REFRAMING UNSUPERVISED MACHINE CONDITION MONITORING AS A SUPERVISED CLASSIFICATION TASK WITH OUTLIER-EXPOSED CLASSIFIERS

Technical Report

Paul Primus

Institute of Computational Perception (CP-JKU)
Johannes Kepler University, Austria

ABSTRACT

This technical report contains a detailed summary of our submissions to the *Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring* (MCM) Task of the *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events 2020* (DCASE). The goal of acoustic MCM is to identify whether a sound emitted from a machine is normal or anomalous. In contrast to the task coordinator's conjecture that 'this task cannot be solved as a simple classification problem,' we show that a simple binary classifier substantially outperforms the provided unsupervised Autoencoder baseline across all machine types and instances, if *outliers* i.e., various other recordings, are available. In addition to this technical description, we release our complete source code to make our submission fully reproducible[1].

***Index Terms*—** Unsupervised Anomaly Detection, Outlier-Exposed Classifiers, Machine Condition Monitoring, DCASE2020

1. INTRODUCTION

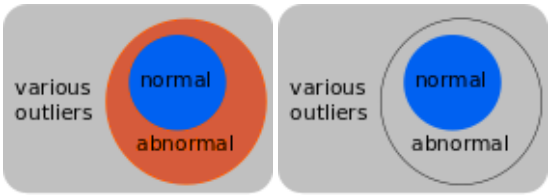
The aim of acoustic Machine Condition Monitoring (MCM) is to detect sounds which deviate from what is considered 'normal' for a specific machine or a class of machines and utilize this information for various downstream tasks, such as failure detection or predictive maintenance.

Anomaly detection methods, in general, can be broadly divided into supervised and unsupervised methods. In the first setting, both normal and abnormal samples are available and labeled; the learning task is to fit a classifier (Fig. 1a). Unfortunately, due to the variety and rare nature of anomalies, it is often hard to define and collect anomalies in practice. In the second setting, only normal recordings are available for learning (Fig 1b). Therefore, the learning objective turns into creating a model of what is regarded as normal (e.g., a density model), which is then used to assign anomaly scores to samples (e.g., based on log-likelihoods).

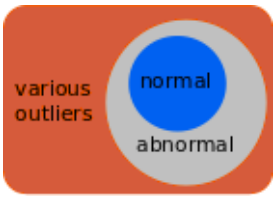
In this technical report, we distinguish between three types of data: *normal data*, which are all possible sounds emitted from a machine in a normal operation state; *abnormal data*, which comprises all sounds emitted from a machine in a non-normal state; and *outliers*, which are all other possible sounds in the audio domain excluding the two previous categories (Fig. 1).

Recent work in unsupervised [1] and supervised [2] anomaly detection leverages large amounts of unlabeled outlier data, which is commonly referred to as *outlier exposure*. Along the same line, we call classifiers trained with outlier data *outlier-exposed classifiers*.

In the following sections we are going to motivate outlier-exposed classifiers for machine condition monitoring, present our results and compare them to the provided Autoencoder baseline [3].



(a) Supervised Setting (b) Unsupervised Setting



(c) Outlier-Exposed Setting

Figure 1: Comparison between supervised (1a), unsupervised (1b), and outlier-exposed anomaly detection settings (1c). Blue area represents the negative, orange the positive class. Samples in the grey area are not available/ used. Although the classical supervised setting would be optimal, real anomalies are rare and/ or expensive to collect.

2. OUTLIER-EXPOSED CLASSIFIERS

If both normal and abnormal samples are available for training, anomaly detection can be considered a classification task, where normal samples belong to the negative class, and abnormal samples belong to the positive class (Fig. 1a). However, the lack of anomalous samples generally prevents us from modeling anomaly detection in this supervised framework.

