ENSEMBLE OF COMPLEMENTARY ANOMALY DETECTORS UNDER DOMAIN SHIFTED CONDITIONS

Technical Report

Jose A. Lopez, Georg Stemmer, Paulo Lopez-Meyer, Pradyumna S. Singh
Juan Del Hoyo Ontiveros, Hector Courdourier

Intel Corporation
{jose.a.lopez, georg.stemmer, paulo.lopez.meyer, pradyumna.s.singh}@intel.com
{juan.antonio.del.hoyo.ontiveros, hector.a.cordourier.maruri}@intel.com

ABSTRACT

We present our submission to the DCASE2021 Challenge Task 2, which aims to promote research in anomalous sound detection. We found that blending the predictions of various anomaly detectors, rather than relying on well-known domain adaptation techniques alone, gave us the best performance under domain shifted conditions. Our submission is composed of two self-supervised classifier models, a probabilistic model we call NF-CDEE, and an ensemble of the three.

***Index Terms*—** DCASE, anomaly detection, domain shift, machine condition monitoring, machine health monitoring.

1. INTRODUCTION

The DCASE2021 Challenge Task 2 is concerned with identifying anomalous behavior from a target machine using sound recordings [1]. A major difference between this task and other DCASE tasks is that it is not supervised. Accordingly, the available training data only contains samples from the normal-state distributions. A further complication added to this challenge is that the acoustic characteristics of the training data and of the test data are different – this condition is known as domain shift and there are some known results for reducing the performance gap between the training and test data [2, 3, 4, 5, 6, 7, 8]. In our experiments, while we recognize the potential of these techniques, we did not generally gain much from using these methods alone.

In our submission, we used two self-supervised classifiers that classified the section IDs similar to the approach several teams followed in DCASE2020 [9, 10, 11, 12, 13]. For a third model, we introduce a model that relies on several normalizing flows to estimate the conditional density of input Mel spectrogram sections and use their combined outputs to produce an anomaly score [14, 15, 16, 17, 18, 19, 20, 21, 22].

In the sequel we describe each model, how it was trained, its hyperparameters, and their respective results. In order to put the results into perspective, we include the baseline scores on Tables 1 and 2. The data used in this challenge is 16 KHz, single-channel, audio. For more details, please see [1, 23, 24].

	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
h-mean AUC	0.6249	0.6171	0.6324	0.6597	0.6192	0.6674	0.5341
h-mean pAUC	0.5236	0.5381	0.5338	0.5276	0.5441	0.5594	0.5054

Table 1: Baseline Autoencoder Scores

	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
h-mean AUC	0.5604	0.5746	0.6156	0.6670	0.6189	0.5926	0.5651
h-mean pAUC	0.5637	0.5161	0.6302	0.5916	0.5737	0.5600	0.5264

Table 2: Baseline MobileNetV2 Scores

2. ARCHITECTURES

The first model described below builds on the work from [9]. In particular, the encoder network has been updated to use 1D convolutions rather than 2D as in [9]. The input to this model is a spectrogram with or without a Mel transformation. The second model builds on the well-known WaveNet architecture [25] by adding an x-vector [26] classification head after the dilated convolutions – in a sense, the WaveNet functions as a time-series encoder for the x-vector component. Both models are trained to reduce the cross-entropy loss between predictions and the section IDs. The third model differs from the first two models in that it is completely unsupervised and attempts to learn several distributions of some Mel spectrogram bins conditioned on the remaining bins. We call these approaches complementary because of the different input modalities and learning approaches. The last system provided is an ensemble of the three.

All our development was done using PyTorch [27] and spectrograms were computed using nnAudio [28]. The third model additionally used the Pyro [29] probabilistic programming library.

2.1. XVector1D

A high-level view of the architecture of the first model is shown in Table 3. We denote additive margin softmax as AMS [30].

We use the term "standardizer" as a preprocessing step done before passing data to the rest of the network. In most cases, this is simply a batch-norm layer with the learnable parameters disabled. In this way, this batch-norm will perform the usual frequency-wise normalization once the running statistics have converged. However, for gearbox and ToyCar we used an AutoDIAL layer [4] instead.

ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
STFT	MEL	STFT	MEL	STFT	STFT	STFT
standardizer	standardizer	standardizer	standardizer	standardizer	standardizer	standardizer
encoder	encoder	encoder	encoder	encoder	encoder	encoder
x-vector	x-vector	x-vector	x-vector	x-vector	x-vector	x-vector
AMS	AMS	AMS	AMS	AMS	AMS	AMS

Table 3: XVector1D High-level Architectures

ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
AutoDIAL	batch-norm	AutoDIAL	AutoDIAL	AutoDIAL	AutoDIAL	AutoDIAL
C(128,192)	C(128,192)	C(128,192)	C(128,192)	C(128,192)	C(128,192)	C(128,192)
5 x C(192,192)	5 x C(192,192)	5 x C(192,192)	4 x C(192,192)	5 x C(192,192)	5 x C(192,192)	5 x C(192,192)
			C(192,90)			

Table 4: Encoder Parameters

The encoder used in this model uses 1D convolutions with kernel size 3 and leaky-relu activations. The number of layers varied with machine as shown on Table 4 – in this table, we use "C" to mean 1D convolution.

The x-vector component used here remains largely the same as in [9] except the interface to the encoder had to be adapted as expected to accept the 1D encoder output.

2.1.1. Preprocessing

This model did not use any special preprocessing or augmentation. The logarithm was taken for both the STFT and the Mel spectrograms. All spectrograms were computed with frequency min and max values set to 100 and 8000 Hertz, respectively.

2.1.2. Training & Results

The model was trained to predict the section ID meta-data parameter using the categorical cross entropy loss function. We found that the spectrogram parameters had a big effect on the performance. Parameters like the number of input samples, the number of points used for the FFT, the hop length can have a significant effect. We generally used the AdamW optimizer with the default learning rate of 1×10^{-3} and weight decay set to 1×10^{-4}. However, we used ASGD with the default learning rate (and no weight decay) for gearbox. Generally, the training losses converge more slowly using ASGD but sometimes the slower trajectory spends more epochs close to an optimal region with respect to AUC and this can yield better results. The training was usually run for 300 epochs, using all the training data from the development and evaluation datasets. Lastly, we computed the average embedding, during training, using the embedding from the layer prior to the final AMS classification layer. At test time, the average embedding was used to compute the cosine and Mahalanobis distances to the test embedding which served as additional options for anomaly scores. Table 5 shows the results.

	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
batch size	128	64	128	64	128	128	64
input samples	16384	16384	16384	98000	16384	16384	98000
no. Mels	2048	128	2048	128	2048	2048	2048
no. FFT	4096	1024	4096	1024	4096	4096	4096
hop	80	512	512	80	512	512	512
scoring	cosine	mahalanobis	softmax	mahalanobis	softmax	softmax	softmax
h-mean AUC	0.6702	0.7193	0.7171	0.8342	0.7799	0.7871	0.9032
h-mean pAUC	0.6233	0.6772	0.7295	0.7443	0.6684	0.6728	0.7724

Table 5: XVector1D Scoring Results

2.2. WaveNet-XVector

We explored the use of a WaveNet model processing the audio samples directly. For details on the architecture we refer the reader to the original publication [25]. In the original paper the authors explain that the model can be readily adapted to classification tasks and in their classification experiment they add a mean pooling layer after the dilated convolutions followed by "a few non-causal convolutions". The training proceeds with two loss terms: one for predicting the next sample and the other is the classification loss. We follow this procedure in that we use a mean pooling layer (with kernel size 10) and train with the two loss functions but instead of using a few convolutions, we use an x-vector component, with AMS top layer, as with the XVector1D model. In this way, one can consider this model a variant of the XVector1D model which uses an audio-only encoder.

	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
input proc.	batch-norm	batch-norm	batch-norm	batch-norm	batch-norm	batch-norm	AutoDIAL
blocks	1	1	1	1	1	1	1
layers	14	14	14	14	14	14	14
dilation ch.	32	32	32	64	64	32	32
residual ch.	32	32	32	64	64	32	32
skip ch.	32	32	32	64	64	32	32

Table 6: WaveNet Parameters

2.2.1. Preprocessing

For valve and ToyTrain we used the Teager-Kaiser energy operator to preprocess the audio [31, 32, 33, 34]. The motivation was that, because the valve noises are sparse and impulsive events, the noise suppression provided by the Teager-Kaiser operator would improve the signal-to-noise ratio in the valve recordings. Despite improving the results for valve and ToyTrain, the improvement was modest.

2.2.2. Training & Results

To train this model, we used the Adamax optimizer with the default learning rate for 200 epochs. Table 7 shows the performance of this model.

	ToyCar	ToyTrain	fan	gearbox	pump	slider	valve
batch size	128	128	128	64	64	128	128
input samples	16384	16384	16384	16384	16384	16384	16384
scoring	softmax	softmax	softmax	softmax	softmax	softmax	softmax
h-mean AUC	0.5843	0.6641	0.8122	0.7156	0.7543	0.7184	0.7297
h-mean pAUC	0.5629	0.5696	0.8025	0.5964	0.6506	0.6239	0.6206

Table 7: WaveNet-XVector Scoring Results

2.3. NF-CDEE

For our third system, we attempt to model the probability density function of the Mel spectrograms of the machine sounds, for a single machine, using normalizing flows. We used the Pyro [29] probabilistic programming library to develop this model. We found that training a model to fit a distribution with the same dimensions as Mel bins to be somewhat unstable. In order to improve the stability we instead estimate several conditional densities and trained them in a single model, minimizing the sum of their negative log-likelihoods. We consider this model an ensemble of conditional density anomaly detectors. Hence, we call this model NF-CDEE, because it uses normalizing flows and it is a conditional density estimator ensemble. Each conditional density estimator fits the distribution of a n-bin segment of input spectrograms conditioned on the remaining bins. This reduces the instability due to dimensionality. The parameter n and the amount of overlap are tunable by the user. For this work, we chose $n = 32$ with no overlap. Each normalizing flow uses a single conditional spline with 16 count-bins and the default hidden layer dimensions – these are also tunable but in our experiments they did not significantly affect the performance.

To summarize, each estimator outputs the probability $p(s_A|s_{A^c})$ where s is a vector of dimension equal to the number

of Mel bins m that is indexed by the set $\mathcal{I} = \{1, \ldots, m\}$. A is an n-element subset of $\mathcal{I}$, and A^c is its complement $\mathcal{I} - A$. We define the likelihood of the normal state as:

$$p(\text{normal}) = \prod_i p(s_{A_i} | s_{A_i^c}) \quad (1)$$

where $i \in [1, \ldots, k]$ and k is a positive integer provided by the user – it is the number of estimators in the ensemble. To train the model, we minimize the negative logarithm of $p(\text{normal})$. Therefore, the output of NF-CDEE is the sum of the individual negative log-likelihoods.

2.3.1. Training & Results

To train this model we converted the input audio to 256-bin Mel spectrograms, computed using 8192-point FFTs with hop-length 512, and applied frequency-wise normalization before passing to the conditional density estimators. Each model was trained with all the sections of the development (or evaluation) training data, per machine type – except for fan for which we trained a model for each section. To further reduce training instability, caused by the normalizing flow determinant computation, we take the mean across the time dimension. This last step was important for stabilizing the training of the ensemble. As previously stated the loss function used was the sum of the negative log-likelihoods and this also served as the anomaly score.

For the optimizer, we used the same optimizer as the XVector1D, with gradient clipping. In our experiments this model generally needs to train for about 50 epochs. Table 8 shows the results.

	ToyCar	**ToyTrain**	**fan**	**gearbox**	**pump**	**slider**	**valve**
batch size	32	32	32	32	32	32	32
input frames	192	192	192	192	192	192	192
m	256	256	256	256	256	256	256
n	32	32	32	32	32	32	32
k	8	8	8	8	8	8	8
scoring	NLL	NLL	NLL	NLL	NLL	NLL	NLL
h-mean AUC	0.8657	0.7797	0.7866	0.8081	0.6993	0.7483	0.6130
h-mean pAUC	0.7831	0.6031	0.6024	0.6513	0.5655	0.6054	0.5275

Table 8: NF-CDEE Scoring Results

2.4. Ensemble

For the last system we combined the three models by first standardizing the training data scores and then searching over a grid of convex combinations, similar to [35]. Table 9 shows the results.

	ToyCar	**ToyTrain**	**fan**	**gearbox**	**pump**	**slider**	**valve**
WaveNet weight	0.03	0.03	1.0	0.04	0.32	0.02	0
XVector1D weight	0.06	0.55	0	0.61	0.68	0.52	1
NF-CDEE weight	0.91	0.42	0	0.35	0	0.46	0
h-mean AUC	0.8745	0.7756	0.8122	0.8613	0.7958	0.8287	0.9032
h-mean pAUC	0.7837	0.7048	0.8025	0.7635	0.6790	0.6925	0.7724

Table 9: Ensemble Scoring Results

3. CONCLUSIONS

We have outlined our submission to the DCASE2021 Challenge Task 2, which featured a domain shift between the training and test distributions. We found it concerning that domain adaptation methods that seem to do well for other modalities, especially vision, do not seem to work as well for audio (at least in our implementations). This discrepancy gives the DCASE2021 Challenge a greater relevance, because it highlights the need for the audio community to generate more effective domain adaptation methods for audio.

Of the models we developed, we find NF-CDEE to be particularly promising because it is unsupervised. In real-world settings it is not always practical to leverage meta-data, even when it is possible to do so. Moreover, expect the ensembling nature of the model to perform better in domain shift situations. Going forward we plan to further develop this model.

4. 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," *arXiv preprint arXiv:2106.04492*, 2021.

[2] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," 2020.

[3] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognition*, vol. 80, pp. 109–117, 2018.

[4] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Autodial: Automatic domain alignment layers," 2017.

[5] ——, "Just dial: Domain alignment layers for unsupervised domain adaptation," 2017.

[6] M. Mancini, L. Porzi, S. R. Bulò, B. Caputo, and E. Ricci, "Boosting domain adaptation by discovering latent domains," 2018.

[7] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," 2018.

[8] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," 2015.

[9] J. A. Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer, and J. Huang, "A speaker recognition approach to anomaly detection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 96–99.

[10] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 46–50.

[11] T. Inoue, P. Vinayavekhin, S. Morikuni, S. Wang, T. Hoang Trong, D. Wood, M. Tatsubori, and R. Tachibana, "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 66–70.

[12] P. Primus, V. Haunschmid, P. Praher, and G. Widmer, "Anomalous sound detection as a simple binary classification problem with careful selection of proxy outlier examples,"

in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 170–174.

[13] Q. Zhou, "Arcface based sound mobilenets for dcase 2020 task 2," DCASE2020 Challenge, Tech. Rep., July 2020.

[14] E. G. Tabak and C. V. Turner, "A family of nonparametric density estimation algorithms," *Communications on Pure and Applied Mathematics*, vol. 66, no. 2, pp. 145–164.

[15] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," 2016.

[16] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2020.

[17] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," 2021.

[18] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," 2019.

[19] H. M. Dolatabadi, S. Erfani, and C. Leckie, "Invertible generative modeling using linear rational splines," 2020.

[20] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," 2015.

[21] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," 2017.

[22] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," 2016.

[23] 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.

[24] 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.

[25] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[28] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolution neural networks," 2020.

[29] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *J. Mach. Learn. Res.*, vol. 20, pp. 28:1–28:6, 2019.

[30] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, p. 926–930, Jul 2018.

[31] P. Maragos, J. Kaiser, and T. Quatieri, "Energy separation in signal modulations with application to speech analysis," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 3024–3051, 1993.

[32] P. Maragos, J. F. Kaiser, and T. F. Quatieri, "On amplitude and frequency demodulation using energy operators," *IEEE Transactions on Signal Processing*, vol. 41, no. 4, pp. 1532–1550, Apr. 1993.

[33] A. Georgogiannis and V. Digalakis, "Speech emotion recognition using non-linear teager energy based features in noisy environments," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2045–2049.

[34] H. Li, H. Zheng, and L. Tang, "Gear fault detection based on teager-huang transform," *International Journal of Rotating Machinery*, vol. 2010, pp. 1–9, 2010.

[35] P. Daniluk, M. Gozdziewski, S. Kapka, and M. Kosmider, "Ensemble of auto-encoder based systems for anomaly detection," DCASE2020 Challenge, Tech. Rep., July 2020.