ANOMALOUS SOUND DETECTION WITH MASKED AUTOREGRESSIVE FLOWS AND MACHINE TYPE DEPENDENT POSTPROCESSING

Technical Report

Verena Haunschmid

Institute of Computational Perception
Johannes Kepler University Linz
Austria
verena.haunschmid@jku.at

Patrick Praher

Software Competence Center Hagenberg GmbH
Hagenberg
Austria
patrick.praher@scch.at

ABSTRACT

This technical report describes the submission from the CP JKU/SCCH team for Task 2 of the DCASE2020 challenge - Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring. Our approach uses a Masked Autoregressive Flow (MAF) model for density estimation trained solely on normal samples. Anomaly scores per input snippet are computed using the negative log likelihood of new samples. The anomaly scores per input audio are aggregated using different metrics depending on the machine type instead of simply averaging them.

Index Terms— Anomalous Sound Detection, Masked Autoregressive Flows, Fault Detection

1. INTRODUCTION

Anomalous sound detection may prevent faults or malicious activities of machines and is therefore of great interest for condition monitoring in many different industrial applications. As it is often the case that lots of normal sound samples for each machine are available, and very little (or no) anomalous ones, Task 2 of DCASE2020 is concerned with the "Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring" [1]. For this task, two datasets (ToyADMOS[2] and MIMII[3]) with a total of 6 different machine types (ToyCar and ToyConveyor from ToyADMOS; valve, pump, fan and slider from MIMII) have been provided. For each machine type recordings of normal and anomalous sounds for a few different machines exist. Using only normal sounds, the goal of this task is to develop an anomaly score calculator that predicts a large value for anomalous sounds and a low value for normal sounds. The systems are evaluated using the area under the receiver operating characteristic (ROC) curve (AUC) and the partial-AUC (pAUC) (with $p = 0.1$). For more details we refer the reader to the task website[1].

In this report we describe our system based on Masked Autoregressive Flows (MAFs) [4] to predict anomaly scores for sounds emitted from a target machine. We also describe the performance gain achieved by using other means of aggregating the anomaly scores per audio sample than simply using the mean over all snippets. Our source code is publicly available [2].

[1] http://dcase.community/challenge2020/task-unsupervised-detection-of-anomalous-sounds

[2] https://github.com/patrick-praher/DCASE2020_T2_Haunschmid_Praher_Public

The rest of the paper is structured as follows. The proposed model, the model training and details of the selected systems are described in Section 2. Results of our approach compared to the baseline on the development test set can be found in Section 3. We conclude in Section 4.

2. PROPOSED SYSTEM

Our approach is inspired by [5] who apply two different flow models for novelty detection in industrial time series data. We focus on one of the approaches, Masked Autoregressive Flows (MAFs), and use the negative log-likelihood (under the distribution learned by our model) of an unseen sample (computed per snippet, aggregated over the whole audio sample) as the anomaly score. The results are improved by summarising the snippet-wise anomaly scores per audio sample using the median (for rotating machines) and the standard deviation (for rectilinearily moving machines) instead of the mean. The used model architecture and the proposed postprocessing are described in the following.

2.1. Masked Autoregressive Flows for Anomaly Detection

Neural density estimators are one type of generative models which is best suited for tasks where evaluating densities is more important than generating new data. Papamakarios et al. [4] state that there are two families of neural density estimators that are both flexible and tractable: autoregressive models and normalizing flows. Certain autoregressive models (when the underlying transformation is invertible) can be viewed as normalizing flows. In their paper [4] they introduce a particular implementation of this type of flows and call them Masked Autoregressive Flows (MAFs). They also show that MAFs are a generalization of the earlier published RealNVP [6]. The advantage of MAFs over other similar architectures is its fast data likelihood estimation which is essential in anomaly detection. For an in depth introduction to MAFs and its relation to other normalizing flows we refer the reader to [4].

We use conditional MAFs which is a natural extension of unconditional MAFs where we estimate the density $p(x|y)$ instead of $p(x)$. This allows us to give side-information such as the machine ID or machine type to our model during training. Using the machine type and machine ID as conditional label results in a 6- and 41-dimensional one-hot encoded vector y, respectively.

