AN ENSEMBLE APPROACH FOR DETECTING MACHINE FAILURE FROM SOUND

Technical Report

Faruk Ahmed,[1] *Phong Nguyen,*[2] *Aaron Courville*[1,3]

[1] Mila-Université de Montréal, {faruk.ahmed, aaron.courville}@umontreal.ca
[2] Hitachi, Central Research Laboratory, phong.nguyen.kj@hitachi.com *
[3] CIFAR Fellow

ABSTRACT

We develop an ensemble-based approach for our submission to the anomaly detection challenge at DCASE 2020. The main members of our ensemble are auto-encoders (with reconstruction error as the signal), classifiers (with negative predictive confidence as the signal), mismatch of the time-shifted signal with its Fourier-phase-shifted version, and a Gaussian mixture model on a set of common short-term features extracted from the waveform. The scores are passed through an exponential non-linearity and weighted to provide the final score, where the weighting and scaling hyper-parameters are learned on the development set. Our ensemble improves over the baseline on the development set.

Index Terms— Anomalous sound detection, auto-encoders, classifiers

1. INTRODUCTION

To identify machines in failure states, human experts can monitor the sound emitted from the machines and raise an alarm if an anomalous sound indicates potential failure. However, performance from experts has been shown to be influenced by many factors such as emotion, health condition, and experience. Automatically identifying mechanical failure can therefore be very useful and a scalable solution.

In practice, real anomalies are rare and highly diverse. Therefore, it is hard to collect sufficiently large and representative samples of anomalous sounds, many of which might be unprecedented in real life. This means that in practice we would have to detect anomalous sounds that are likely unobserved in a training dataset.

Anomalous sound detection (ASD) is a well-known topic and has been studied for a long time, and various approaches have been investigated. In early studies, acoustic features for detecting anomalies are hand-crafted based on the characteristics of the target machine. Benefiting from the development of deep learning, deep neural network-based methods that do not require extensive knowledge of the features for machine sounds have also been actively studied.

We propose an ensemble-based approach in our submission for the task. We use auto-encoders with reconstruction errors as the signal, a self-predictive heuristic involving the *shift theorem* in signal processing, and a Gaussian mixture model on a set of common short-term features extracted from the waveform. The scores are combined with an exponential non-linearity and weighted to provide the final score, where the weighting and scaling hyper-parameters are learned on the development set. The reason we chose an ensemble approach is that we found different models to perform at different relative strengths for different machine types as well as machine IDs. Our ensemble improves the performance for ASD task compared to the baseline on the development dataset.

2. PROBLEM STATEMENT

DCASE 2020 Challenge Task 2's goal is to identify whether the sound emitted from a target machine is normal or anomalous. The main challenge of this task is to detect unknown anomalous sounds under the condition that only normal sound samples have been provided as training data, while samples of known anomalous sounds may be used for model selection and hyper-parameter tuning.

Let an L-point-long-time-domain observation $\boldsymbol{x} \in \mathbb{R}^L$ be an observation which includes a sound emitted from the target machine. Our task is to identify from this sequence whether the state of the target machine is normal or anomalous. The waveform includes background noise, which makes things harder, since now we must learn to attend to the machine alone when judging its state.

To perform anomaly detection, typically one needs to produce an *anomaly score* $\mathcal{S}_\theta(\boldsymbol{x})$, parameterised by θ, such that a larger score implies increased probability that the input signal is anomalous. The target is determined to be anomalous when the anomaly score $\mathcal{S}_\theta(\boldsymbol{x})$ exceeds a determined threshold ϕ:

$$\text{Decision} = \begin{cases} \text{Anomalous,} & \text{if } \mathcal{S}_\theta(\boldsymbol{x}) > \phi \\ \text{Normal,} & \text{otherwise} \end{cases}$$