To be able to still treat anomaly detection as a classification task, we extend the definition of the positive class to include various other samples, i.e., samples that are neither normal nor abnormal but still in the same domain (*outliers* for short). If the anomaly detection problem is framed this way, the classifier has to distinguish normal samples from any other possible sample in the same domain. For

[1] https://gitlab.cp.jku.at/paulp/dcase2020_task2

ResNet			
Type	#K	KS 1	KS 2
Conv	$c \cdot 2^0$	5	
BN	-	-	
RB	$c \cdot 2^0$	3	1
Max Pool	-	2	-
RB	$c \cdot 2^0$	3	3
Max Pool	-	2	-
RB	$c \cdot 2^0$	3	a
RB	$c \cdot 2^0$	3	b
Max Pool	-	2	-
RB	$c \cdot 2^1$	1	1
RB	$c \cdot 2^2$	1	1
RB	$c \cdot 2^2$	1	1
Conv	1	1	-
BN	-	-	-
GAP	-	-	-

Residual Block (RB)	
Type	KS
Conv	KS 1
BN	
Conv	KS 2
BN	
Add Input	

Table 1: Model architecture for audio classification by Koutini et al. [4]. #K and KS are the number of kernels and kernel size, respectively. Residual Blocks (RB) consist of two Convolutional (Conv) layers with #K kernels, each followed by a Batch Normalization (BN) layer. GAP is a Global Average Pooling Layer. All nonlinearities are ReLUs. a and b are set to either 1 or 3 to control the receptive filed of the network. c controls the number of convolution filters.

simplicity, we call classifiers trained this way *outlier-exposed classifiers*. Note that, in our experiments, no anomalies are available, and we therefore exclusively train on normal samples and outliers (Fig. 1c).
For acoustic MCM (and arguably many other tasks), outliers are comparably cheap and easy to collect, if not already available in abundance.

3. EXPERIMENTS

We now give a more detailed account of the data sets used, the preprocessing steps, the neural network architecture, the training procedure, and the results compared to the provided baseline [3].

3.1. Experimental Setup

First, it is necessary to clarify the terms 'Machine Type' and 'Machine ID.' The term 'Machine Type' refers to a family of machines, and 'Machine ID' is an identifier for a specific machine instance of a machine type. All our models are trained for a specific machine instance and not per machine type as done for the baseline system. We find the baseline method to perform *slightly better* if trained per machine ID, but reporting these results here would be out of scope.

3.1.1. Dataset

We experiment with the ToyADMOS [5] and MIMII [6] data sets, as provided by the task organizers[2]. The union of both sets comprises of normal/ abnormal recordings for six machine types: fan, pump, slider, valve, toy car, and toy conveyor. For each machine type, three or four sets of recordings taken from one distinct machine instance are given for development and three or four more

[2]http://dcase.community/challenge2020/task-unsupervised-detection-of-anomalous-sounds#audio-dataset

ID	a	b	c	lR	loss
1	3	3	128	10^{-4}	BCE
2	3	3	64	10^{-4}	BCE
3	3	3	64	10^{-4}	AUC
4	3	3	64	10^{-5}	BCE
5	1	1	64	10^{-4}	BCE
6	3	3	64	10^{-5}	ACU
7	3	1	64	10^{-4}	BCE
8	3	1	64	10^{-5}	BCE
9	3	1	64	10^{-4}	AUC
10	1	1	64	10^{-4}	AUC
11	3	1	64	10^{-5}	AUC
12	1	1	64	10^{-5}	AUC
13	1	1	64	10^{-5}	BCE

Table 2: Model Variants. a and b are receptive filed modifiers. c is the number of base channels used. Lr and loss give the learning rate and the loss used, respectively.

for final evaluation. The recordings of each machine instance are split into a training and a testing subset. While the training set only contains normal samples, the test data contains both normal and abnormal samples. Labels of the test data are only known for machine instances in the development set; labels of the test data in the evaluation set are unknown to participants and used to rank the challenge submissions. Since the labeled test sets of the development set must not be used for training, we only use it to select models and report results. For a more detailed description of the data sets we would like to refer the reader to [3], [5], and [6].
To train an outlier-exposed classifier for a specific machine ID, we use the training set associated with the given machine ID as negative samples and the normal sounds of *all other instances and machine types* as positive samples. We get better training results if only samples from those machines which are contained in the same dataset are used, i.e., we do not use samples from ToyADMOS and MIMII at the same time for training. This discrepancy could be attributed to the different recording conditions, which make samples from different data sets trivially distinguishable based on simple statistics such as mean and standard deviation.