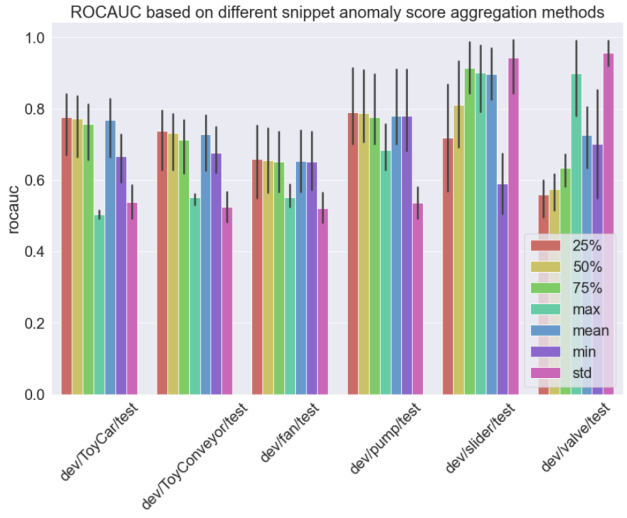

Figure 1: This figure shows the AUC computed for the same model, but using different metrics to aggregate the anomaly scores per input audio. The black lines indicate the variance between the individual machines.

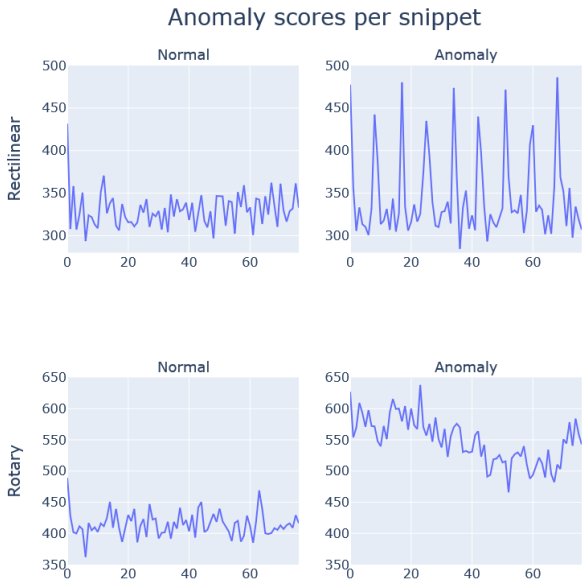

Figure 2: This figure shows the anomaly scores computed by one of our flow based models for some representative examples for a rectilinearly moving machine (valve: normal_id_02_00000038, anomaly_id_02_00000003) and a rotary machine (fan: normal_id_06_00000043.wav, anomaly_id_06_00000010.wav).

Our implementation of MAFs is based on a publicly available Pytorch implementation[3].

We experimented with different number of frames per snippet (2, 4, 6, 8, 10), different number of units in the hidden layer (hidden_size $\in$ 512, 1024, 2048), different number of autoregressive layers (n_blocks $\in$ 2, 4, 6, 8, 10, 12), different number of hidden layers per block (n_hidden $\in$ 1, 2). All models were trained using batch normalization. The parameters of the selected systems are summarised in Section 3.

2.2. Postprocessing

Preliminary experiments have shown that aggregating the snippet anomaly scores into one prediction per audio sample can be improved by using other metrics than the mean (as used by the baseline). This behaviour was observed across different model types, including a fully connected Autoencoder similar to the baseline system, a fully convolutional Autoencoder, and Masked Autoregressive Flows. The AUC for one flow based model using different metrics for aggregating the snippet anomaly scores is shown in Figure 1. For the ToyCar, ToyConveyor, fan, and pump, several metrics (including mean/median) perform similarly well, whereas for the slider and especially the valve we can observe that other metrics such as the maximum value and the standard deviation outperform the others.