The data used for this task comprises parts of ToyADMOS [1] and the MIMII Dataset [2] consisting of normal and anomalous operating sounds of two types of toys and four types of real machines. The anomalous sounds in these datasets were collected by deliberately damaging the target machines. The following six types of toy and real machines are used in this task: TOYCAR and TOYCONVEYOR from ToyADMOS, and FAN, PUMP, SLIDER, and VALVE from the MIMII Dataset. For simplifying the task, only the first channel of multi-channel recordings are provided; all recordings are regarded as single-channel recordings of a fixed microphone. Each recording is a single-channel approximately 10-sec length audio that includes both a target machine's operating sound and environmental noise. The sampling rate of all signals has been downsampled to 16 kHz. The *Machine ID* is defined as the identifier of each individual of the same type of machine, which in the training dataset can be of three or four and that of test dataset can be three. Development dataset includes (i) around 1,000 samples of normal

*During the team's participation, Phong was a visiting researcher at Mila. He was not involved in the organisation of the challenge, nor was he privy to any information that conferred undue advantage to his team. Phong took part in the challenge with the same information as all other participants in this challenge.

sounds for training and (ii) 100–200 samples each of normal and anomalous sounds for the test for each Machine Type and Machine ID. Evaluation dataset consists of around 400 test samples for each Machine Type and Machine ID, none of which have a condition label. The Machine IDs of the evaluation dataset are different from those of the development dataset. Thus, additional training dataset is also provided with around 1,000 normal samples for each Machine Type and Machine ID used in the evaluation dataset.

The task is evaluated using the area under the curve (AUC) of the receiver operating characteristic (ROC), and partial-AUC (pAUC). The pAUC is AUC calculated from a portion of the ROC curve over a pre-specified range of interest. Evaluation with pAUC is practically motivated: if an ASD system produces false alerts frequently, it causes more unnecessary expense. Therefore, it is especially important to increase the true-positive rate under low FPR conditions. In this task, $p = 0.1$ is used for this evaluation.

3. APPROACH

In this section, we first describe the members of our ensemble separately, and then describe how these are combined for the final scoring.

3.1. Autoencoders

We make use of the baseline auto-encoding model released by [2]. This is an MLP based autoencoder that compresses 5 frames of 128 mel-energies into an 8-dimensional space, with BatchNorm and ReLU non-linearities, and trained with Adam. Based on this model, we also add another similar autoencoding model, by only replacing linear layers with residual linear layers, keeping everything else the same. The normality signal is the mean squared reconstruction loss as is usual.

$$\mathcal{S}_{\text{ae}}(x) = ||x - \text{reconstruction}(x)||_2. \tag{1}$$

3.2. Classifiers

One of the key issues with practical anomaly detection for high-dimensional noisy data is that the models must be aware of what deviations constitute an indication of an anomalous situation in the context of interest [3]. For example, a density-based model might find a sudden loud voice in the background as a low-likely event if it has not encountered such events before. But a ideal, context-aware, anomaly detector for machine failure should know to attend solely to the sound of the machine, and raise an alarm only when the input sound indicates machine failure, regardless of variations in the background.

This suggests that providing discriminative information is likely to result in a predictive distribution upon a semantically-aware feature space. Since the provided dataset mixes in noise randomly with the machine sound, we expect that a classifier that discriminates between various machine types, or one that recognises a particular machine's sound from among others, would lead to more informative predictive distributions for anomaly detection [4]. An added advantage of a classifier that discriminates between various machines is that it can be reused for each of the machines separately without needing separate training. Based on these intuitions, we add four classifier models to our set of models: a linear 6-way classifier that predicts machine type from 32 5-frame mel-energies and a convolutional 6-way ResNet classifier that also predicts machine type from 128 5-frame mel-energies; a linear binary classifier that discriminates a specific machine from all others using 32 5-frame mel-energies and a convolutional ResNet binary classifier that does the same using 128 5-frame mel-energies.

We use 2 sets of tricks in our classification based models, which originate in developments in the literature about out-of-distribution/anomaly detection, and includes some of our own intuitions. These tricks are as follows.

