A SPEAKER RECOGNITION APPROACH TO ANOMALY DETECTION

Jose A. Lopez [1]*, Hong Lu* [1]*, Paulo Lopez-Meyer* [2]*, Lama Nachman* [1]*, Georg Stemmer*[3]*, Jonathan Huang*[4]

[1] Intel Corp, Intel Labs, 2200 Mission College Blvd., Santa Clara, CA 95054, USA,
[2] Intel Corp, Intel Labs, Av. Del Bosque 1001, Zapopan, JAL, 45019, Mexico,
[3] Intel Corp, Intel Labs, Lilienthalstraße 15, 85579, Neubiberg, Germany,
[4] Work done at Intel, 2200 Mission College Blvd., Santa Clara, CA 95054, USA,
{jose.a.lopez, hong.lu, lama.nachman}@intel.com
{paulo.lopez.meyer, georg.stemmer}@intel.com
{jonathan.huang}@ieee.org

ABSTRACT

We present our submission to the DCASE 2020 Challenge Task 2, which aims to promote research in anomalous sound detection. We found that a speaker recognition approach enables the use of all the training data, even from different machine types, to detect anomalies in specific machines. Using this approach, we obtained good results for 5 out of 6 machines on the development data. We also discuss the modifications needed to surpass the baseline score for the remaining (ToyConveyor) machine which we found to be particularly difficult. On the challenge evaluation test data, our results were skewed by the system's uninspiring performance on the Toy machines. However, we placed 18th in the challenge due to our results on the industrial machine data where we reached the top 5 in team pAUC scores.

***Index Terms*—** DCASE, anomaly detection, anomalous sounds, machine condition monitoring, machine health monitoring, speaker recognition.

1. INTRODUCTION

The DCASE 2020 Challenge Task 2 is concerned with promoting research into identifying anomalous behavior from a target machine using sound recordings [1]. The benefits of detecting anomalies in machines early are well understood and include: reduced downtime, improved efficiency, and useful life extension. Furthermore, the cost of audio sensors is relatively low which makes them an attractive choice for widespread adoption. However, it is generally difficult to realize these benefits while maintaining a low false alarm rate. This is largely due to the lack of data to learn from and a lack of negative training examples in particular.

A major difference between this task and other DCASE challenge tasks is that it is not supervised. Accordingly, the available training data only contains samples from the normal state distributions. The data for this challenge are derived from two existing datasets [2, 3] and was provided in stages.

Among approaches for unsupervised anomaly detection using deep neural networks (DNN), autoencoder based approaches stand out and are frequently encountered in the literature [4, 5, 6, 7, 8]. The idea behind using an autoencoder as an anomaly detector is straightforward and compelling[1]: one trains an encoder network to learn a lower dimensional representation of the normal state data, one trains a decoder to reconstruct the normal inputs from the encoded representation, and if all goes well one can effectively use the reconstruction error as an anomaly score. The autoencoder is expected to reconstruct normal data better than anomalies. However, this is not guaranteed and in practice an autoencoder may do a good job of reconstructing anomalies as well [9]. This is the problem we encountered in our early experiments with autoencoders and why we ended-up looking in a different direction.

In our approach, we used all the training data provided in the "development" and "additional" datasets [10, 11]. Although these data do not contain abnormal samples, they do contain other information – they contain the machine type and ID number. We leveraged this information to train a neural network to predict the machine ID of an input sample, and we leveraged the machine type in the normalization step discussed in Section 2.3.

The DNN architecture used here is composed of a (Mel or STFT) spectrogram layer, followed by a 2D CNN encoder, followed (optionally) by a variant of the x-vector model used for speaker recognition [12], and capped with either a fully connected (FC) or added margin softmax (AMS) layer also found in the speaker recognition literature [13]. We employ two scoring methods and take the best one for a given machine type.

A high level description of the architectures can be found in Table 1.

Fan	**Pump**	**Slider**	**Valve**	**ToyCar**	**ToyConv**
STFT	MEL	MEL	MEL	MEL	MEL
encoder	encoder	encoder	encoder	encoder	encoder
x-vector	x-vector	x-vector	AMS	x-vector	FC
AMS	AMS	AMS		AMS	

Table 1: High Level Architectures

2. METHODOLOGY

In this section we detail our implementation, DNN training strategy, and scoring methods. All the code was written in Python and the models were developed using PyTorch [14].

