Proceedings of the Sixth International Conference on
Railway Technology: Research, Development and Maintenance
Edited by: J. Pombo
Civil-Comp Conferences, Volume 7, Paper 8.8
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.7.8.8

Detecting Anomalies Inside Rolling Stocks using Audio Streams and Deep Learning

S. Afanou

Centre d'Ingénierie du Matériel, SNCF Voyageurs
Le Mans, France

Abstract

This paper addresses the enhancement of passenger safety and comfort in public transport by automatically processing the audio streams from an anomaly detection system. The importance of anomaly detection has recently captured the attention of numerous researchers. Consequently, automated techniques, primarily based on artificial neural networks, are increasingly being adopted. This expansion is largely driven by the availability of large datasets and the use of graphics processing units, which facilitate the training of such models. Thus, these technologies have become the foundation for models that meet the railway industry's needs in ensuring the safety of passengers in train coaches. Although these models are promising and deliver high performance, they do so at the expense of significant system complexity and high computational costs. The results obtained confirm the benefits of using audio signals to detect unusual events and highlight some challenges in defining the appropriate set of model hyperparameters.

Keywords: anomaly detection, auto encoder, artificial intelligence, machine learning, audio streams.

1 Introduction

Audio anomaly detection presents a challenge for several reasons: Firstly, the definition of anomaly cases can vary depending on the circumstances and the use case. For example, the same activity might be considered normal or abnormal (such as a child's screams in a train's family area versus an adult's screams). Secondly, the accessibility and collection of abnormal data require significant effort due to the rarity of these events in real life.

For these reasons and others, anomaly detection is generally regarded as an unsupervised learning problem, aiming to train a model using only normal data. During the testing phase, events and activities that are poorly estimated by the model are considered anomalies. More specifically, in this work, we aim to detect anomaly cases using audio and video data through unsupervised approaches.

Generally, Let's consider the set U consisting of frames of unlabeled sound samples, noted as X_N, and under the hypothesis that the majority of these samples follow the normal distribution p_N; i.e., $(x \in X_N) \sim p_N$. Then, an anomaly detection (AD) problem is the process of identifying a sample test as either a normal sample that follows the distribution p_N, or as an anomaly:

$$\mathrm{AD}(\mathcal{F}(y)) \begin{cases} \text{Normal,} & \text{si } \mathcal{M}(\mathcal{F}(y), p_{\mathcal{N}}) \leq \tau \\ \text{Anomalie,} & \text{sinon} \end{cases}$$

Where M is the metric used for calculating the distance between a given instance and the distribution of normal data. F is a feature extractor that transforms the input data into a set of discriminative features. τ is the threshold from which a data point is considered as an anomaly.

The aim of this paper is to detect anomalous data from audio data using unsupervised approaches. Thus, this paper includes methods that address issues which can meet our objective. To this end, we introduce a railway-specific dataset that was generated artificially. We present in the following a description of the proposed approach for the generation of this dataset.

2 Methods

2.1 Dataset creation

An embedded railway environment is a unique setting where new acoustic challenges must be taken into account. This acoustic environment is extremely noisy and non-stationary. It consists of a blend of many sound sources originating from mechanical, electrical, and electronic subsystems operating concurrently, as well as noises made by passengers. In this context, we propose constructing a dedicated database by combining railway background noises with sounds of normal and abnormal events. These are detailed in the subsequent sections, which are then followed by an explanation of the mixing method we employ.

2.1.1 Railway background sounds

Railway background sounds were recorded during technical runs aboard various SNCF trains (suburban, regional, and high-speed) to create variability and to make our system less specific. The portable recording equipment was placed in the middle and at the end of the train. A total of six hours of background sounds were captured. The audio signal was recorded on a single 32-bit channel and was sampled at 44.1 kHz. These background sounds consist of a mix of engine noises, the friction of wheels on rails, the sound of air conditioning, commercial audio announcements, and so forth.

2.1.2 Normal, abnormal events and additional sounds

Four types of abnormal sound events (gunshots, screams, glass breaking, and spraying) along with two types of normal events (conversation and music) were selected. The sound samples for these categories were obtained from the Freesound website [1]. Each sample was reviewed to verify the audio content before being included in the final dataset. These sounds were recorded as a 32-bit mono channel signal with a sampling rate of 44.1 kHz.

In the context of a commercial train, other routine operational sounds such as buzzers and the opening and closing of doors may occur. We have enriched the dataset of normal event sounds with these additional sounds from another railway audio dataset.