1. Softmaxes tend to saturate, and as they do, it becomes harder to use the predictive confidence as a differential signal. To minimise this issue, we use a high L2 penalty on the weight matrix used to compute the logits. We also scale the logits, as suggested in [5], with a high temperature.
2. Existing works using predictive softmax confidences tends to use the maximum softmax value, regardless of the prediction this implies, which makes sense in a multi-class context. Since in our setting, we are interested in modelling a particular class at a time, we use the predictive confidence corresponding to the particular class of interest, for our models that are trained to categorise machine type.

$$\mathcal{S}_{\text{classifier}}(x) = -p(y = \text{machine}|x). \tag{2}$$

3.3. GMM on hand-crafted features

We learned a GMM on 34 short-term audio features using [6]. The time-domain features (features 1–3) are directly extracted from the raw audio waveform. The frequency-domain features (features 4–34, apart from the MFCCs) are based on Discrete Fourier Transforms (DFT). The cepstral domain (e.g. used by the MFCCs) results are acquired after applying the Inverse DFT on the logarithmic spectrum. In addition to these features, we also include mean, max, min, and covariance features from sliding windows of the raw waveform.

3.4. Phase-shift discrepancy

One of the members in our ensemble is a parameter-free self-predictive heuristic based on the *shift theorem* in signal processing. We select 3-second snippets from a given audio clip, and predict a Δ-time-shifted version through the equivalent phase-shift in the Fourier domain. This is then contrasted with the actual audio at the selected time in the future. More concretely, we first perform an FFT transform on the the clip $x(t)$,

$$x(f) = \text{FFT}[x(t)], \tag{3}$$

followed by a phase-shift corresponding to a time-shift of Δ,

$$x^{\Delta}(f) = e^{-j2\pi f\Delta}x(f), \tag{4}$$

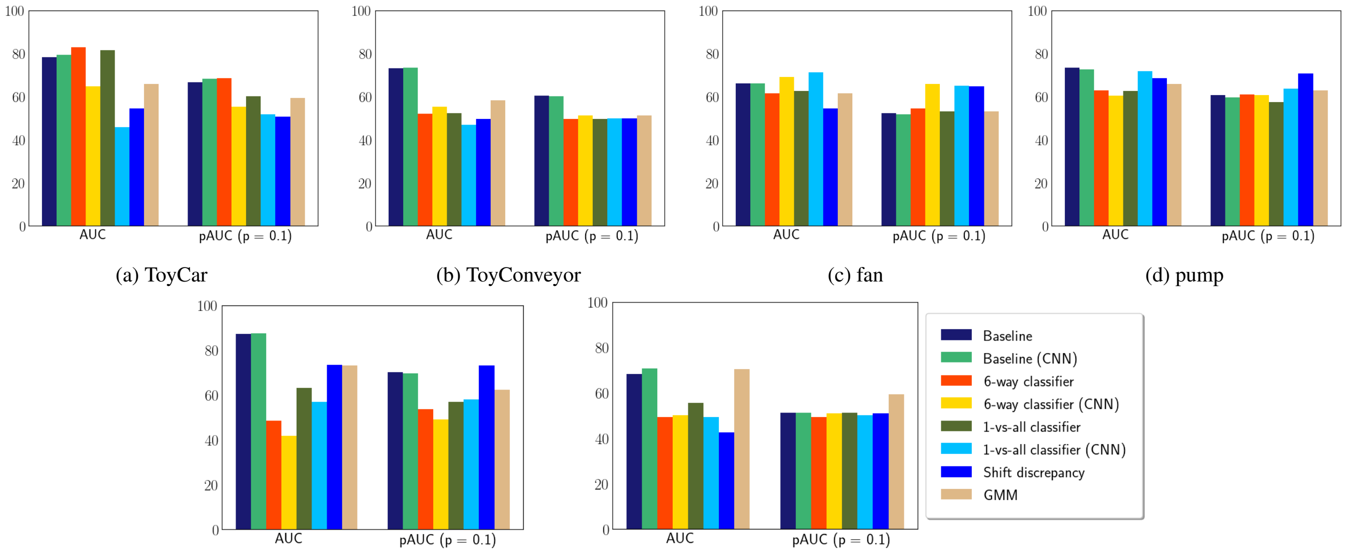
and then return to the time-domain via the inverse transform