From our experience in condition monitoring this behavior is expected. The recorded sound of the machine types fan, ToyCar, ToyConveyor and pump result from a rotary motion (e.g. motor and axle components) whereas valves and sliders move rectilinearly. An anomaly in steadily operating rotating systems is expected to produce a shift in the anomaly scores for a majority of snippets which can be robustly detected by the median. On the contrary a fault in the periodically moving rectilinear systems only shows in a portion of the snippets and the anomaly score is canceled out when averaging. The anomaly scores per snippet for two different examples are shown in Figure 2, comparing a normal and an anomalous sample for a rectilinearliy and one rotary moving machine, respectively.

Examples from other machines follow the same behavior. It can easily be seen that averaging the anomaly scores for the rotating machine would provide a clear decision boundary. For the rectilinearly moving machines, the anomaly scores are low most of the time and averaging would cancel out the peaks that indicate anomalies in between.

2.3. Model training

The raw audio was preprocessed as described in the baseline system. For simplicity we cut each spectrogram to 311 bins (10 seconds). Before passing the spectrograms into the flow based models, each spectrogram was normalized using the mean and standard deviation per frequency bin. Those means and standard deviations were computed on the training set (a) per machine id (`norm_per_set=True`), or (b) for the whole training set (`norm_per_set=False`).

The audio samples in the development set were split into training (90%) and validation (10%) randomly. For training the Adam optimizer [7] with a learning rate of 0.0005 was used, and each model was trained for a maximum of 1000 epochs, with checkpointing when the validation loss improved by at least 0.5, and early stopping enabled with a patience of 50 epochs. 'Haunschmid_CPJKU_task2_2', 'Haunschmid_CPJKU_task2_3', and 'Haunschmid_CPJKU_task2_4' were trained for 220, 232, and 199 epochs, respectively.

[3] https://github.com/kamenbliznashki/normalizing_flows

	nh	hs	nb	# params	Val. Loss
System 2	1	2048	4	29.72M	408.42
System 3	2	1024	4	14.87M	413.85
System 4	1	1024	6	16.01M	411.59

Table 1: Parameters and training details of our MAF-based submitted systems. nh: number of hidden layers per block; hs: size of hidden layer (units); nb: number of MADE-blocks

	ToyCar	ToyConv.	fan	pump	slider	valve
Baseline	78.77	72.53	65.83	72.89	84.76	66.28
System 1	79.49	73.58	66.30	73.65	91.44	85.24
System 2	81.92	73.46	75.69	79.84	93.55	94.50
System 3	79.99	73.87	75.00	78.83	93.57	94.56
System 4	79.38	72.81	75.23	79.13	93.47	94.49

Table 2: AUC averaged over the machine type for the submitted systems and the reported baseline.

2.4. Submitted Systems

In total four different systems are submitted to the challenge. 'Haunschmid_CPJKU_task2_1' uses the provided baseline model and applies our proposed postprocessing to see how it compares with other submissions. The systems 'Haunschmid_CPJKU_task2_2', 'Haunschmid_CPJKU_task2_3' and 'Haunschmid_CPJKU_task2_4' consist of a MAF model and our proposed postprocessing. We ranked our models similarly as described in the task description (in Step 3 we did not take into account whether the averaged ranks are the same). We picked the 3 best performing models based on this ranking on the development test set.

To keep the systems simple (as desired by the organizers) the three previously mentioned systems each consist of one model trained on all training data. The best performing MAF models were among those trained using normalization per machine ID (`norm_per_set=True`), using 4 frames per snippet, and conditioning on the machine ID. Other parameters of our selected systems are summarised in Table 1.

3. RESULTS

In this section we summarise the results of our submitted systems on the development test set and compare to the baseline results reported on the challenge website. The AUC per machine ID is shown in Figure 3, AUC and pAUC averaged per machine type are shown in Tables 2 and 3.

From the results in Tables 2 and 3 it can be seen that we improved the performance on the majority of the machine types (fan, pump, slider, valve). Interestingly, those are all machines from the MIMII [3] dataset. Looking at the AUC results per machine type in the MIMII dataset for our best performing submission, we can see that we perform between around 7 (pumps) and 28 (valves) percentage points better than the baseline. The performance improvement is not evenly distributed across different machines per type (visible in Figure 3). For pAUC the performance gap is even larger, we were able to improve the results by 31 percentage points for valves. For machines from the ToyADMOS [2] dataset the results improved only slightly (most obvious for the AUC of System 2).

