UNSUPERVISED ANOMALOUS SOUND DETECTION BY SIAMESE NETWORK AND AUTO-ENCODER

Technical Report

Jan Tožička, Karel Durkota, Michal Linda

NeuronSW SE
{jan.tozicka, karel.durkota, michal.linda}@neuronsw.com

ABSTRACT

This paper describes our submission to the DCASE 2021 challenge Task 2 "Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring under Domain Shifted Conditions." Acoustic-based machine condition monitoring is a challenging task with a very unbalanced training dataset. In this submission, we propose two methods for anomaly detection and then their combination. The first method is based on feature extractor using Siamese Network with triplet loss and KNN for the actual anomaly detection. The second method uses very small auto-encoder on top of the OpenL3 embeddings. The combination of these two approaches selects the best performing method for each machine type.

This is a novel approach and have not been used by NeuronSW SE so far.

***Index Terms*—** Predictive Maintenance, Anomaly Detection, Siamese Network, Auto-encoder, OpenL3

1. INTRODUCTION

Machine condition monitoring is an essential component of predictive maintenance. It allows to schedule maintenance work to fix machine problems in the earliest stages and thus reducing maintenance costs and preventing consequential damages. Acoustic emission monitoring can be used for machine condition analysis and prognosis. ISO 22096[1] suggests that the nature of acoustic emissions can be used even without an understanding of the operating mechanics of the monitored machine. The recent progress in AI allows us to create an automatic machine condition monitoring system. To allow a large scale, we need a system that does not require the knowledge of the monitored system's operation mechanics. Nevertheless, it is impossible to collect all possible failures for a newly monitored machine without such knowledge. In practice, it is exceptional to get even any example of a failed state. Unfortunately, most of the recently developed AI methods require a huge amount of well-labeled examples, which makes them unusable for the task of machine condition monitoring. Task of learning from a few or even just one sample is called a few or one-shot learning. On the other hand, anomaly detection methods seem suitable for this problem as it lacks the samples representing the failure modes of the monitored machines.

We have experimented with two different approaches. The first one is based on Siamese Network[1] and KNN anomaly detector and the second one on OpenL3 embeddings and auto-encoder.

[1]https://www.sis.se/api/document/preview/908883/

2. SIAMESE NETWORK WITH KNN

This approach can be divided into three phases: (1) converting audio to spectrograms, (2) training Siamese Network as a classifier, and (3) train KNN, using Siamese Network encoder, for anomaly detection. We will describe each part separately.

2.1. Audio Transformation

First, we transform all audio samples into spectrograms using mel-spectrogram using Librosa python package. We used method *melspectrogram* with parameters as follows: n_fft=4096, hop_length=2048, n_mels=128, power=2.0 and fmin=10. Finally, values were converted to decibels and standardized.

2.2. Siamese Network

We used Siamese Network with triplet-loss introduced in [2]. The network is essentially an *encoder* that transforms the inputs into multi-dimensional latent space using the same weights. However, the network is trained to encode input data in such a way, that different input classes are distant in the latent space. To this end, network encodes three images called *anchor* (A), *positive* (P) and *negative* (N). *Anchor* and *positive* must come from the same class while *negative* must come from a different class then *anchor*. The network is trained to minimize distance between A and P, while maximize distance between A and N. Formally, loss function is as follows (it is slightly different than in the original paper):

$$\mathcal{L}(A, P, N) = \max(D(A, P) - D(A, N + margin, D(A, P))$$

where $D(x, y) = \|x - y\|^2$ is euclidean distance and α is a margin between positive and negative samples, in our case $\alpha = 10$.

During the training, random triplets are generated following the mentioned rule. For this challenge, we considered combination of machine type and section as a class (e.g. slider_id_00 and slider_id_01 belongs to different classes).

The network architecture is as follows:

- Input (131x79)
- Conv (64 @ 9x9) + BN + MaxPooling2D + ReLU
- Conv (128 @ 7x7) + BN + MaxPooling2D + ReLU
- Conv (256 @ 5x5) + BN + MaxPooling2D + ReLU
- Conv (512 @ 3x3) + BN + MaxPooling2D + ReLU
- Dense (512) + ReLU
- Dense (256) + ReLU

- Output (256)

Training From the training dataset we used 95% of the data of each class to generate 1 million triplets to train Siamese Network. After the training stagnated (patience=15 epochs), we continued with fine-tune training of seven individual Siamese Networks, one per each machine type (fan, gearbox, pump, slider, valve, ToyCar, and ToyTrain). Here, each classifier is trained to distinguish id's of a specific machine type.

2.2.1. Submission 1

Our first submission is the results of Siamese network's before fine-tunning.

2.2.2. Submission 2

In our second submission is the Siamese Network after fine-tuning.

2.3. KNN Anomaly Detection

Finally, once the Siamese network is trained, we use it as an encoder to tranform data into latent space, upon which KNN is trained. Standard KNN for anomaly detection [3] is trained using PyOD [2] library. To train KNN we use the default setting from PyOD with n_neighbors 5, method large, radius 1.0, and leaf_size 30.

3. OPENL3 WITH AUTO-ENCODER

Next approach we evaluated is based on pretrained OpenL3[4, 5] embeddings with a small auto-encoder.

For our experiments we've choosed OpenL3 with the following parameters:

- Input Representation: *mel128*
- Content Type: *env*
- Embedding Size: *512*
- Hop Size: *0.1*

Each 20 frames (corresponding to 2 seconds of audio) of OpenL3 embeddings are averaged and the 512-dimensional output is passed into auto-encoder. Since the embeddings are supposed to be well preprocessed, the chosen auto-encoder is very small (33.6k parameters):

- Input (512)
- Dense (32) + ReLU
- Dense (4) + ReLU
- Dense (32) + ReLU
- Dense (512)
- Output (512)

The anomaly score is then calculated as the average reconstruction error over all aggregated frames of the audio sample.

4. RESULTS

To evaluate both approaches, we used Development data of DCASE2021 Task-2 Challenge [6, 7, 8]. In Table 1 we summarize the AUCs and pAUCs for $p = 0.1$ of all four submissions (for detailed results see Appendix). We can see that submissions 2 and 3 are suitable for different kind of domains (highlighted in the table) and thus we decided to submit the fourth system choosing the best method for each machine type. Specifically, submission 3 outperformed submission 2 in domains *gearbox* and *ToyCar*.

	Submis. 1		Submis. 2		Submis. 3	
problem	AUC	pAUC	AUC	pAUC	AUC	pAUC
fan	54.4	52.6	**62.1**	**58.3**	57.8	51.9
gearbox	56.6	49.7	65.7	58.2	**73.9**	**58.9**
pump	60.1	52.6	**62.4**	**57.0**	56.9	53.5
slider	69.2	60.0	**68.4**	**58.5**	64.7	54.3
valve	63.7	57.7	**74.5**	**64.6**	55.1	52.1
ToyCar	54.1	50.5	60.5	55.0	**74.1**	**58.2**
ToyTrain	49.8	50.4	**59.4**	**58.8**	62.0	49.6
Average	57.7	53.1	64.4	58.5	62.0	54.1

Table 1: Harmonic means of AUCs and pAUCs on development dataset. The Submission 4, which combines highlighted models, scored average AUC: 67.8, and pAUC: 59.2

5. CONCLUSION

We have evaluated two different approaches, Siamese Network with KNN and OpenL3 embeddings with AE. Their combination scored average scored average AUC: 67.8 %, and pAUC: 59.2 %, and outperformed the baseline AutoEncoder solution by 5.6 and 5.9 percent points, respectively.

6. ACKNOWLEDGMENT

We want to thank NeuronSW SE company, for providing computational resources to train the algorithms.

Appendix

Detailed results for each class is shown in Table 2.

7. REFERENCES

[1] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[3] S. Ramaswamy, R. Rastogi, and K. 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.

[4] J. Cramer, H.-H. Wu, J. Salamon, and J. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 05 2019, pp. 3852–3856.

