DCASE 2020 CHALLENGE TASK2 TECHNICAL REPORT

Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring

Vipin K Agrawal

Milpitas, CA 95035 USA
vagrawal@msense.ai

Shiv Shankar Maurya

Bangalore, KN 560105 India
ssmaurya@msense.ai

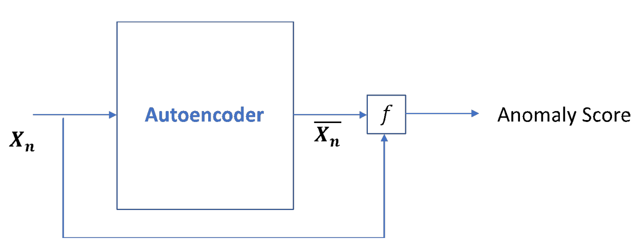
ABSTRACT

Autoencoders are a very popular approach in detecting anomalies in a system, where reconstruction error is generally used as an anomaly score. However, the reconstruction errors, generated in such manners, contain external noises of the system, making reconstruction errors as anomaly scores less effective. In this brief, we present an additional hypothesis that autoencoders may introduce additional statistical noise in the reconstruction errors as well.
Our proposal includes a design of an autoencoder, lays out a theoretical basis of designing a noise filter for reconstruction errors, and outlines various aggregation methods to reduce the effect of the noise. While further work is still needed, we are able to show the accuracy improvement by using various aggregation methods.

1. INTRODUCTION

Anomaly detection is a field of detecting outlier data indicating deviation from the normal behavior with generally some bad connotation. Anomalous events happen relatively infrequently but are disastrous in nature.

A popular approach in deep learning-based anomaly detection is to build a deep learning model, Autoencoder, to reduce the dimensionality of the data, and then reconstruct the input sample. An autoencoder belongs to a family of machine learning models, called neural networks, and more specifically deep neural networks. An autoencoder consists of an encoder, and a decoder.



And anomalous event is when anomaly score A_n is more than a threshold th:

$$f = \begin{cases} \mathbf{0}, & \boldsymbol{normal \quad if\ A_n < th} \\ \mathbf{1}, & \boldsymbol{anomaly\ if\ A_n \geq th} \end{cases}$$

The encoder maps a input vector X_n into a hidden representation Z_n. The decoder tries to reconstruct the sample back to vector $\overline{X_n}$. The difference between $\overline{X_n}$ and X_n is called reconstruction error.

reconstruction error $A_n = f(\overline{X_n} - X_n)$

Reconstruction error can be treated as an anomaly score A_n

The following issues may arise in the above system:

A. An autoencoder network may be able to learn anomalies equally well with perfect reconstruction, or anomalies and normal events are represented by the network with very close reconstruction errors.

   This issue can be solved by developing an effective autoencoder architecture and training it by providing ample amounts of training data. Training data can also be augmented to increase the accuracy of the system.

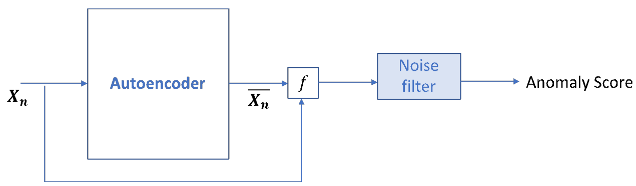
B. Presence of external noises may adversely affect the reconstruction error of the system. X_n is inherently noisy, and depending on the deployment, may affect the anomaly score for the given samples.

   If the signal-to-noise ratio (SNR) is high, there are methods to solve this issue using digital signal processing or machine learning based approaches. However, this issue requires a system approach to solve the problem if SNR is low, for example calculating anomalies over a longer period of time to filter out the short-term noises.

C. The presence of internal noise in an autoencoder means that the output of the autoencoder $\overline{X_n}$ is noisy.

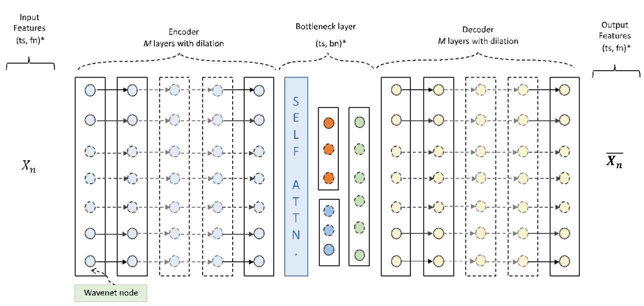
This issue can be solved by A) properly training the autoencoder, B) by regularizing the training process, and C) by analyzing the data over a longer time etc.

Because of the noise as mentioned above, the anomaly detection system may be updated as the following:



2. AUTOENCODER ARCHITECTURE

The proposed autoencoder is inspired by the Wavenet architecture. It consists of an encoder and a decoder with a bottleneck layer and a self-attention layer. Unlike the Wavenet architecture, all time steps are preserved in the bottleneck layer.



*** Input features**

As in the baseline system, we use a log-mel-spectrogram of the input X_n

- FFT calculation over 4096 or 8192 samples depending on the machine type, mentioned in the challenge.
- Analysis frame size 64 ms
- Log mel-band energy bands fn: 128 bands
- Input time-steps to the autoencoder ts: 4 or 32 Analysis frames depending on the machine type.

*** Hyperparameters**

The following are the additional model parameters apart from ts and fn

- Number of layers *M*
- Number of bottlenecks bn
- Use machine ID or not at the bottleneck layer.