[1] Indeed the baseline model provided in this challenge is autoencoder based.

2.1. Data Processing

The DCASE 2020 Task 2 dataset consists of 10-second audio files that include the sound of the target machine and environmental noise. There are six types of machine categories. ToyCar and ToyConveyor are from the ToyADMOS dataset [2]. Valve, Pump, Fan, and Slider are from the MIMII dataset [3]. Within each machine category there are a number of machine IDs, for a total of 41 possible sound categories. The interested reader is referred to the dataset references for details on the recording procedures. However, all the audio files contain a single-channel and use a 16kHz sampling rate.

For spectral features, we used the Python package, nnAudio [15], to transform the input audio into either a Mel or STFT spectrogram in the so called "stand alone" mode. That is, we did not use the trainable kernel options. The Mel spectrograms were generated using the HTK option. The optimal spectrogram settings varied with machine type. We provide these settings in Section 3.

2.2. Network Architectures

All architectures utilized the 2D CNN encoder shown in Figure 1. The encoder utilizes progressively smaller kernel sizes and ends with a max pool layer. For Fan, Pump, Slider, and ToyCar we followed the encoder with a variant of the x-vector model from [12]. The x-vector model uses several 1D convolutions, stats pooling [16], and FC layers to produce embeddings that can be fed to a classifier. For these machines, including the x-vector component increased the sum of AUC and pAUC by 5 to 7 percent. Figure 2 shows the variant used here[2]. For all machine types, except ToyConveyor, we capped the embedding layers with an additive margin softmax layer [13] which has the effect of increasing inter-machine distance in the embedding space for improved accuracy. See [13, 17] for details. For ToyConveyor, we simply followed the encoder with an FC classifier. Finally, in all models we utilized ReLU activations or one of its variants.

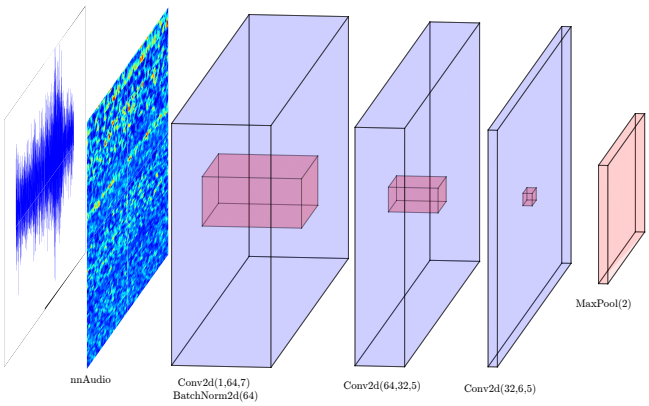

Figure 1: Spectrogram layer and convolutional encoder model

2.3. Training Strategy

All the models use the same training strategy, except for the ToyConveyor model which we discuss separately. At training time, a

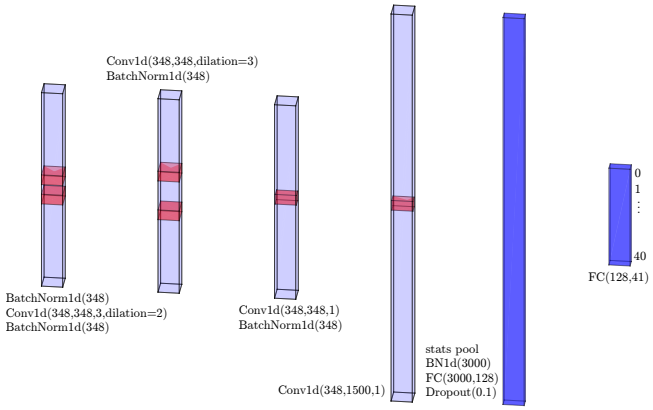

Figure 2: X-vector model with AMS top layer

contiguous $\frac{10}{7}$-second clip was randomly sampled from the training files. We used this method primarily because it was a way of increasing the batch size given the memory constraints of our hardware, but also because early experiments performed better this way. Batches of 64-128 such samples were used during each epoch, for between 100 and 200 epochs to ensure all the data are sampled.

At the output of the spectrogram layer, before inputting to the encoder, we normalized the spectrogram by subtracting the column-wise mean, and dividing by the column wise standard deviation, of all the training spectrograms of the same machine type.