3.1.2. Pre-Processing

Preprocessing is done in a similar way as in the baseline system [5]: First, the raw audio is normalized to zero mean and standard deviation one. Then, we re-sample the audio signals to 16000Hz and compute a mono-channel Short Time Fourier Transform using 1024-sample windows and a hop-size of 512 samples. We weight the resulting power spectrogram with a mel-scaled filterbank of 128 filters and apply the logarithm to dampen large outliers.

3.1.3. Network Architecture

We use the model architecture (Table 1) introduced by Koutini et al. [4], a receptive-field-regularized, fully convolutional, residual network (ResNet) [7] tested in various other audio-related classification tasks [8, 9]. We slightly adopt the receptive field by changing filter sizes as described in Table 1 and Table 2.

3.1.4. Training

We train the model on random snippets of 256 frames length to minimize the *Binary Cross Entropy* (BCE) or the *Area Under the Curve* (AUC) loss [10] for 100 epochs using ADAM update rule [11] with $beta_1 = 0.9$ and $beta_2 = 0.99$ and a batch size of 64. Batches are stratified to contain 32 positive and 32 negative samples. The number of base channels, the receptive field modifiers, the loss function, and the initial learning rate for each model are given in Table 2. We decay the learning rate by a factor of 0.99 each epoch.

3.2. Results

The final anomaly score for each test sample is computed by averaging the logit outputs over 256-frame windows with hop size one. The performance of each model variant in terms of AUC and pAUC averaged per machine type compared to the baseline system is given in Figures 2 & 3. Note that all outlier-exposed classifiers outperform the baseline system by a large margin.

4. SUBMISSIONS

To identify the overall best model we rank the model variants according to the procedure described in the task description. Finally, we submit four systems to the challenge:

1. Anomaly scores computed by overall best model variant, i.e., model variant ID 1.
2. Anomaly scores computed by best model variant in terms of average AUC and pAUC per machine type.
3. Anomaly scores median averaged over the five best model variants in terms of average AUC and pAU per machine type.
4. Anomaly scores median averaged over all model variants per machine type.

5. CONCLUSION

The results of our experiments suggest a simple binary classifier, in combination with a set of task-unrelated outliers, can significantly outperform commonly used unsupervised anomaly detection methods such as Autoencoders. Strategies to further enhance the performance of outlier-exposed classifiers might involve task-specific post-processing and feature engineering.

6. ACKNOWLEDGMENTS

I thank Verena Haunschmid and Patrick Praher for initial discussions, Khaled Koutini for making the implementation of the receptive-field-regularized ResNet available, and Gerhard Widmer for his helpful suggestions.

7. REFERENCES

[1] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. [Online]. Available: https://openreview.net/forum?id=HyxCxhRcY7

[2] L. Ruff, R. A. Vandermeulen, B. J. Franks, K. Müller, and M. Kloft, "Rethinking assumptions in deep anomaly detection," *CoRR*, vol. abs/2006.00339, 2020. [Online]. Available: https://arxiv.org/abs/2006.00339

[3] 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

[4] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive-field-regularized CNN variants for acoustic scene classification," *CoRR*, vol. abs/1909.02859, 2019. [Online]. Available: http://arxiv.org/abs/1909.02859

[5] 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

[6] 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

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[8] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification," in *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*, 2019, pp. 1–5. [Online]. Available: https://doi.org/10.23919/EUSIPCO.2019.8902732

[9] K. Koutini, S. Chowdhury, V. Haunschmid, H. Eghbal-zadeh, and G. Widmer, "Emotion and theme recognition in music with frequency-aware rf-regularized CNNs," *CoRR*, vol. abs/1911.05833, 2019. [Online]. Available: http://arxiv.org/abs/1911.05833

[10] 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 Trans. Audio Speech Lang. Process.*, vol. 27, no. 1, pp. 212–224, 2019. [Online]. Available: https://doi.org/10.1109/TASLP.2018.2877258

