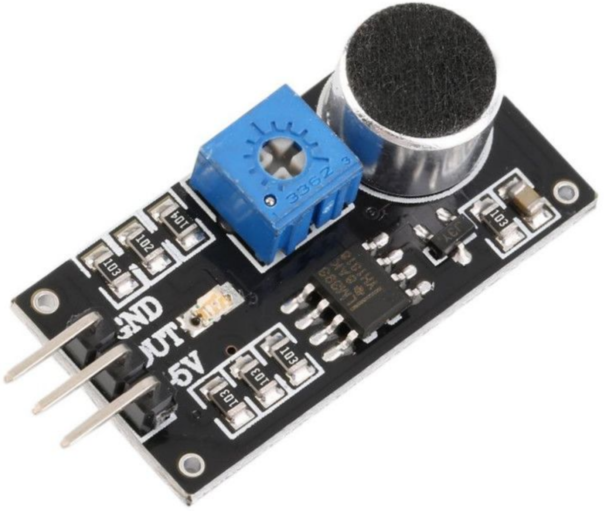
Smart Audio Sensors: Anomalous Detection for IIoT

Eric Leclair

What is a Smart Audio Sensor?

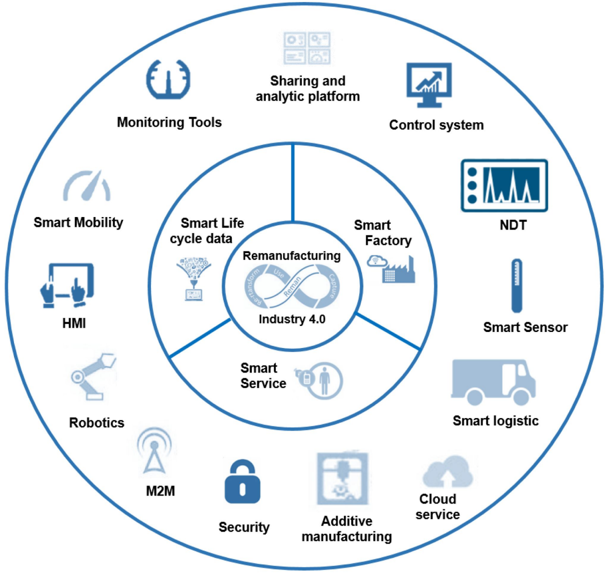
- Everyday devices are becoming more powerful and highly connected everyday, allowing us to push more computation towards the IoT edge

- A smart audio sensor could be any form of edge-connected IoT device that combines an audio sensor with local computation to record and preprocess data



Why Smart Audio Sensors?

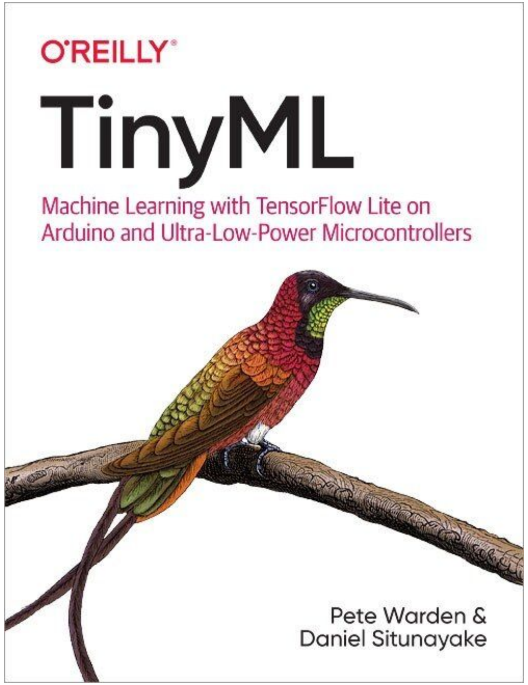
- Greater sense of automation in Industry 4.0
- Early detection of problems in machinery
- Highly applicable, and easy to install