	ToyCar	ToyConv.	fan	pump	slider	valve
Baseline	67.58	60.43	52.45	59.99	66.53	50.98
System 1	68.60	61.31	53.11	60.18	78.71	59.08
System 2	67.05	60.98	62.13	69.48	87.76	81.86
System 3	66.05	61.59	60.87	68.73	88.65	81.37
System 4	65.54	60.60	61.11	69.09	88.00	81.90

Table 3: pAUC averaged over the machine type for the submitted systems and the reported baseline.

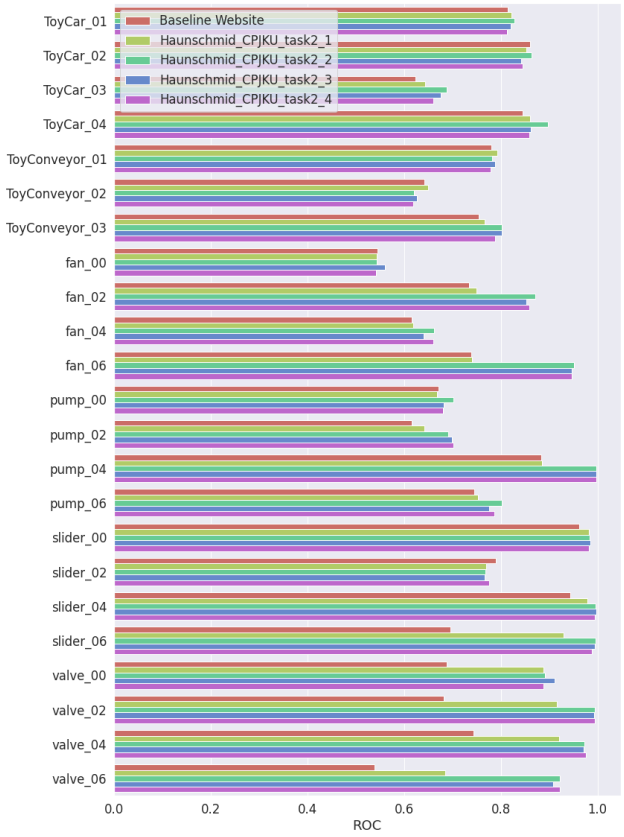

Figure 3: This plot shows the AUC per machine id for the reported baseline and the four submitted systems.

4. CONCLUSION

This technical report describes our submissions for Task 2 - Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring - of the DCASE2020 challenge. We use Masked Autoregressive Flows to learn the density of the training set (which only contains normal samples). Our experiments showed that normalizing the spectrograms is necessary for the MAF to learn properly, and that normalizing per machine ID outperforms normalizing the data with the global mean and standard deviation across many architectures. We also saw that using the machine ID as label when conditioning the MAF outperforms conditioning on the machine type. To keep it simple we do not use any of the provided external data, data augmentation or ensemble methods. The anomaly score for each audio sample is computed by aggregating the nega-

tive log likelihood scores for all snippets per audio sample. Instead of averaging over the snippet scores (as in the baseline method) we use the median for rotary moving machines and the standard deviation for rectilinearly moving machines, respectively. Combining both ideas we reach the same performance as the baseline for two machine types (ToyCar, ToyConveyor) and outperform the baseline by a large margin for the majority of machine types (fan, pump, slider, valve).

5. ACKNOWLEDGMENT

We thank Paul Primus for helpful discussions.

Parts of this research have been funded within the project AutoDetect (FFG project no. 862019) and by the BMK, BMDW, and the Province of Upper Austria in the frame of the COMET Programme managed by FFG.

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] G. Papamakarios, I. Murray, and T. Pavlakou, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 2338–2347. [Online]. Available: http://papers.nips.cc/paper/6828-masked-autoregressive-flow-for-density-estimation

[5] M. Schmidt and M. Simic, "Normalizing flows for novelty detection in industrial time series data," *CoRR*, vol. abs/1906.06904, 2019. [Online]. Available: http://arxiv.org/abs/1906.06904

[6] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=HkpbnH9lx

[7] 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*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980