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

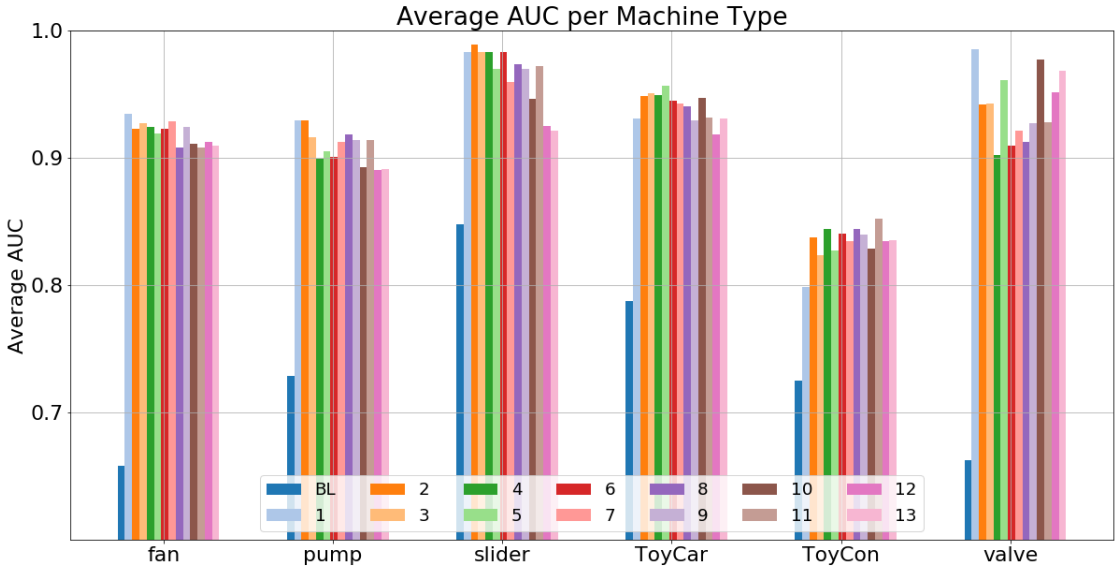


Figure 2: Average AUC per Machine Type

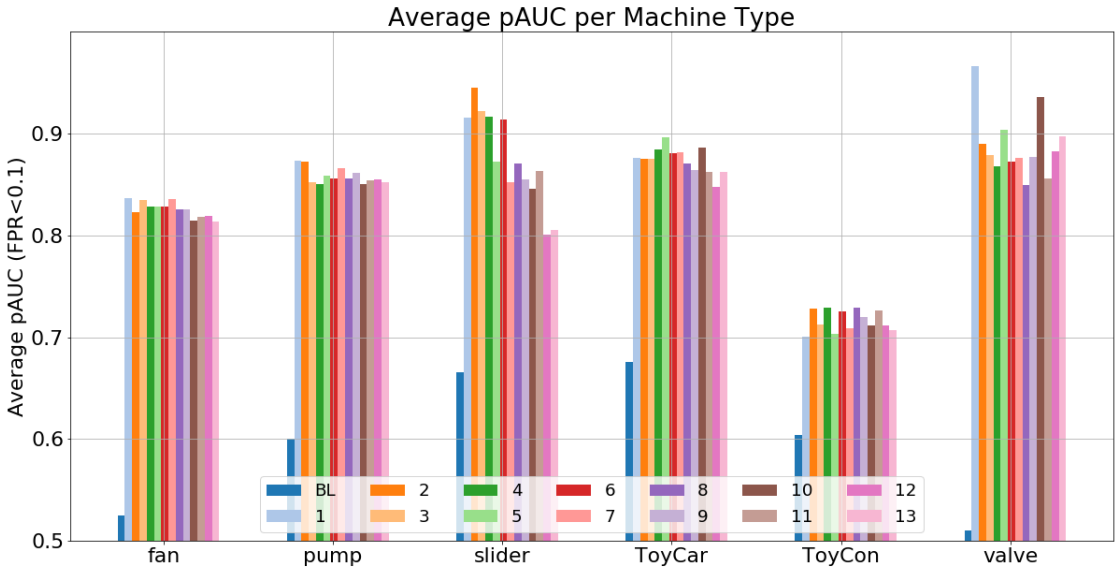


Figure 3: Average pAUC per Machine Type