official-AE and Official-AEMA) [20]. Official-AE and Official-AEMA are the same dense autoencoder model, yet utilize norm distance and Mahalanobis distance as the scoring metric respectively.

D. Result on DCASE22

The results on the DCASE22 dataset are shown in Table II, which illustrates the AUC of 7 machine types in the source domain and target domain(AUCs, AUCt), as well as the overall pAUC of each machine type. The best performance for each metric is marked in bold. It is observed that MobileAnoNet outperforms the baseline systems on the bearing, fan, gearbox, slider, and valve and achieves a general improvement of 6.9%.

E. Results on DCASE23

The results on the DCASE23 dataset are shown in Table III. Our method consistently outperforms other approaches in terms of average performance across all machine types on both datasets. The performance of each machine type remains consistent with the DCASE22. The overall harmonic mean has improved by 8.8%.

F. Visualization

Fig.2 displays the distribution of embeddings obtained from all normal audio samples in the DCASE23 dataset, using t-SNE to visualize on a 2-D flat [22]. The target domain is indicated by different markers. The use of Center Loss promotes compactness within the same machine-type features. Furthermore, the features of different machine types are separated into several small clusters, demonstrating that the model learning has achieved the desired effect of the mixed loss function.

Fig.3 shows the embedding distributions of the slider and ToyTrain of the DCASE23 dataset, on which anomaly detectors are employed, with the anomalous data marked in black. For the slider, the embeddings of anomalous data are clustered in the corner, contrary to the distributions of normal embeddings. Thus, anomalies can be readily identified. For ToyTrain, anomalous embeddings are scattered and mixed with normal embeddings, causing the back-end anomaly detector to perform poorly in this case.

G. Ablation Study

In order to validate the necessity and effectiveness of multi-task classification, we compare MobileAnoNet with two single-task models on the DCASE23 dataset. The first one is trained only by the machine type label, and the second one is trained only by the working condition labels. As shown in Table IV, it can be observed that our multi-task approach outperforms the model trained only by the machine type label, indicating the improvement in model performance due to the inclusion of operating condition classification.

V. Conclusion

This paper proposes a multi-label classification-based anomaly detection method, MobileAnoNet, which transforms the unsupervised anomaly detection task into a supervised proxy task. By utilizing readily available labels, the model undergoes multi-classification training, allowing the feature extractor to learn deep-level features from the input. The back-end detector, leveraging algorithms such as KNN, can provide anomaly scores with low time complexity, promoting computational efficiency. The proposed approach has been tested on

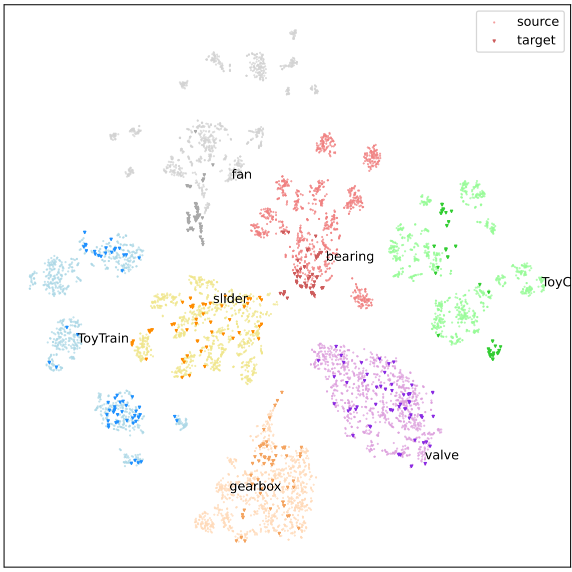
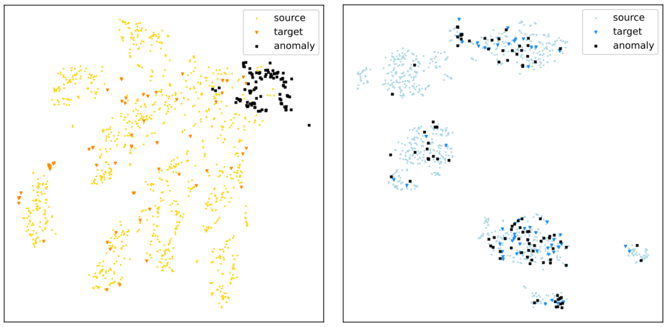


Fig. 2: Embeddings of DCASE23 clips.



(a) (b)

Fig. 3: Embeddings of slider (a) and ToyTrain (b)

two DCASE datasets and achieved significant improvements over the baselines provided by DCASE organizers in 2023. The trained network and anomaly detector can be deployed on mobile devices for distributed anomaly detection in AIoT systems.

References

[1] Kota Dohi, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, Takashi Endo, Masaaki Yamamoto, Yuki Nikaido, and Yohei Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," *In arXiv e-prints: 2205.13879*, 2022.

[2] 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 the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 1–5.

[3] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14318–14328.

[4] Anbai Jiang, Wei-Qiang Zhang, Yufeng Deng, Pingyi Fan, and Jia Liu, "Unsupervised anomaly detection and localization of machine audio: A gan-based approach," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[5] Jose A Lopez, Georg Stemmer, Paulo Lopez-Meyer, Pradyumna Singh, Juan A del Hoyo Ontiveros, and Héctor A Cordourier, "Ensemble of complementary anomaly detectors under domain shifted conditions.," in *DCASE*, 2021, pp. 11–15.

[6] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," *arXiv preprint arXiv:2110.02642*, 2021.

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

[8] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[9] David MJ Tax and Robert PW Duin, "Support vector data description," *Machine learning*, vol. 54, pp. 45–66, 2004.

[10] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.

[11] Qiuhan Meng and Songye Zhu, "Anomaly detection for construction vibration signals using unsupervised deep learning and cloud computing," *Advanced Engineering Informatics*, vol. 55, pp. 101907, 2023.

[12] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

[13] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. Springer, 2016, pp. 499–515.

[14] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[15] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Chinese Conference on Biometric Recognition*. Springer, 2018, pp. 428–438.

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[17] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[18] 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 the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022, pp. 1–5.

[19] Feiyang Xiao, Youde Liu, Yuming Wei, Jian Guan, Qiaoxi Zhu, Tieran Zheng, and Jiqing Han, "The dcase2022 challenge task 2 system: Anomalous sound detection with self-supervised attribute classification and gmm-based clustering," Tech. Rep., DCASE2022 Challenge, July 2022.

[20] Noboru Harada, Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, and Masahiro Yasuda, "First-shot anomaly detection for machine condition monitoring: A domain generalization baseline," *In arXiv e-prints: 2303.00455*, 2023.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.