2.1.3 Database samples generation process

Here, we explain the method used to blend background noises, abnormal sounds, and additional audio to create individual sequences for our new database. Each sound sequence produced is 10 seconds long, and we adhere to the following workflow to process each sequence in the dataset:

1. We randomly select a background sound from our collection. The audio sample's gain is randomly set between 0 and -10 dB to ensure variety while avoiding audio saturation.

2. We randomly choose between 0 to 3 abnormal events for detection. Their timing within the 10-second background is also determined at random. Overlapping of samples is permitted, and a random gain of between -5 and -15 dB is applied to each.

3. We then randomly determine the inclusion of normal events, with their temporal placement and gain chosen within the same range as abnormal events. Labels are generated concurrently with the integration of the samples

of the detectable events. The labeling uses One Hot encoding: each sequence that corresponds to a detectable event has a unique label tensor. Each element of this tensor starts at 0, except for the frames when a detectable event occurs, which are set to 1.

The distribution of durations for both abnormal events and additional sounds is detailed in Table 1

	Event type	Class	Duration
Train set	Normal events	Human conversation	3471 wav files
		Music	
Test set	Abnormal events	Gunshot	1157 wav files
		Scream	
		Glass break	
		Spray	
	Normal events	Human conversation	
		Music	

Table 1 : Dataset distribution

2.2 Machine learning models

2.2.1 ID-Conditioned Auto-Encoder for unsupervised anomaly detection

The first model we selected was chosen from the article "ID-Conditioned Auto-Encoder for unsupervised anomaly detection" [2]. The work proposed in this article constitutes an adaptation of the C2AE method (Class conditioned auto-encoder for open-set recognition) [3] for unsupervised anomaly detection.

Indeed, this issue can be viewed as a specific case of an open-set recognition problem, where unsupervised learning for the detection of abnormal cases is considered binary classification. In this scenario, there is only one class available during the learning process (the negative class, i.e., normal examples).

Open-set recognition refers to the challenge of identifying unknown classes during the test phase while maintaining the network's performance for the known classes (those seen in the learning phase). More simply put, the authors propose a label-conditioned Auto-Encoder consisting of:

- **Encoder E**: X → Z, which matches a feature vector X from the input space X to a code E(X) in the latent space Z.
- **Decoder D**: Z → X, which takes the code Z from the latent space Z and reconstructs it into an output D(Z) of the same size as the input vector from X.
- **Conditioning composed of two functions, H_γ and H_β**: Y → Z, which take an input label l (one-hot) from Y and map it to two vectors $H_\gamma(l)$ and $H_\beta(l)$ that are the same size as the codes derived from Z.

Architecture: The proposed model consists of an encoder E, a decoder D, and a conditioning mechanism H_γ, H_β, all of which are fully connected networks (FC nets).

The encoder features four dense blocks. Similarly, the decoder is composed of four dense blocks followed by a dense layer. Each dense block includes a sequence of operations: dense layer → batch normalization → ReLU activation.

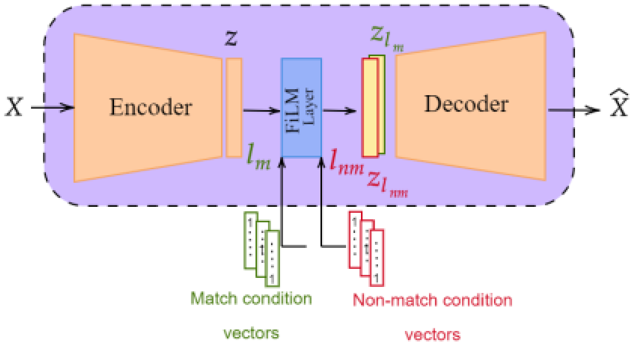


Fig1: IDCAE model architecture

As for the conditioning, it is implemented using the FiLM [4] technique. It takes the labels l as input and consists of a dense layer followed by a sigmoid activation function to produce $H_\gamma(l)$, and a separate dense layer (applied to the label) to produce $H_\beta(l)$.

2.2.1 Anomaly detection in raw audio using deep autoregressive networks

The second model we selected was chosen from the article "Anomaly detection in raw audio using deep autoregressive networks" [5]

The work presented in [5] is based on the use of an autoregressive generative model for unsupervised anomaly detection. The model employed is WaveNet [6], an approach initially proposed for speech generation that is based on PixelRNN [7] and PixelCNN [8], two methodologies introduced for generating high-resolution images.