3. NOISE FILTERS AND AGGREGATION METHODS

Developing effective noise filters are dependent on a deployment and other factors. For lack of time, we have not investigated the effective noise filters, and ML based approaches to noise filtering yet. However, following anomaly scores are evaluated and manually selected.

A. Calculate the Mean Square Error over the complete sample (default and state-of-art)

$$\boldsymbol{MSE = \frac{1}{N}\sum_{1}^{N}(\overline{X_n} - X_n)^2}$$

Where N is the number of the frames per sample.

B. Calculate the Median Square Error – this allows to filter sudden onset of the excessive noise for a short duration.

$$Median(\overline{X_n} - X_n)^2$$

C. The proposed error E1 calculation which involves adding the frequency mel bands over all the frames per sample, and then calculating MSE.

$$\boldsymbol{E1 = \frac{1}{NL}\sum_{1}^{L}\left[\sum_{1}^{N}\overline{X_n} - X_n\right]^2}$$

Where N is number of frames per sample, and L is the number of the log-mel-bands. This is helpful if the present noise in the system behaves like the white noise, and cancels itself over the period of time, resulting in only minor left-over noise in all of the bands.

D. The proposed error E2 calculation which involves adding frequency bands over all of the frames, and then calculating the mean absolute error (MAE) over all frequency bands.

$$E2 = \frac{1}{NL}\sum_{1}^{L}\left\|\sum_{1}^{N}\overline{X_n} - X_n\right\|$$

Where N is number of frames per sample, and L is the number of the log-mel-bands. This is helpful if the noise present in the system behaves like white noise and cancels itself over a period of time, and still there is significant left-over noise in one or more of the bands.

4. RESULTS

Baseline results are generated after running for 100 epochs.

Table 1. Baseline results

Machine Type	AUC	pAUC	Loss
ToyCar	78.77	67.58	MSE
ToyConveyor	72.53	60.43	MSE
Fan	65.83	52.45	MSE
Pump	72.89	59.99	MSE
Slider	84.76	66.53	MSE
Valve	66.26	50.98	MSE

All the machine types were trained for 500 epochs with early stopping patience of 50 epochs, and run on the test data.

Table 2. Accuracy results using proposed scheme

Machine Type	AUC	pAUC	Loss
ToyCar	95.64	85.99	E2
ToyConveyor	86.52	70.81	E2
Fan	86.71	70.58	E1
Pump	88.71	72.04	E1
Slider	92.36	76.11	E1
Valve	88.61	75.34	MSE

Table 3. Improvements

Machine Type	AUC	pAUC
ToyCar	16.87	18.41
ToyConveyor	13.99	10.38
Fan	20.88	18.13
Pump	15.82	12.05
Slider	7.6	9.58
Valve	22.35	24.36

5. REFERENCES

[1] http://dcase.community/workshop2020/.

[2] Adversarially Learned Anomaly Detection (https://arxiv.org/pdf/1812.02288.pdf)

[3] Unsupervised Detection of Anomalous Sound based on Deep Learning and the Neyman-Pearson Lemma (https://arxiv.org/pdf/1810.09133.pdf)

[4] Noise Reduction Using Minimum Mean Square Estimators (MMSE) (https://www.vocal.com/noise-reduction/minimum-mean-square-estimators)

[5] Yuma Koizumi, Shoichiro Saito, Hisashi Uematsu, Noboru Harada, and Keisuke 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), 308–312. November 2019. URL: https://ieeexplore.ieee.org/document/8937164.

[6] Harsh Purohit, Ryo Tanabe, Takeshi Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei 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), 209–213. November 2019. URL: http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Purohit_21.pdf.

[7] Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, Toshiki Nakamura, Yuki Nikaido, Ryo Tanabe, Harsh Purohit, Kaori Suefusa, Takashi Endo, Masahiro Yasuda, and Noboru Harada. *Description and discussion on DCASE2020 challenge task2: unsupervised anomalous sound detection for machine condition monitoring.* In arXiv e-prints: 2006.05822, 1–4. June 2020. URL: https://arxiv.org/abs/2006.05822.

[8] WaveNet: A Generative Model for Raw Audio (https://arxiv.org/abs/1609.03499)

[9] J. Chorowski, R. J. Weiss, S. Bengio and A. van den Oord, "Unsupervised Speech Representation Learning Using WaveNet Autoencoders," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041-2053, Dec. 2019, doi: 10.1109/TASLP.2019.2938863.

[10] A. Polyak and L. Wolf, "Attention-based Wavenet Autoencoder for Universal Voice Conversion," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 6800-6804, doi: 10.1109/ICASSP.2019.8682589.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 3, Article 15 (July 2009), 58 pages. DOI:https://doi.org/10.1145/1541880.1541882

[12] Chong Zhou and Randy C. Paffenroth. 2017. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD '17*). Association for Computing Machinery, New York, NY, USA, 665–674. DOI:https://doi.org/10.1145/3097983.3098052

[13] Ruoying Wang, Kexin Nie, Tie Wang, Yang Yang, and Bo Long. 2020. Deep Learning for Anomaly Detection. In *Proceedings of the 13th International Conference on Web*

Search and Data Mining (*WSDM '20*). Association for Computing Machinery, New York, NY, USA, 894–896. DOI:https://doi.org/10.1145/3336191.3371876