[2] https://pyod.readthedocs.io/

problem	Submis. 1 AUC	Submis. 1 pAUC	Submis. 2 AUC	Submis. 2 pAUC	Submis. 3 AUC	Submis. 3 pAUC
fan-0 S	50.9	50.4	45.7	48.3	**66.3**	**54.8**
fan-0 T	50.7	48.2	62.6	53.3	**69.8**	**54.9**
fan-1 S	63.8	60.1	**85.3**	**81.1**	49.3	49.6
fan-1 T	58.9	54.2	**89.5**	**77.1**	48.6	48.3
fan-2 S	52.7	48.9	**61.9**	53.0	59.0	**53.4**
fan-2 T	51.9	**55.7**	50.7	52.0	**60.0**	51.3
gearbox-0 S	59.4	50.1	**74.8**	**70.0**	60.1	54.1
gearbox-0 T	59.8	50.1	**84.0**	**72.7**	83.9	67.4
gearbox-1 S	56.9	51.9	57.1	48.5	**81.8**	**61.5**
gearbox-1 T	57.2	51.0	53.6	49.6	**83.0**	**62.5**
gearbox-2 S	52.0	47.8	65.4	**57.0**	**66.2**	54.5
gearbox-2 T	54.9	47.7	68.6	**59.7**	**72.9**	59.5
pump-0 S	57.4	51.3	47.4	51.3	**65.9**	**64.3**
pump-0 T	54.0	50.7	53.1	**51.2**	**58.1**	50.9
pump-1 S	79.3	66.8	**84.9**	**71.9**	71.5	55.1
pump-1 T	65.0	48.9	**81.9**	**59.3**	36.8	47.6
pump-2 S	57.7	51.8	63.2	**59.8**	**63.8**	56.1
pump-2 T	53.6	49.8	**61.6**	**53.1**	59.6	50.3
slider-0 S	75.9	65.4	**75.9**	**65.4**	67.8	55.4
slider-0 T	63.2	53.1	63.2	**53.1**	**63.3**	51.9
slider-1 S	84.9	66.6	**84.9**	**66.6**	81.0	63.4
slider-1 T	**65.3**	**56.2**	**65.3**	**56.2**	56.0	51.1
slider-2 S	**80.4**	**68.5**	**80.4**	**68.5**	66.5	54.8
slider-2 T	**55.0**	**54.0**	**55.0**	**54.0**	58.9	50.8
valve-0 S	67.8	64.3	**74.0**	**69.2**	53.0	49.9
valve-0 T	64.6	57.2	**80.4**	**69.4**	45.3	49.2
valve-1 S	54.2	52.1	**71.6**	**64.6**	52.8	52.1
valve-1 T	62.4	54.2	68.7	54.6	**71.2**	**60.8**
valve-2 S	80.8	77.8	**98.3**	**94.2**	62.5	51.7
valve-2 T	58.6	48.8	**63.1**	**51.2**	53.0	50.3
ToyCar-0 S	48.4	50.9	61.5	**61.5**	**79.1**	59.8
ToyCar-0 T	55.2	49.6	**60.2**	52.7	59.1	**53.5**
ToyCar-1 S	49.0	48.8	56.5	52.5	**72.9**	**58.1**
ToyCar-1 T	51.6	49.2	59.4	55.1	**83.0**	**64.2**
ToyCar-2 S	62.4	53.8	67.4	57.1	**85.3**	**60.1**
ToyCar-2 T	61.1	51.0	58.8	52.0	**73.5**	**54.9**
ToyTrain-0 S	43.4	47.4	**55.5**	**55.5**	34.5	47.4
ToyTrain-0 T	48.0	54.1	**55.9**	**54.2**	47.9	48.2
ToyTrain-1 S	50.8	49.1	**68.8**	**65.2**	63.8	54.9
ToyTrain-1 T	47.7	51.4	54.0	**60.6**	**55.9**	49.8
ToyTrain-2 S	56.5	49.4	63.6	**59.5**	**65.8**	47.8
ToyTrain-2 T	54.7	51.5	**62.3**	**59.8**	54.7	50.3
Average	57.7	53.1	**64.4**	**58.5**	62.0	54.1

Table 2: Detailed results for each section, source(S) and target (T). Maximal values in each row are highlighted.

[5] R. Arandjelović and A. Zisserman, "Look, listen and learn," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[6] 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.

[7] 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.

[8] 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," *In arXiv e-prints: 2106.04492, 1–5*, 2021.