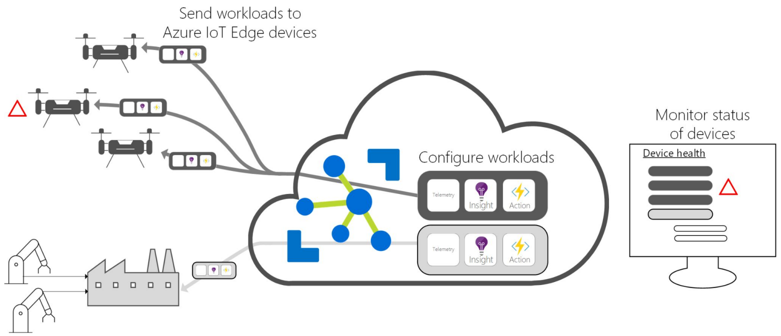
How can we Make Smart Audio Sensors?

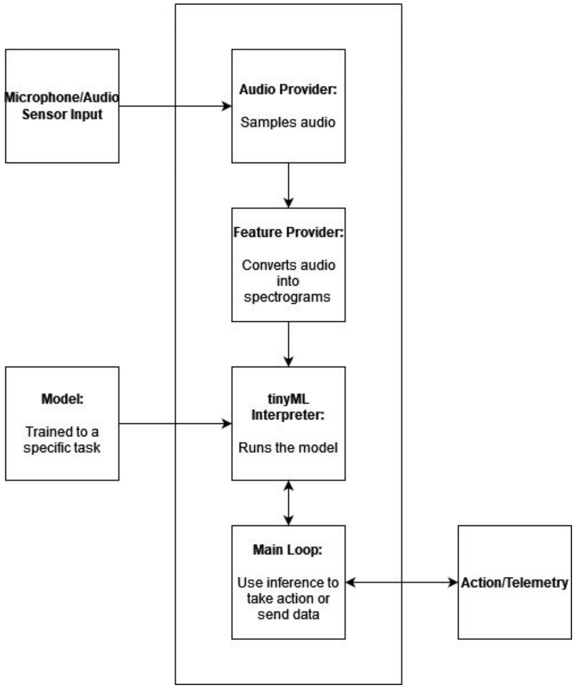
- Through tinyML we can achieve distributed intelligence, by bringing machine learning to microcontrollers and other IoT edge devices

- We can then devise a new framework for IoT edge devices, implementing machine learning for audio signals

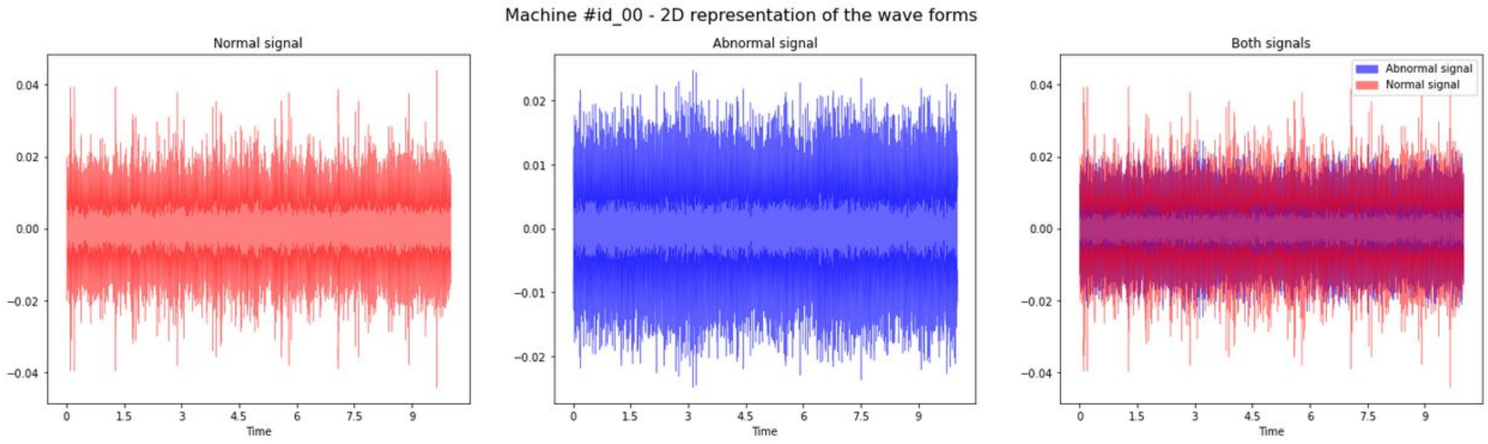


The Smart Audio Sensor Framework

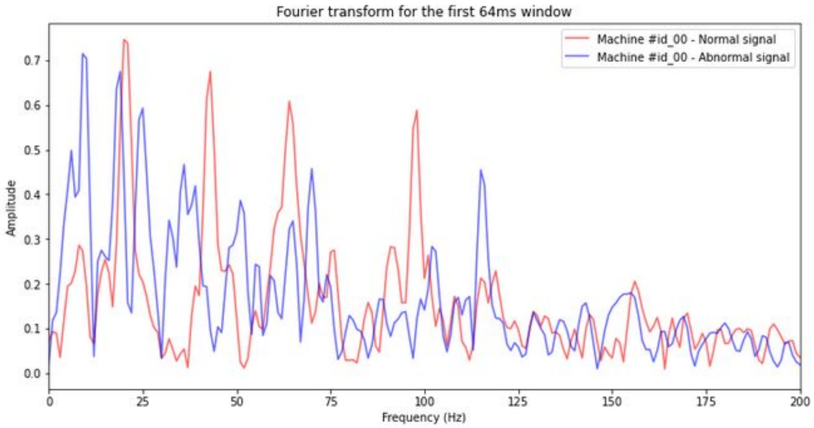


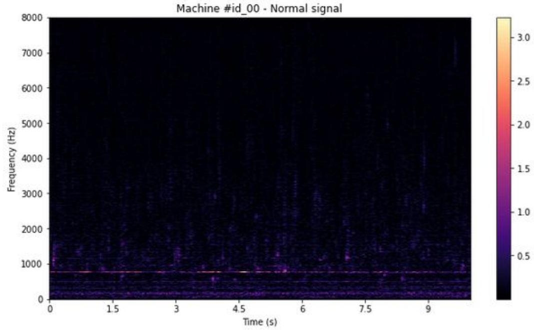


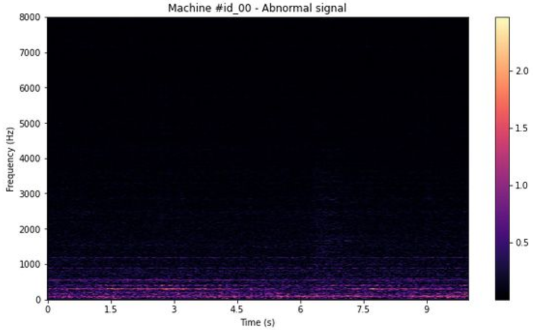
Visualizing Audio



Using Fourier Transforms





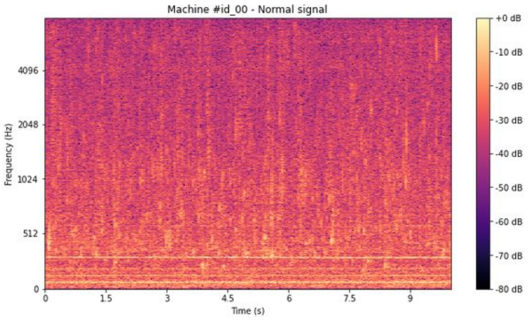


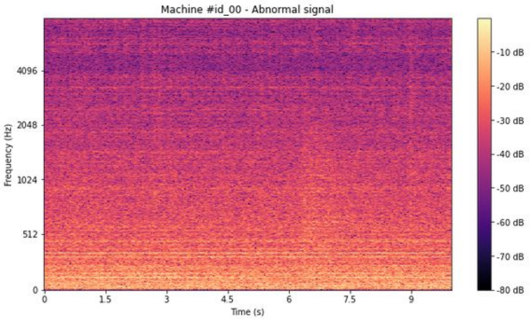
How do Humans Perceive Audio?

- Humans perceive audio logarithmically, hence why audio is measured in decibels (dB)

$$P_{dB} = 10log_{10}\left(\frac{P}{P_{ref}}\right)$$

- We can take a logarithmic transformation of the power spectrum, to help extrapolate audio features and carriers



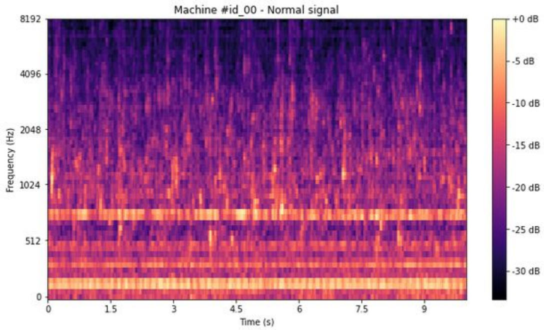


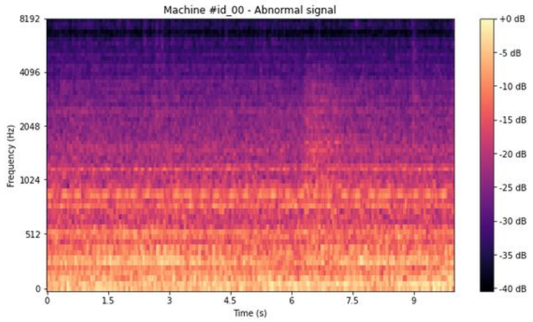
The Mel Scale

- However decibels are not good enough to model human hearing, resulting in the Mel Scale:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

- The Mel Scale aims to correct for our perception of pitch, in equal intervals, however, this can also help a computer dilate the spectrogram for analysis

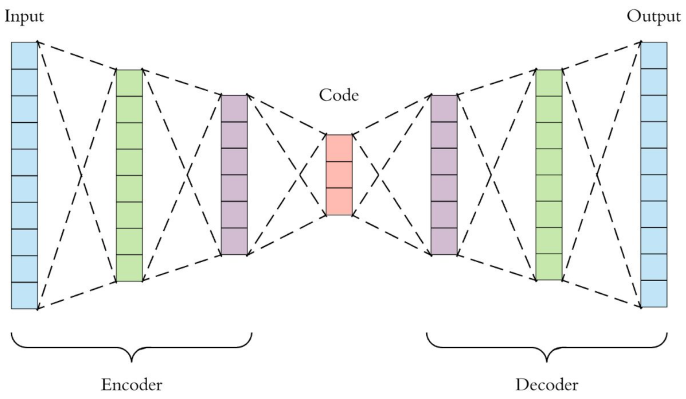




Autoencoder Topology

- Autoencoders are a unique form of artificial neural networks, that contain the same amount of inputs and outputs

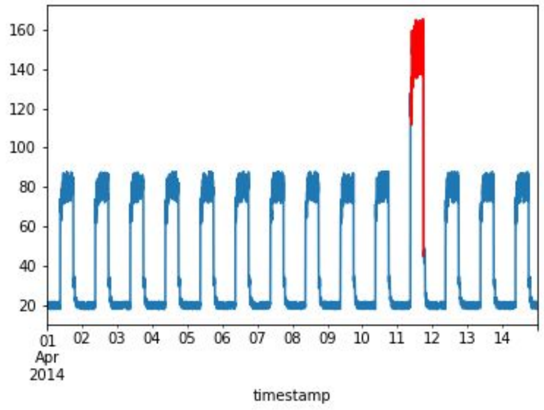
- Through an Encoder and Decoder stage we can ‘compress’ information into a smaller dimension and train it on its ability to reconstruct the signal



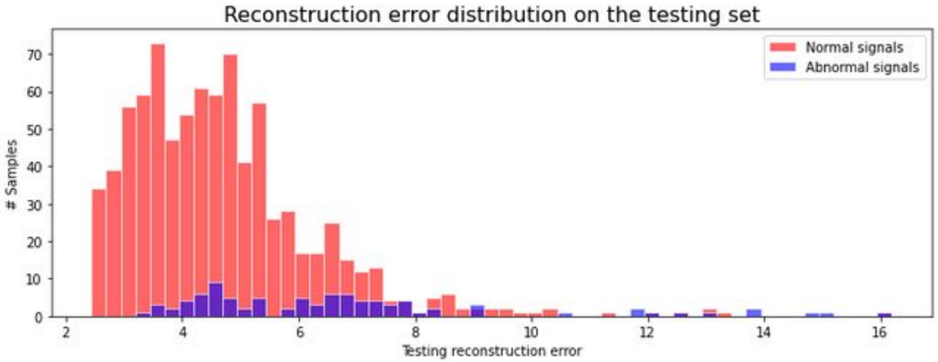
Using Autoencoders for Anomalous Detection

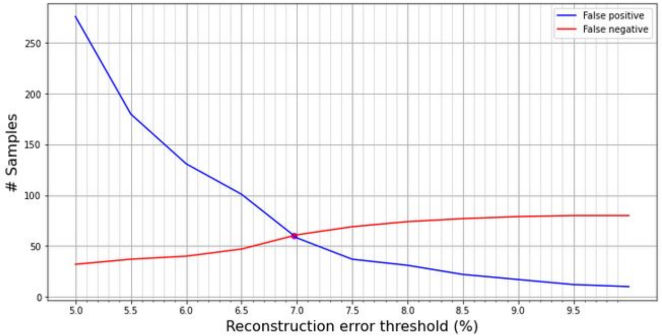
- Through autoencoders we can reconstruct our signal from a lower dimensional information based on the features of our input

- This can then be compared with the actual data stream, highlighting any anomalies



Anomalous Detection in an Audio Sample





Simple Autoencoder vs. Convolutional Autoencoder

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 320)]             0

 dense (Dense)               (None, 128)               41088

 dense_1 (Dense)             (None, 128)               16512

 dense_2 (Dense)             (None, 16)                2064

 dense_3 (Dense)             (None, 128)               2176

 dense_4 (Dense)             (None, 128)               16512

 dense_5 (Dense)             (None, 320)               41280

=================================================================
Total params: 119,632
Trainable params: 119,632
Non-trainable params: 0
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_7 (InputLayer)        [(None, 64, 32, 1)]       0

 conv2d_16 (Conv2D)          (None, 64, 32, 32)        320

 max_pooling2d_10 (MaxPoolin  (None, 32, 16, 32)       0
 g2D)

 conv2d_17 (Conv2D)          (None, 32, 16, 32)        9248

 max_pooling2d_11 (MaxPoolin  (None, 16, 8, 32)        0
 g2D)

 conv2d_transpose_10 (Conv2D  (None, 32, 16, 32)       9248
 Transpose)

 conv2d_transpose_11 (Conv2D  (None, 64, 32, 32)       9248
 Transpose)

 conv2d_18 (Conv2D)          (None, 64, 32, 1)         289

=================================================================
Total params: 28,353
Trainable params: 28,353
Non-trainable params: 0
```