We used a different strategy to obtain the best result from the ToyConveyor data. The strategy was motivated by the fact that the ToyConveyor data were too easily differentiated from other machines, even of the same type. We divided each 10-second sample into 7 parts and used the 7 parts as a batch because we obtained better results using smaller batches. Thus, each batch was composed of data from the same ToyConveyor machine ID[3]. With a probability of $\frac{1}{2}$, we simulated anomalies by corrupting 1 of the 7 parts by linearly combining the part spectrogram with a spectrogram from another ToyConveyor machine ID according to (1).

$$S_i = \lambda S_j + (1 - \lambda) S_i \tag{1}$$

where S_i is the spectrogram of ToyConveyor ID i and $i \neq j$. In contrast to the so called *mix-up* data augmentation method [18], we did not randomly select λ – we fixed λ to 0.03 and randomly selected the machine ID j. Selecting a larger λ resulted in the model (too) easily identifying the anomaly, leading to overfitting. By using this training strategy, we increased the sum of AUC and pAUC by approximately 18 percent.

For all models, we used a categorical cross-entropy loss function with l_1 regularization on the encoder weights, to prevent overfitting, and the Adamax optimizer with the default learning rate.

2.4. Scoring

We used two scoring methods. The first is simply 1 minus the softmax probability of the specific machine ID. Clearly, if the model is certain a sample belongs to machine ID i, the ith output will be close to 1, resulting in a lower anomaly score. Conversely, as the

[2]A 1D x-vector model alone can obtain good classification performance, but in our experiments it was not as good as the 2D encoder with a classifier.

[3]In contrast to other models, the output of the ToyConveyor anomaly detector only has 7 classes, one for each ToyConv. ID and 1 *Other* class.

uncertainty increases, so will the anomaly score. In the ToyConveyor case, since the model output categories also include an Other class, we add this softmax probability to the anomaly score as well.

The second scoring method is the cosine distance between the average normal embedding[4], recorded at training time, and the embedding of the test sample computed at test time. Generally, the two scoring methods produced similar scoring results. However, we selected the scoring method that produced the largest sum of AUC and pAUC.

3. RESULTS

We summarize the results of our development work in Table 2. Using our approach, the ToyConveyor case proved the most difficult, followed by Fan. The Slider and Valve machines were the easiest to obtain good results for, followed by ToyCar and Pump. The rankings given in the tables below are provided on a per-team basis. That is, if a team provided 5 entries we compare against their best score as reported on [19].

	Fan	**Pump**	**Slider**	**Valve**	**ToyCar**	**ToyConv**
batch size	64	64	64	128	64	7
no. Mels	128	256	128	128	128	128
no. FFT	1024	1024	1024	1024	1024	1024
hop	512	80	80	512	80	512
fmin	1	100	10	0	10	0
fmax	4000	7700	7700	8000	4000	4000
scoring	cos dist.	cos dist.	cos dist.	softmax	softmax	softmax
AUC	0.8823	0.9321	0.9997	0.9989	0.9573	0.7417
pAUC	0.8057	0.8619	0.9982	0.9941	0.9032	0.6586
AUC rank	5	4	2	1	3	23
pAUC rank	4	4	2	1	3	7

Table 2: Scoring Results On Development Data

For the reference we also list the average baseline results, provided in [1], on Table 3.

	Fan	**Pump**	**Slider**	**Valve**	**ToyCar**	**ToyConv**
AUC	0.6583	0.7289	0.8476	0.6629	0.7877	0.7253
pAUC	0.5245	0.5999	0.6653	0.5098	0.6758	0.6043

Table 3: Average Baseline Scoring Results

Table 4 summarizes our results on the challenge evaluation test data [19]. These results were ranked 18th out of 117 entries.

	Fan	**Pump**	**Slider**	**Valve**	**ToyCar**	**ToyConv**
AUC	0.9303	0.9398	0.9888	0.9680	0.8659	0.7121
pAUC	0.9067	0.9072	0.9538	0.9061	0.8185	0.6141
AUC rank	9	5	2	4	12	30
pAUC rank	5	3	2	4	11	30

Table 4: Scoring Results On Evaluation Data

For reference, we also provide the top result for each machine, as listed in [19], on Table 5. Notice, for a given machine the top AUC and pAUC scores may come from different challenge entries. Please see [19] for details.

[4]The embedding is the 128D output of the x-vector component.