$$\hat{x}(t - \Delta) = \text{Inverse-FFT}[x^{\Delta}(f)]. \tag{5}$$

The normality signal we use is the L1 difference in prediction,

$$\mathcal{S}_{\text{f}} = ||\hat{x}(t - \Delta) - x(t - \Delta)||_1. \tag{6}$$

While this method does not sidestep the potential issue of unexpected background noise being a confounding factor, we find it to help for certain machine types. In general, it is plausible that a machine obeying a certain periodicity might be diagnosed as being in an anomalous condition when the periodicity suddenly breaks.



(a) ToyCar

(b) ToyConveyor

(c) fan

(d) pump

(e) slider

(f) valve

Figure 1: Relative performances of our models on the development set. We find that different models perform with different strengths for different machines, and furthermore (not shown in figure) they perform with similar variance for different IDs.

3.5. Combining scores

While our classifier based scores are bounded in $[0, 1]$, being predictive confidences from a softmax, the other scores, while positive, are typically not bounded by 1. In order to bring these scores to a comparative range and spread, we learn a 2-hyperparameter non-linear transform, $\lambda \exp \mathcal{S}(x)/t$, where λ and t are set through tuning on the development set. To add the same degrees of freedom to the classifier-based methods, we also learn the same hyper-parameters for scores from these models. The final score is therefore computed as

$$\mathcal{S}_{\text{ensemble}} = \sum -\lambda_m \exp \frac{-\mathcal{S}(x)}{t_m}, \tag{7}$$

where m indexes a member of the ensemble.

4. EXPERIMENTS AND DEVELOPMENT RESULTS

4.1. Architecture and training details

4.1.1. Mel features

As in the provided baseline method, we used sliding windows on log-Mel spectrograms (with an energy-transform) of the audio as the input features for the models in our ensemble (with the exception of the phase-shift and GMM method). As in the provided baseline, we used a framesize of 1024, a hop size of 512 and 128 mel filters.

4.1.2. Autoencoders

The baseline model is a reimplementation of [2], where the encoder and decoder are both 4-layer MLPs with 128 dimensions, along with BatchNorm and ReLU activations. The latent dimension is 8. Training is done with Adam, using a learning rate of 1e-3 for 100 epochs.

The additional autoencoder we add to the ensemble is exactly the same, with the difference that the linear layers are replaced with residual linear blocks. For both models, we also preprocess the inputs by scaling and shifting with the mean and variance.

We train 3 runs of all autoencoder models, and average their scores (using the same λ and t parameters for all runs).

4.1.3. Classifiers

Our linear classifier consists of 4 linear residual blocks with 128 dimensions per linear layer, and BatchNorm+ReLU. We find that using more frames and higher mel-energies lead to improved performance, and so we use 256 frames from the mel-spectrogram per datapoint, and the last 32 energies of the 128 that are extracted. We add a coefficient of 0.5 for the L2 penalty on the weight matrix for the logits, and scale the logits by a temperature of 1e3 when computing the predictive confidence. We train with Adam for 50 epochs, with an initial learning rate of 1e-4, which we scale by 10 at the 10-th, 30-th, and 40-th iterations. We use 3 of

Our ResNet-based classifier has the following architecture: CONV → RESIDUALBLOCK(stride=2) → RESIDUALBLOCK(stride=2) → RESIDUALBLOCK → RESIDUALBLOCK (+BatchNorm+ReLU) → SPATIAL MEAN-POOL → LINEAR. All blocks use 16 channels. We find that using fewer frames and all mel-energies lead to improved performance for this model, and so we use 5 frames from the mel-spectrogram per datapoint. We add a coefficient of 0.5 for the L2 penalty on the weight matrix for the logits, and scale the logits by a temperature of 1e3 when computing the predictive confidence. We train with Adam for 10 epochs, with a learning rate of 1e-4. Compared to the linear classifier, we train for fewer epochs because every epoch now consists of far more datapoints, since we use 5 frames instead of 256 per spectrogram.

