DCASE CHALLENGE 2020: UNSUPERVISED ANOMALOUS SOUND DETECTION OF MACHINERY WITH DEEP AUTOENCODERS

Technical Report

Anahid Jalali[1], *Alexander Schindler*[1], *Bernhard Haslhofer*[1],

[1] Austrian Institute of Technology, Vienna, Austria,
anahid.jalali@ait.ac.at
alexander.schindler@ait.ac.at
bernhard.haslhofer@ait.ac.at

ABSTRACT

In our work, we present an unsupervised anomalous sound detection framework trained on DCASE2020 audio dataset. This dataset is a subset of two datasets ToyADMOS and MIMII. We use the state of the art anomaly detection approach, deep autoencoder architecture trained on Mel-spectrograms. This architecture uses LSTM-RNN units to learn the normal condition of the machine, and is proven efficient at detecting diverse machine anomalies. Our trained model on MIMII dataset achieves average result of 73.51% AUC and 57.90% pAUC, resulting in an improvement compared to the baseline system with the average results of 72.44% AUC and 57.48 pAUC. The average performance of the baseline system on ToyADMOS dataset is 75.65% AUC and 64% pAUC, where our model reaches to average of 73.21% AUC and 61.91% pAUC. Our system reaches overall average of 73.41% AUC and 59.27% pAUC on the development data set, with overall similar performance to the baseline system with average of 73.51% AUC and 59.66% pAUC.

***Index Terms*—** anomaly detection, anomalous sound detection, machine learning

1. INTRODUCTION

Automatic Anomalous Sound Detection (ASD) is a system that identifies abnormal sounds emitted from a specific equipment and is considered as an essential technology in industry 4.0 [1]. Such systems are often used for machine condition monitoring and aim to detect unknown anomalous sounds. In real-life practices, anomalies are infrequent and are of various forms. Therefore, an extensive and time consuming data collection process is needed to capture all the variations of anomalies from a machine. Thus, only data from normal condition of the machinery is collected and used as training samples and the system only learns the natural routine of the targeted equipment to recognize an abnormal behaviour. DCASE2020 challenge of unsupervised anomalous sound detection [2] focuses on this issue, where participants are asked to use the provided audio dataset and submit their results. The audio dataset provided by organizers of this task contains recordings of 6 different types of machines that are parts of ToyADMOS[3] and MIMII Dataset[4]; Pump, Fan, Slider, ToyCar, ToyConveyor and Valve. Each machine type has maximum of four machine id, which indicates the machine's identifier. The dataset is available under 3 different releases:

- Development set: contains a train set and a test set for each machine (roughly 83 hours)
- Extra training set: contains more training data for each machine (roughly 44.88 hours)
- Evaluation set: contains evaluation data for each equipment (roughly 19.70 hours)

Furthermore, DCASE community provides a baseline system [1], a dense autoencoder with 8 layers (4 encoding and 4 decoding layers) each with 128 units. The bottleneck of this architecture has 8 units with rectified linear unit (ReLU) activation function. Each layer of the autoencoder is followed by a batch normalization layer and a dense layer with size 640 (number of features) is defined as its output layer. This model is trained on 5-consecutive (2*P+1, where P is the context window size) frames of log Mel band energies of the size 128 and 64 ms analysis window (50% hope size) resulting an input with the dimension of 640. Evaluation metrics used for this task are Area Under Receiver Operating Characteristic (ROC) curve (AUC) 1 and the partial AUC (pAUC) as illustrated in 2.

$$AUC = \frac{1}{N_{-}N_{+}} \sum_{i=1}^{N_{-}} \sum_{j=1}^{N_{+}} \mathcal{H}(\mathcal{A}_{\theta}(x_j^{+}) - \mathcal{A}_{\theta}(x_i^{-})) \tag{1}$$

$$pAUC = \frac{1}{\lfloor \frac{pN_{-}}{2} \rfloor N_{+}} \sum_{i=1}^{\lfloor pN_{-} \rfloor} \sum_{j=1}^{N_{+}} \mathcal{H}(\mathcal{A}_{\theta}(x_j^{+}) - \mathcal{A}_{\theta}(x_i^{-})) \tag{2}$$

where

$$\mathcal{H}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Baseline results on MIMII dataset has the average AUC of 72.44% and average pAUC of 57.48% and on ToyADMOS dataset, it reaches the average AUC of 75.65% and pAUC of 64%. Overall average results of the benchmark system on both datasets have 73.51% AUC of and 59.66% pAUC. More details of their results for each machine type and machine id is provided in section 3.

In this work, we consider a popular Recurrent Neural Network (RNN) architecture, called sequential Long Short Term Memory (LSTM) to build an autoencoder for our unsupervised anomaly detection system. We use Mel-spectrograms for our model's input as they prove robust in capturing audio features and appropriate input for training neural networks [5]. In the remaining of this report, we present our model architecture (section 2) and our experimental results (section 3) as well as the discussion of our analysis.