Given the rarity of anomalous cases, the proposed network is trained solely on normal data. To detect anomalies, audio samples are fed into the trained network, which then generates a sequence corresponding to the input. If the audio sample is normal, the generator will successfully produce a signal similar to the input; however, if the sample is anomalous, the generator will fail to reconstruct it accurately. The mean squared error is used to calculate the distance between the generated signal and the actual signal: a small distance indicates a normal case, while a large distance suggests an anomaly.

Architecture: Like [8], the proposed model employs 2 stacks of 10 dilated causal convolution layers, totaling 20 layers. Residual blocks and skip connections are utilized throughout the network, with an exponential increase in the dilation rate in each stack. The number of filters used in the skip connections is 512, and in the residual blocks, it is 256.

Figure 2 provides an overview of a residual block (indicated by dashed lines) in WaveNet. The figure also illustrates the overall architecture of the network, highlighting multiple stacks of residual blocks and blocks with skip connections. This detailed representation helps in understanding the complex structure and the flow of data within the network.

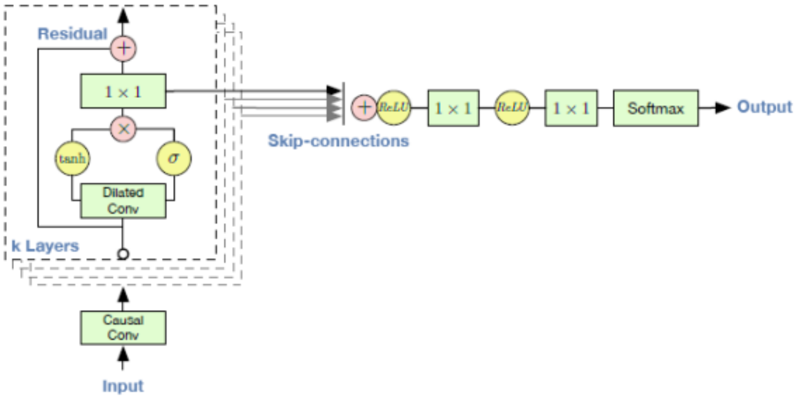


Fig 2 : Overview of the Residual Block and the Overall Architecture of WaveNet:

3 Results

3.1 Evaluation metrics

The evaluation metrics in this context are often the same as those used in classification problems.

3.1.1 Confusion matrix

To assess the prediction quality of a model, it is evaluated based on its predictions compared to the observed data.

By definition, a true positive (TP) is an outcome where the model correctly predicts the positive class. Similarly, a true negative (TN) is an outcome where the model correctly predicts the negative class. A false positive (FP) is an outcome where the model incorrectly predicts the positive class. Lastly, a false negative (FN) is an outcome where the model incorrectly predicts the negative class.

3.1.2 ROC curve and Area Under Curve (AUC)

ROC Curve: An ROC (receiver operating characteristic) curve is a graph that depicts the performance of a model by varying a certain threshold. This curve plots the true positive rate (TPR) against the false positive rate (FPR).

The true positive rate (TPR), also known as recall, is defined as follows:

$$TPR = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

The false positive rate (FPR) is defined as follows:

$$FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

AUC: This stands for the area under the ROC curve. It provides a comprehensive measure of performance across all possible classification thresholds. This metric has the advantage of being scale-invariant and independent of the classification threshold, making it an attractive and widely used measure for evaluating model performance.

pAUC: This refers to the partial AUC. It is an enhanced metric of AUC that focuses on maximizing the true positive rate while maintaining a low false positive rate.

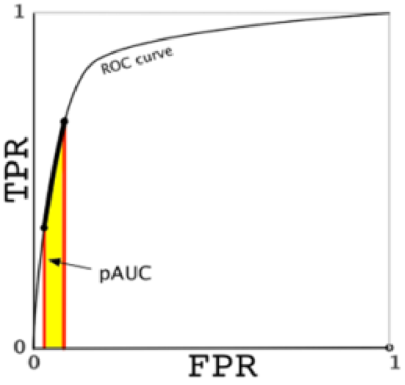


Fig3: pAUC

3.2 Models evaluation

We conducted a two-stage evaluation of our two models. The first stage involved benchmarking the models on a generalist dataset named DCASE using default settings. Subsequently, to save time, only the better-performing model was selected, trained, and optimized on the railway dataset.