We use the same architectural and training details to train two sets of these classifiers. One set trains both for a 6-way classification task, to predict the machine types. The other set trains for a binary

Feature ID	Feature Name
1	Zero Crossing Rate
2	Energy
3	Entropy of Energy
4	Spectral Centroid
5	Spectral Spread
6	Spectral Entropy
7	Spectral Flux
8	Spectral Rolloff
9-21	MFCCs
22-33	Chroma Vector
34	Chroma Deviation
35-38	Mean, Max, Min, Covariance of the waveform

Table 1: Hand-crafted features used for the GMM

classification task, to predict a particular machine type against all others.

We train 3 runs of all classifier models, and average their scores (using the same λ and t parameters for all runs). Due to lack of time, in our submission, we only have one run for the convolutional one-vs-all classifier.

4.1.4. *GMM on short-term audio features*

Our GMM consists of 10 mixture components, with full covariance matrices. We use 38 features with 5 frame-stacking, such that each datapoint is a vector of length 190. In Table 1, we list the features. We used the Python library pyAudioAnalysis at https://github.com/tyiannak/pyAudioAnalysis for extracting these features.

4.2. Development set results

We find the best λ_m and t_m per model from searching over a grid of

$$\lambda_m = \{0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10, 50, 100, 1000, 10000\},$$

and

$$t_m = \{0, 0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 1000\}.$$

Since a complete grid search would involve searching over 156^8 settings per machine, we use a random grid search instead, testing 2000 random settings per machine.

In Figure 1, we show relative performances of all our models on all of the machines. We find that different models perform with different strengths for different machines. While the figure does not show it, we found that different models perform particularly differently for specific IDs in the dataset. This was another motivating reasons for us to adopt an ensemble approach.

In Figure 2, we show that we outperform the baseline using our ensemble of models.

5. CONCLUSION

We find that ensemble approaches can be effective at improving performance at practical anomaly detection tasks, since different models can excel at capturing different nuances of data. While this comes at the cost of a memory overhead, given that hardware memory costs continue to decrease and that the potential benefits involve early detection of machine failure and potentially replacing error-prone human labour, the overhead might be worth it in a cost-benefit analysis.

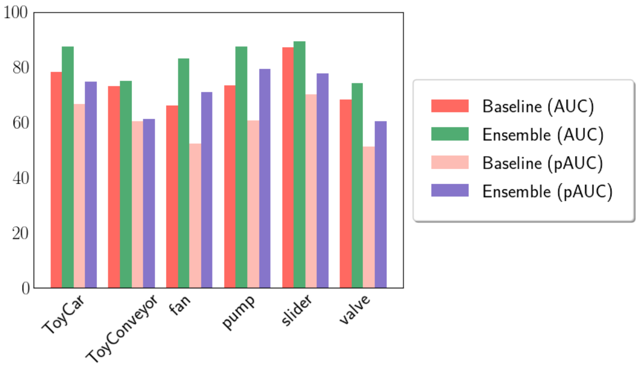


Figure 2: Performance of our ensemble on the development set. While we outperform the baseline for all machine types, we do not appreciably improve for ToyConveyor and slider.

6. REFERENCES

[1] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "Toyadmos: A dataset of miniature-machine operating sounds for anomalous sound detection," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 313–317.

[2] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," *arXiv preprint arXiv:1909.09347*, 2019.

[3] F. Ahmed and A. Courville, "Detecting semantic anomalies," in *Proceedings of 34th AAAI Conference on Artificial Intelligence*. AAAI, 2020.

[4] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proceedings of 5th International Conference on Learning Representations*, 2017.

[5] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proceedings of 6th International Conference on Learning Representations*, 2018.

[6] T. Giannakopoulos, "pyaudioanalysis: An open-source python library for audio signal analysis," *PloS one*, vol. 10, no. 12, 2015.