2. MODEL ARCHITECTURE

One of the popular architectures of Artificial Neural Networks (ANNs) for unsupervised anomaly detection is Deep-Autoencoders [6, 7]. This architecture aims to rebuild the given input data while lowering the reconstruction error. The reconstruction error is the difference between the actual input and reconstructed output. Moreover, RNNs have proven to be robust in capturing temporal behaviour of the data through their feedback connections and therefore, appropriate for time series data such as audio signals. A LSTM-RNN cell consists of an input gate, an output gate and a forget gate. Use of a tanh function increases the ability of this cell to capture as much information about the past (or future for a bi-directional cell) and a forget gate to drop the less relevant information. A sequential LSTM-autoencoder is a stack of LSTM layers (encoding layer), which encode the information and their outputs are passed into a bottleneck layer, which is a LSTM layer with smaller size together with a repeat vector layer. A repeat vector layer, as its name states, repeates its input vector multiple times. For a LSTM-autoencoder, it repeats the encoded information for n_ts times, where n_ts is the number of time steps. The encoded information is then passed as the input to a stack of sequential LSTM layers to reconstruct the original input. This architecture is depicted in 2. In this figure, n_units is the size of the LSTM, $bottleneck\_size$ is the size of the LSTM cells used as the bottleneck and are smaller than its previous layer. We use a fully connected time distributed Dense layer as our autoencoder's output layer.

3. RESULTS

The feature dimension to our model is 128 log mel-bands that are extracted from 0.064 seconds analysis time window with 0.032 ms overlap over 15 time steps. Our autoencoder has 4 encoding and 4 decoding layers with a bottleneck of the size 8. A fully connected dense layer is used as the output layer of the model resulting in 755776 of total parameters. The activation function in each layer is a tanh function and a dropout of size 0.2 is set at each encoding layer. We use RMSProp with 0.0001 learning rate and 0.01 decay to compile the model. The model is trained on 90% of the train set and evaluated on the remaining 10%, over 100 epochs. We further set an early stopping to monitor the evaluation loss with 20 patience. We choose this set of parameters based on our parameter optimization processes. Using early stopping, we monitor the changes in evaluation loss at each training epoch and will stop at x^{th} training epoch (where $x <= patience\_value$), if we observe no improvement in the evaluation loss[8].
The results of our experiments compared to DCASE2020 baseline system is presented in the following tables.
We further investigate the results in depth for both datasets. For this purpose, we take the reconstructed error on the train set as anomaly threshold and create the predicted labels to calculate the confusion matrix for each machine. First, we discuss the results of our model on ToyADMOS, where it almost reaches the baseline system. The outcome of our model on ToyCar, in terms of AUC and pAUC value is lower than the benchmark. Investigating the model outputs, we notice the higher tpr (true positive rate) of our model (86.68%) compared to the baseline system (78.28%) over all anomalous samples.

Figure 1: Architecture of a sequential LSTM-autoencoder

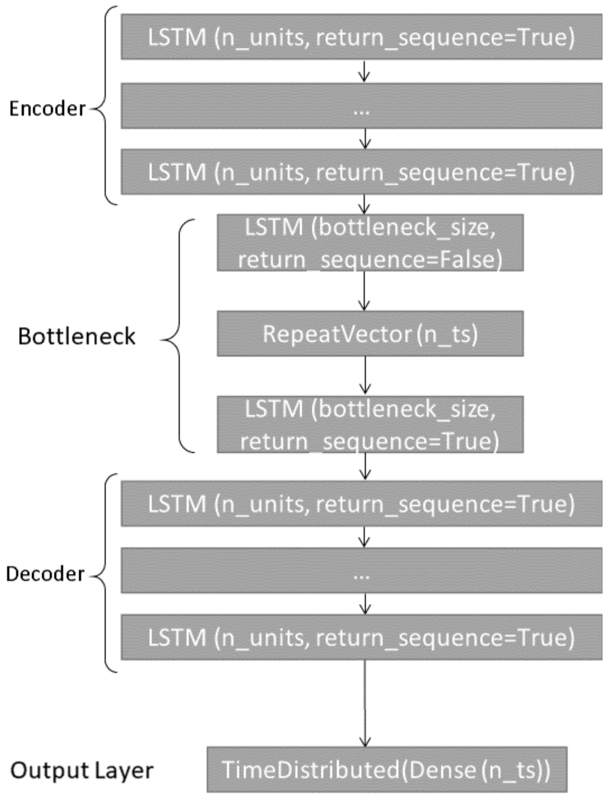

Encoder
LSTM (n_units, return_sequence=True)
...
LSTM (n_units, return_sequence=True)
Bottleneck
LSTM (bottleneck_size, return_sequence=False)
RepeatVector (n_ts)
LSTM (bottleneck_size, return_sequence=True)
Decoder
LSTM (n_units, return_sequence=True)
...
LSTM (n_units, return_sequence=True)
Output Layer
TimeDistributed(Dense (n_ts))