	Fan	**Pump**	**Slider**	**Valve**	**ToyCar**	**ToyConv**
AUC	0.9979	0.9776	0.9984	0.9782	0.9560	0.9262
pAUC	0.9892	0.9260	0.9917	0.9493	0.9130	0.8056

Table 5: DCASE Best Scores

4. CONCLUSIONS

We have outlined our speaker recognition approach to the DCASE 2020 Challenge Task 2 which uses the machine IDs themselves to make an unsupervised problem supervised. In an industrial setting, it is common to have many machine types and instances. Therefore, we believe that using machine ID and machine type information this way does not significantly decrease the applicability of the method. In the case of a single machine type and instance, it may still be possible to apply the method by using synthetic anomalies or recordings from public datasets.

It is our intuition that this approach succeeded because the spectral content of these machines is sufficiently similar to make the task of separating the machine IDs challenging. This was less so for the Toy classes, especially for ToyConveyor. For example, in one experiment we classified the ToyConveyor IDs and used data from the remaining machines as Other, during training both the training and validation accuracies quickly exceeded 99%. This led us to look to data augmentation methods for ToyConveyor, as described in Section 2.3. Consequently, we expect that this approach may not work in cases where the spectral content of the machine sounds differs greatly.

5. FUTURE WORK

Our work deserves further development. Obviously, the performance of our proposed model architectures on some machine types has room for improvement. We feel that there are improvements to be gained by: tuning the encoder hyper-parameters, substituting the encoder for a pre-trained model, using an attention mechanism, or by fusing several model predictions. We also believe it is important to identify when our approach is expected to work better than autoencoders and otherwise characterize the augmentation samples, i.e. synthetic anomalies, that are useful.

6. REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *arXiv e-prints: 2006.05822*, June 2020, pp. 1–4. [Online]. Available: https://arxiv.org/abs/2006.05822

[2] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, November 2019, pp. 308–312. [Online]. Available: https://ieeexplore.ieee.org/document/8937164

[3] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII Dataset: Sound dataset for

malfunctioning industrial machine investigation and inspection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, November 2019, pp. 209–213. [Online]. Available: http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Purohit_21.pdf

[4] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *CoRR*, vol. abs/1901.03407, 2019. [Online]. Available: http://arxiv.org/abs/1901.03407

[5] D. Y. Oh and I. D. Yun, "Residual error based anomaly detection using auto-encoder in smd machine sound," *Sensors*, vol. 18, no. 5, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/5/1308

[6] E. Principi, F. Vesperini, S. Squartini, and F. Piazza, "Acoustic novelty detection with adversarial autoencoders," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3324–3330.

[7] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," *CoRR*, vol. abs/1701.01546, 2017. [Online]. Available: http://arxiv.org/abs/1701.01546

[8] E. Marchi, F. Vesperini, S. Squartini, and B. Schuller, "Deep recurrent neural network-based autoencoders for acoustic novelty detection," *Computational Intelligence and Neuroscience*, vol. 2017, 09 2016.

[9] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, p. 212–224, Jan 2019. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2018.2877258

[10] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Dcase 2020 challenge task 2 development dataset," Mar. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3678171

[11] ——, "DCASE 2020 Challenge Task 2 Additional Training Dataset," Apr. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3727685

[12] 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.

[13] J. Huang and T. Bocklet, "Intel Far-Field Speaker Recognition System for VOiCES Challenge 2019," in *Proc. Interspeech 2019*, 2019, pp. 2473–2477. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2894

[14] 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.

[15] K. W. Cheuk, H. H. Anderson, K. Agres, and D. Herremans, "nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolution neural networks," *ArXiv*, vol. abs/1912.12055, 2019.

[16] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech 2017*, 2017, pp. 999–1003. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-620

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

[18] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *CoRR*, vol. abs/1710.09412, 2017. [Online]. Available: http://arxiv.org/abs/1710.09412

[19] Unsupervised detection of anomalous sounds for machine condition monitoring - challenge results. [Online]. Available: http://dcase.community/challenge2020/task-unsupervised-detection-of-anomalous-sounds-results

[20] W. Lu, Y. Cheng, C. Xiao, S. Chang, S. Huang, B. Liang, and T. Huang, "Unsupervised sequential outlier detection with deep architectures," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4321–4330, 2017.

[21] E. Rushe and B. M. Namee, "Anomaly detection in raw audio using deep autoregressive networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3597–3601.