3.2.1 Benchmarking on DCASE

The DCASE Challenge is an annual competition organized by several major universities around the world and sponsored by large companies such as Google, Amazon, and others. Each session of the challenge presents multiple tasks centered around audio, with training and testing data provided for each. At the end of each challenge, a workshop is organized to discuss the results and the models proposed.

DCASE Challenge 2020: In this section, we specifically focus on Task 2 of the DCASE 2020 challenge, which addresses the issue of "Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring."

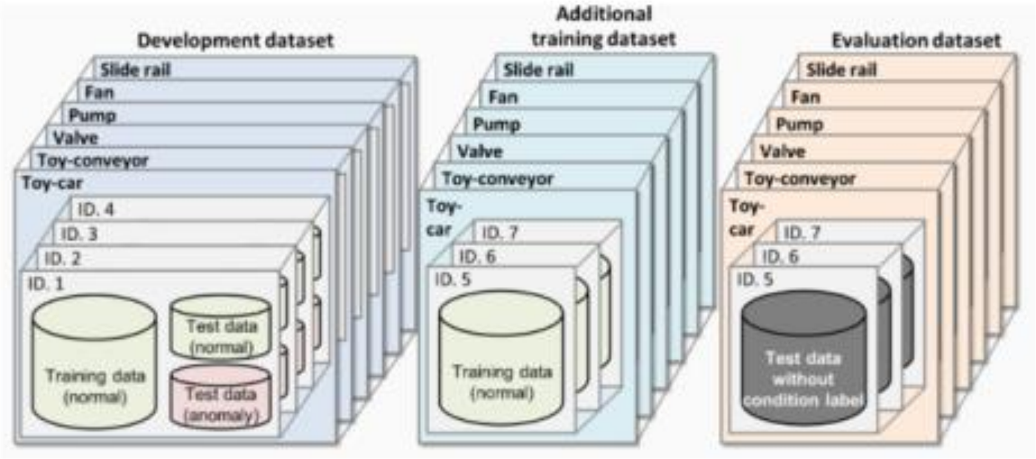


Fig 4 : DCASE dataset

As Figure 4 indicates, the provided data consists of three datasets: a development dataset, an additional training dataset, and a test dataset. These data are subsets of the ToyADMOS and MIMII datasets, comprising normal/abnormal sounds from six types of real machines and toys. Each recording is approximately a single audio channel of 10 seconds in length that includes both the operating noise of the target machine and ambient noise.

Each machine type corresponds to a label that identifies the machine, defining an individual of the same type of machine.

- **Development Dataset**: Consists of 3 to 4 machine identifiers (machine IDs). Each dataset corresponding to a machine ID comprises 100-200 samples of normal and anomaly data (training data and test data). The anomaly data are used only for testing the models' performance and are not used in the learning phase.
- **Additional Dataset**: These are additional data that can be used for model training. It contains 1000 normal samples.
- **Evaluation Dataset**: Contains the same types of machines provided in the development set. It includes 400 normal and abnormal samples for each machine ID. The IDs in this dataset are different from those in the development set.

Below are the results we obtained:

Params	**Metrics**	**IDCAE**	**WaveNet**

Default	AUC	**0,75**	0,72
	pAUC	**0,77**	0,69
Model complexity (parameters)		~400k	~2M
Running Time (50 epoch)		~20 min	~2 days
Prediction Time (10s wav file)		~21,4 ms	~2 sec

Table 2: Benchmarking performance.

Although the AUC and pAUC performance of the two models are very similar, the complexity of the WaveNet model appears to be significantly higher compared to IDCAE. This complexity results in heavier burdens during training and in making predictions with this model. It is important to remember that we are in an embedded context where the chosen model must be compatible with the constraints of real-time execution. Therefore, it is only natural that we opt for the IDCAE model for further experimentation.

3.2.2 Training and testing on railway dataset.

The IDCAE model previously identified was then trained, tested, and optimized on the synthetically generated railway dataset.

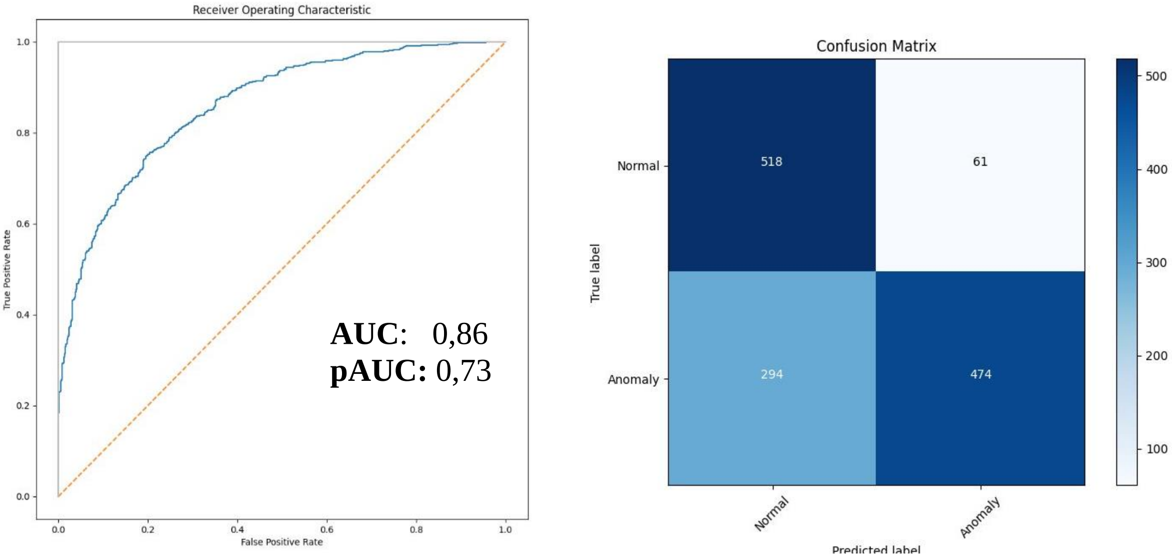
The first step in the optimization involved training the model with several sets of hyperparameters. Table 2 displays all the tested hyperparameters and highlights (in red) those that produced the best performance.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
hop_length	[**512**, 1024]	learning_rate	[**0.01**, 0.001, 0.1]
mels	[256, **128**, 64]	C	[10, **5**]
n_fft	[2048, 1024, **4096**]	alpha	[0.85, **0,75**]
epochs	[200, 50, 150, 80, 100, 300, **250**, 120]	nm_loss_name	[‘L1’, **‘L2’]**
encoder	[64, 32, 16] **[128, 64, 32, 32]** [128, 64, 32, 16, 8] [256, 128, 64, 32] [256, 128, 64, 32, 16] [64, 64, 32, 16]	decoder	[128, 128, 128, 64] [512, 512, 256, 128] [64, 128, 128, 128, 128] [256, 256, 128, 128] [128, 128, 64]

			[512, 256, 128, 128] [256, 128, 128, 128]

Table 2 : Hyperparameters tuning

This set of optimal hyperparameters enabled us to generate the ROC curve, AUC, pAUC, and the following confusion matrix:



AUC: 0,86
pAUC: 0,73

Fig 5 : ROC curve and confusion matrix

We can easily highlight the importance of hyperparameter optimization, which allowed us to achieve quite high performance. By analyzing the confusion matrix, we can observe a high number of false negatives (sounds predicted as normal when they are abnormal). We will now examine the impact of the decision threshold τ.

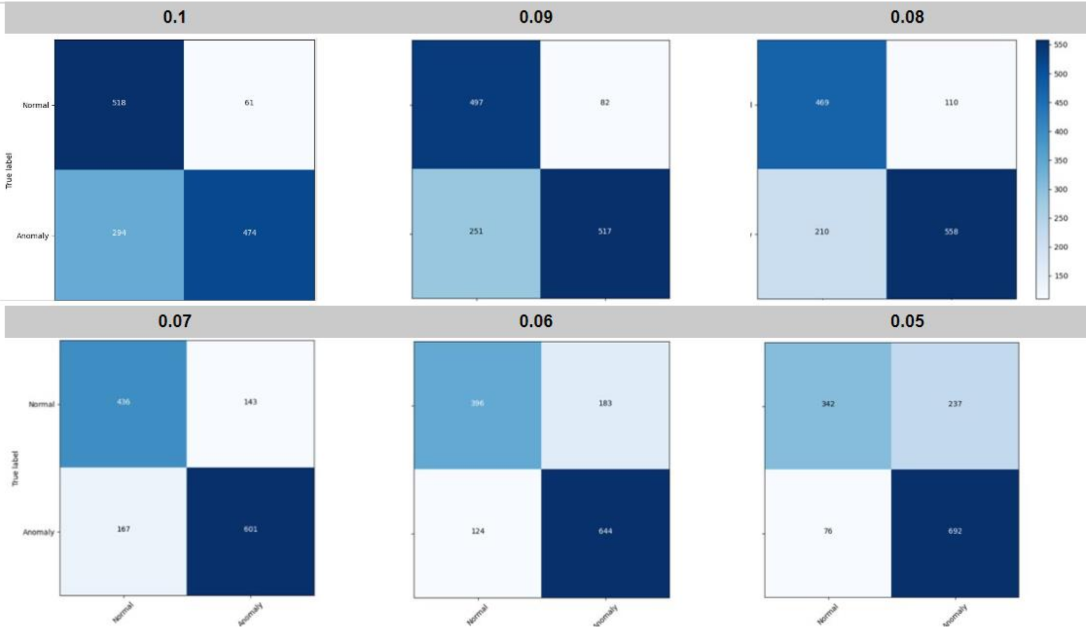


Fig 6 : Impact of the decision threshold on the confusion matrix

It becomes quite clear that lowering the decision threshold substantially reduces the number of false negatives. However, this reduction in false negatives is accompanied by an increase in false positives. Therefore, it is crucial to find a compromise on what error rate is acceptable in production when the system is mounted on a train.

Finally, it is possible to analyze which classes of anomalies are most prevalent among the false negatives produced by the model.

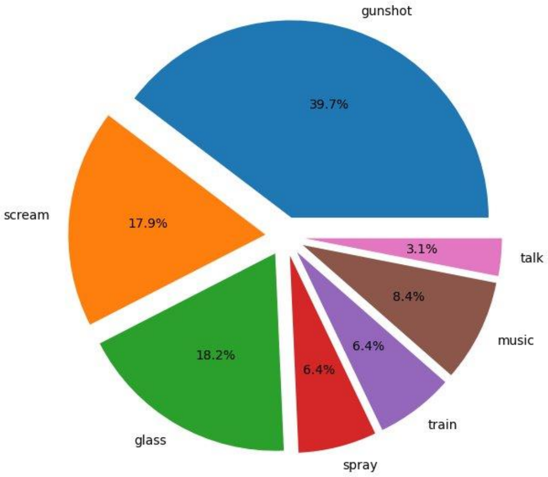


Fig 7 : Distribution of classes among false negatives

It appears that very impulsive noises such as screams and gunshots are most represented among the false negatives. This distribution can be interpreted by the fact that impulsive noises are present in the normal sounds on a train (electrical relays, wheel screeching, etc.). For screams, one could suggest, for example, confusion between children's screams, which are normal, and adult screams, which are not.

4 Conclusions and Contributions

This work has confirmed that anomaly detection systems can add significant value to the railway industry, particularly in identifying any unusual and rare events. The use case for anomaly detection is very useful in assisting railway companies with their mission to ensure the safety and security of their public transport users.

Models based on autoencoders have proven particularly effective in distinguishing between normal and abnormal events, achieving an AUC score of 0.86. Our study also highlighted the importance of selecting the decision threshold τ to find an acceptable compromise on the error rate.

Further modeling work and enriching the dataset are the next steps to further minimize the model's errors.

References

[1] Font, F., Roma, G., Serra, X.: Freesound technical demo. In: ACM Int. Conf. on

[2] Multimedia. pp. 411–412. Barcelona, Spain (October, 21 2013)Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, Toshiki Nakamura, Yuki Nikaido,Ryo Tanabe, Harsh Purohit, Kaori Suefusa, Takashi Endo, Masahiro Yasuda, et al.Description and discussion on dcase2020 challenge task2 : Unsupervised anomalous sound detection for machine condition monitoring. arXiv preprint arXiv :2006.05822,

[3] Poojan Oza and Vishal M Patel. C2ae : Class conditioned auto-encoder for open-set recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2307–2316, 2019.

[4] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film : Visual reasoning with a general conditioning layer. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[5] Ellen Rushe and Brian Mac Namee. Anomaly detection in raw audio using deep autoregressive networks. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3597–3601. IEEE, 2019.

[6] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet :A generative model for raw audio. arXiv preprint arXiv :1609.03499, 2016.

[7] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In International Conference on Machine Learning, pages 1747–1756. PMLR,

[8] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. arXiv preprint arXiv :1606.05328, 2016.