Table 1: Results of LSTM-autoencoder compared to Dense autoencoder (Baseline) on ToyCar

ToyCar				
Machine ID	Dense-autoencoder		LSTM-autoencoder	
	AUC	pAUC	AUC	pAUC
01	81.36%	68.40%	80.89%	67.75%
02	85.97%	77.72%	85.35%	78.01%
03	63.30%	55.21%	60.81%	54.79%
04	84.45%	68.97%	75.47%	65.01%
Average	78.77%	67.58%	75.63%	66.39%

Table 2: Results of LSTM-autoencoder compared to Dense autoencoder (Baseline) on ToyConveyor

ToyConveyor				
Machine ID	Dense-autoencoder		LSTM-autoencoder	
	AUC	pAUC	AUC	pAUC
01	78.07%	64.25%	76.63%	61.26%
02	64.16%	56.01%	63.95%	54.16%
03	75.35%	61.03%	71.81%	57.47%
Average	72.53%	60.43%	70.80%	57.63%

However, baseline system achieved a lower fpr (false positive rate) (68% recall) compared to our model (63% recall). Our model's performance on ToyConveyor reached 70% precision and 57% re-

Table 3: Results of LSTM-autoencoder compared to Dense autoencoder (Baseline) on Fan

Fan				
Machine ID	Dense-autoencoder		LSTM-autoencoder	
	AUC	pAUC	AUC	pAUC
00	54.41%	49.37%	56.68%	49.32%
02	73.40%	54.81%	74.74%	54.24%
04	61.61%	53.26%	62.82%	52.40%
06	73.92%	52.35%	75.03%	52.25%
Average	65.83%	52.45%	67.32%	52.05%

Table 4: Results of LSTM-autoencoder compared to Dense autoencoder (Baseline) on Pump

Pump				
Machine ID	Dense-autoencoder		LSTM-autoencoder	
	AUC	pAUC	AUC	pAUC
00	67.15%	56.74%	68.96%	55.57%
02	61.53%	58.10%	58.63%	58.65%
04	88.33%	67.10%	92.48%	74.31%
06	74.55%	58.02%	75.69%	55.52%
Average	72.89%	59.99%	73.94%	61.01%

Table 5: Results of LSTM-autoencoder compared to Dense autoencoder (Baseline) on Slider

Slider				
Machine ID	Dense-autoencoder		LSTM-autoencoder	
	AUC	pAUC	AUC	pAUC
00	96.19%	81.44%	97.61%	88.52%
02	78.97%	63.68%	79.67%	64.81&
04	94.30%	71.98%	93.20%	67.06%
06	69.59%	49.02%	69.50%	49.49%
Average	84.76%	66.53%	84.99%	67.47%

Table 6: Results of LSTM-autoencoder compared to Dense autoencoder (Baseline) on Valve

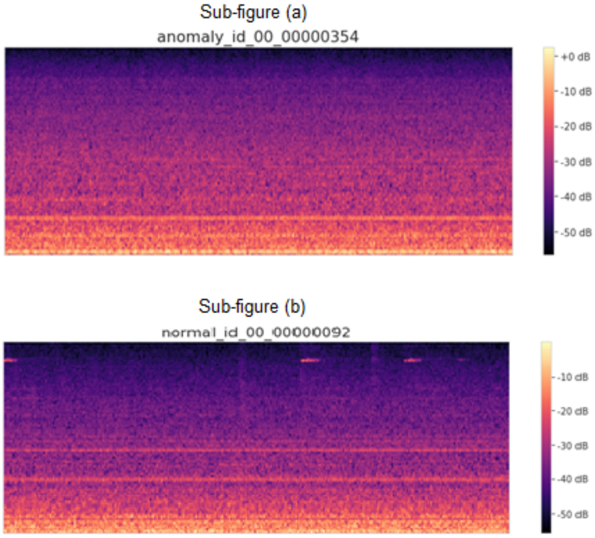
Valve				
Machine ID	Dense-autoencoder		LSTM-autoencoder	
	AUC	pAUC	AUC	pAUC
00	68.76%	51.70%	70.99%	51.57%
02	68.18%	51.83%	66.75%	52.36%
04	74.30%	51.97%	78.06%	51.84%
06	53.90%	48.43%	55.50%	48.50%
Average	66.28%	50.98%	67.82%	51.07%

call (similar to benchmark). However, here we also have higher fpr (53.81%) compared to the benchmark (52.77%), resulting in slightly lower AUC and pAUC value.

We further investigate the results of our experiments for each machine in MIMII dataset. In this dataset, our model always outperformed the baseline system.

Our study of model's output on fan shows that 66.16% anomalous cases are correctly detected as anomalies and the remaining cases is falsely detected as normal condition. We further investigate the spectrograms of such samples and notice the similarities between normal condition samples and such anomalies. This is shown in figure 3. The sub-figure (a) is an anomalous condition that is falsely detected as normal, compared to a true negative sample (sub-figure (b)).

Figure 2: Comparison of a false negative sample (fan's anomalous condition detected as normal condition) with a true negative sample (fan's normal condition). Sub-figure (a) is an anomalous case and is falsely detected as normal, whereas sub-figure (b) is correctly detected as normal condition.



Looking at the training data, we notice the similarities of the spectrograms between the normal conditions and these false positives. This can be seen in figure 3, where two normal condition samples are compared. Sub-figure (a) is falsely detected as anomaly, whereas sub-figure (b) is correctly detected as normal machine condition.

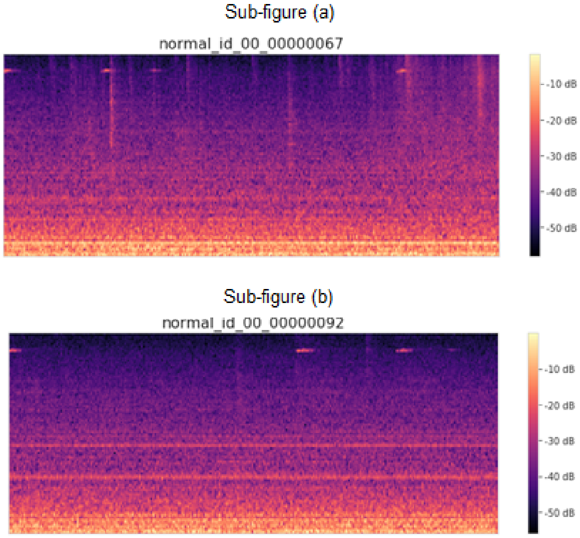
Moreover, we compare baseline's confusion matrix with ours, where we have lower fpr (baseline has 42.2% fpr while ours achieved 39.5%). Instead, baseline system has lower false negative cases (27.45% fnr, where our model has 33.83%). Since our model has lower fpr, we have achieved a higher AUC and pAUC.

Analyzing the results of pump for all its machine ids, our model reaches overall 82.45% truly detected anomalies and 47.77% truly detected healthy state (no anomalies). We have lower false positive and false negative rate compared to the benchmark resulting higher precision achieved by our model (70%) compared to the benchmark (65%).

We further investigate the results of our model on slider, where we observe a clear difference among all the machine ids of this equipment, noticeable in both audio and spectrogram. Our model detected almost all the anomalies. For this machine, our model performed as good as the benchmark. However, in terms of precision and classification, our model slightly outperformed the baseline by 1% (baseline accuracy is 78.99% as our model achieved 79.92%) .

Investigating the results of our model on valve, we notice higher tpr here as well, also slightly higher fpr. Overall, our model outperformed the benchmark in all AUC, pAUC, precision, accuracy and F1-score by almost 3%.

Figure 3: Comparison of a false positive sample (fan's normal condition detected as anomaly) with a true negative sample (fan's normal condition). Sub-figure (a) is falsely detected as anomaly, whereas sub-figure (b) is correctly detected as normal condition.



4. CONCLUSION

In this work, we used an LSTM-Autoencoder to detect anomalous sounds emitted from machines. For this purpose, we extracted log mel-band energies from the audio data. We segmented extracted mels into 15 consecutive frames to capture the temporal behaviour of the features. Our proposed system on MIMII dataset achieved an average result of 73.51% AUC and 57.90% pAUC, resulting in a slight improvement compared to the baseline system with an average results of 72.44% AUC and 57.48 pAUC. The baseline system on ToyADMOS dataset achieved average 75.65% AUC and 64% pAUC, where our model reached an average of 73.21% AUC and 61.91% pAUC. Our system performed similar to the baseline system (total average of 73.51% AUC and 59.66% pAUC) and has total average of 73.41% AUC and 59.27% pAUC on the development data set.

5. ACKNOWLEDGMENT

Thanks to Austrian Research Promotion Agency (FFG) for funding this work, which is a part of industrial project under the name DeepRUL, project ID 871357.

6. REFERENCES

[1] http://dcase.community/challenge2020/.

[2] 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

[3] 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

[4] 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

[5] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 1996–2000.

[6] B. Bayram, T. B. Duman, and G. Ince, "Real time detection of acoustic anomalies in industrial processes using sequential autoencoders," *Expert Systems*, p. e12564, 2020.

[7] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[8] https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping.