

エッジAIを使ったパイプ異常検出

Spresense と Neural Network Console による自己診断

ソニーセミコンダクタソリューションズ（株）

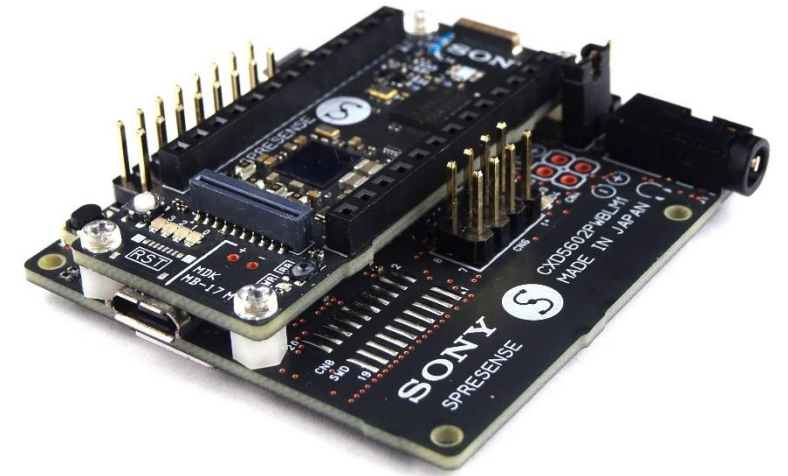
太田 義則





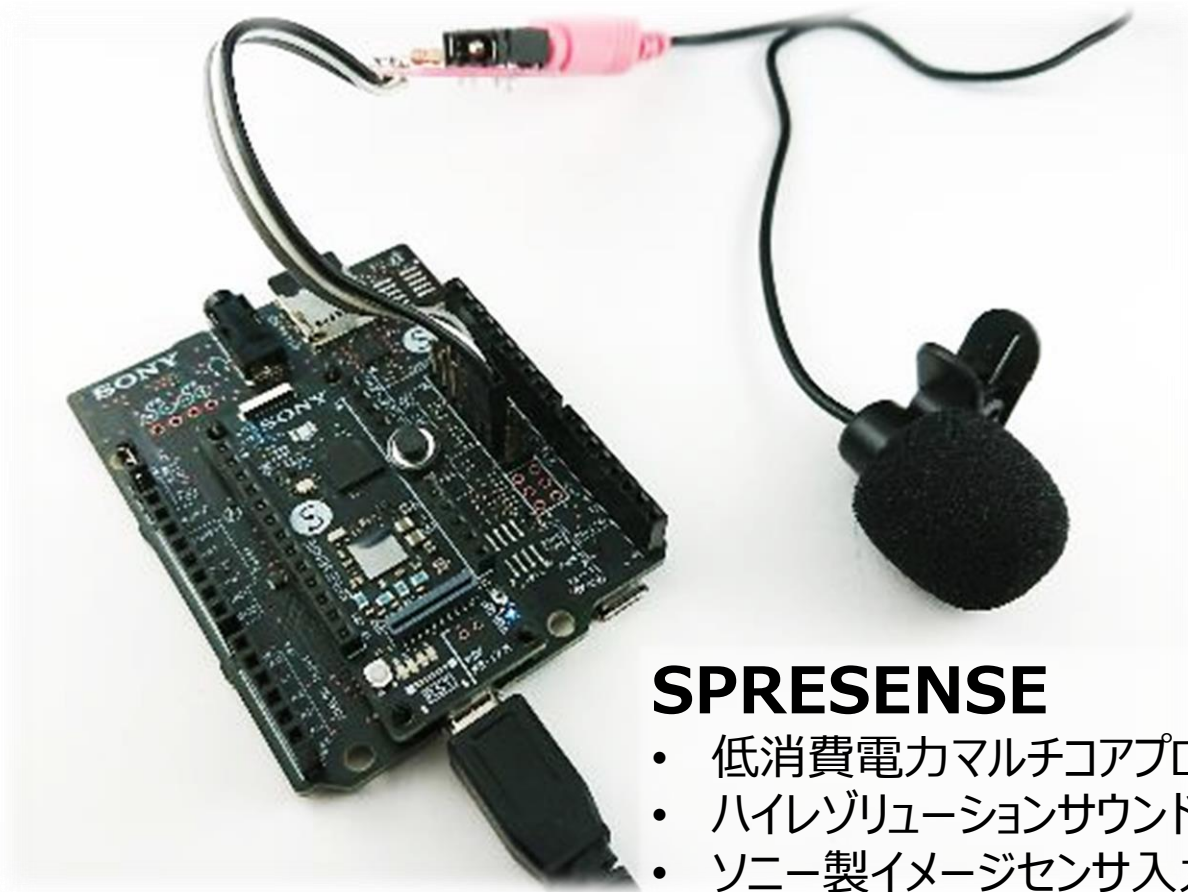
これからお話する内容

- **Spresense**と**Neural Network Console**の紹介
- 空気パイプのモデルの考察
- パイプの異常を検知するための**AI**
- 実際のモデルで異常検知を検証する



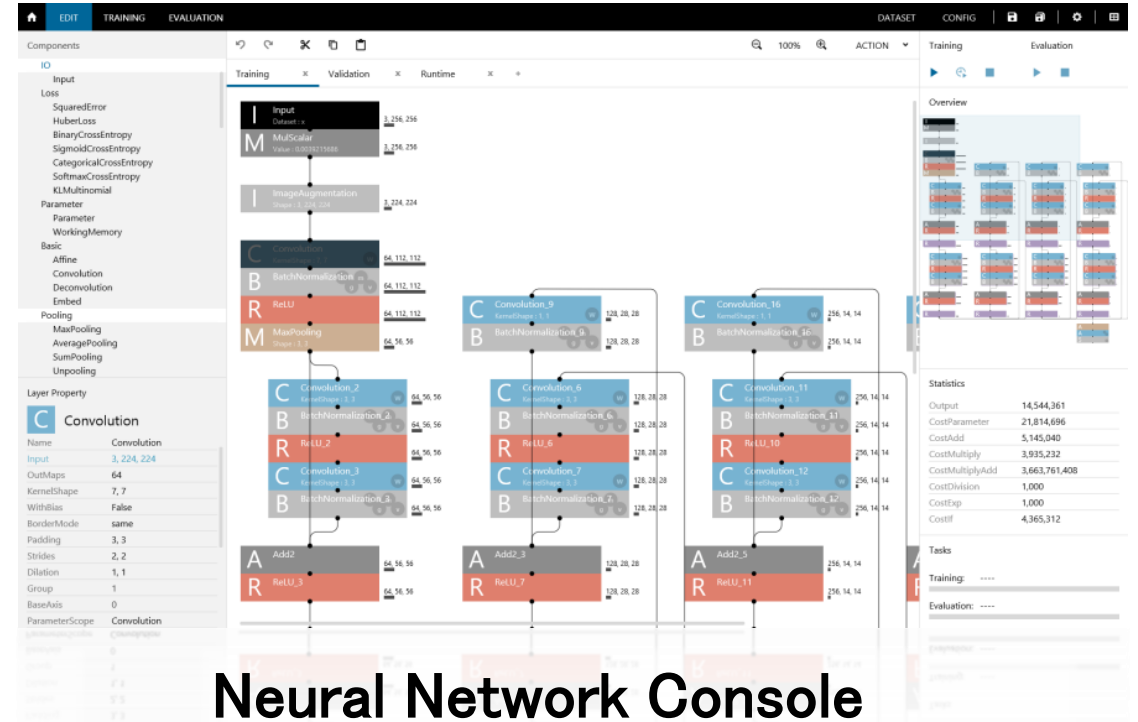


SPRESENSE と Neural Network Consoleの紹介



SPRESENSE

- 低消費電力マルチコアプロセッサ
- ハイレゾリューションサウンド入出力
- ソニー製イメージセンサ入力

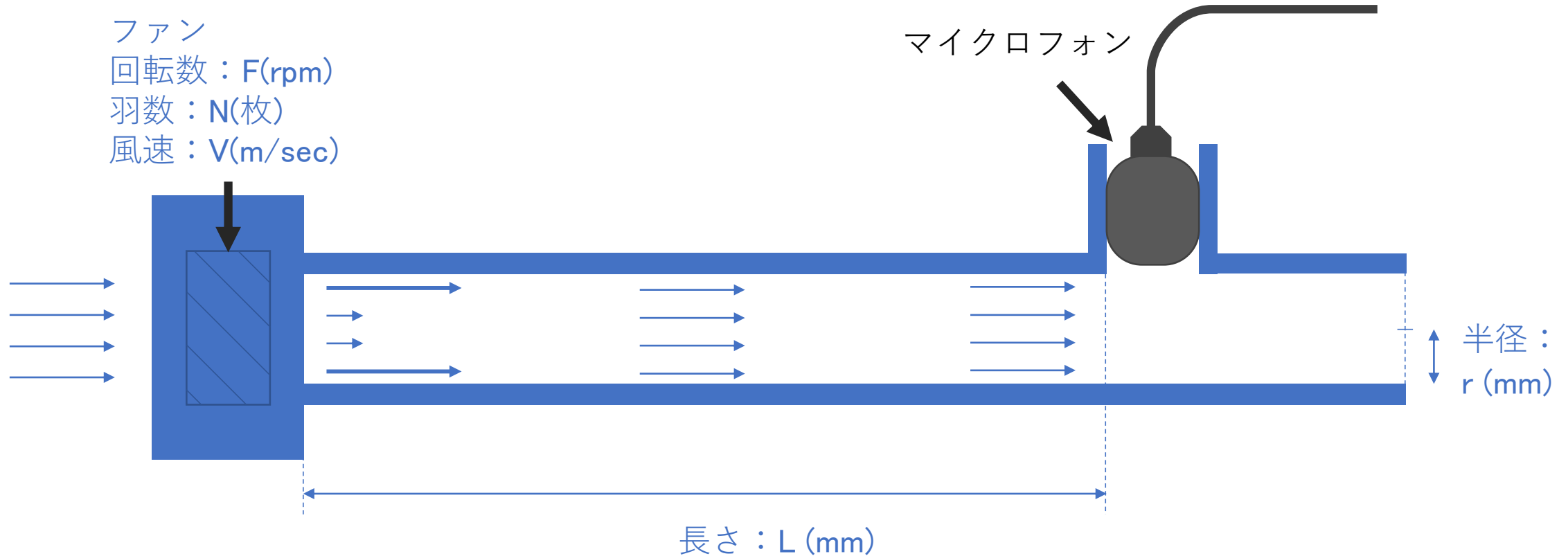


Neural Network Console

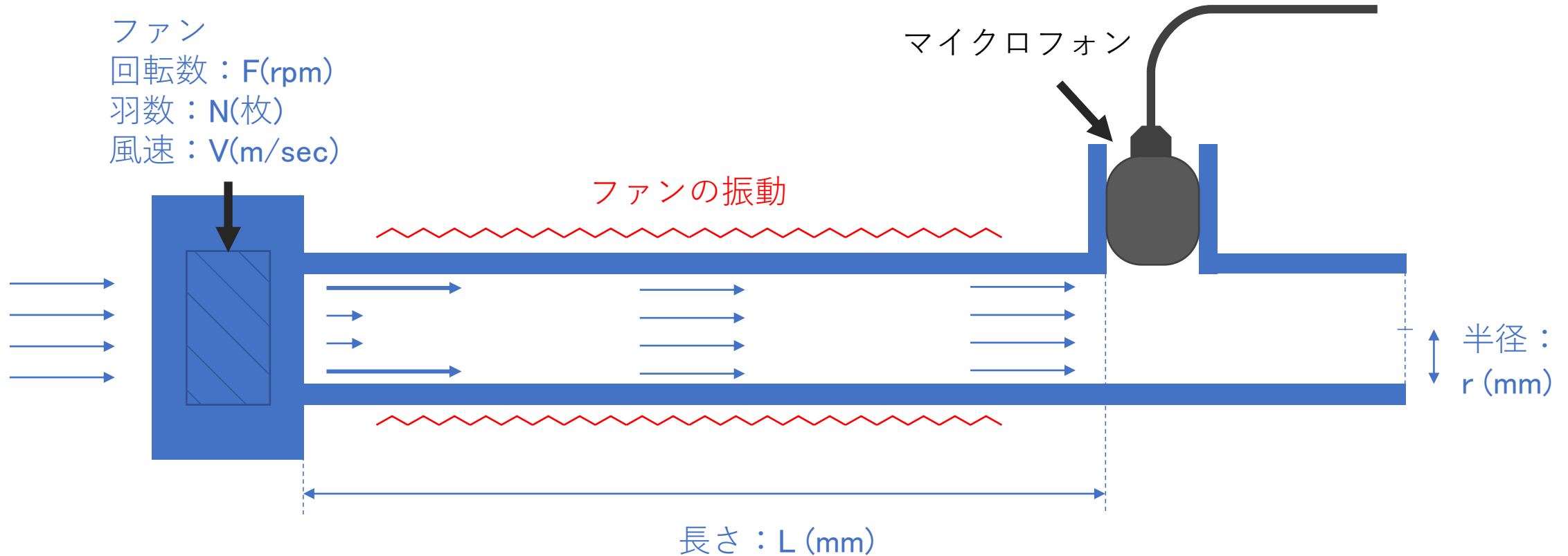
- コーディング不要のディープラーニングツール
- ドラッグ＆ドロップによる簡単編集
- 組込み向けライブラリを出力



空気パイプモデルの考察

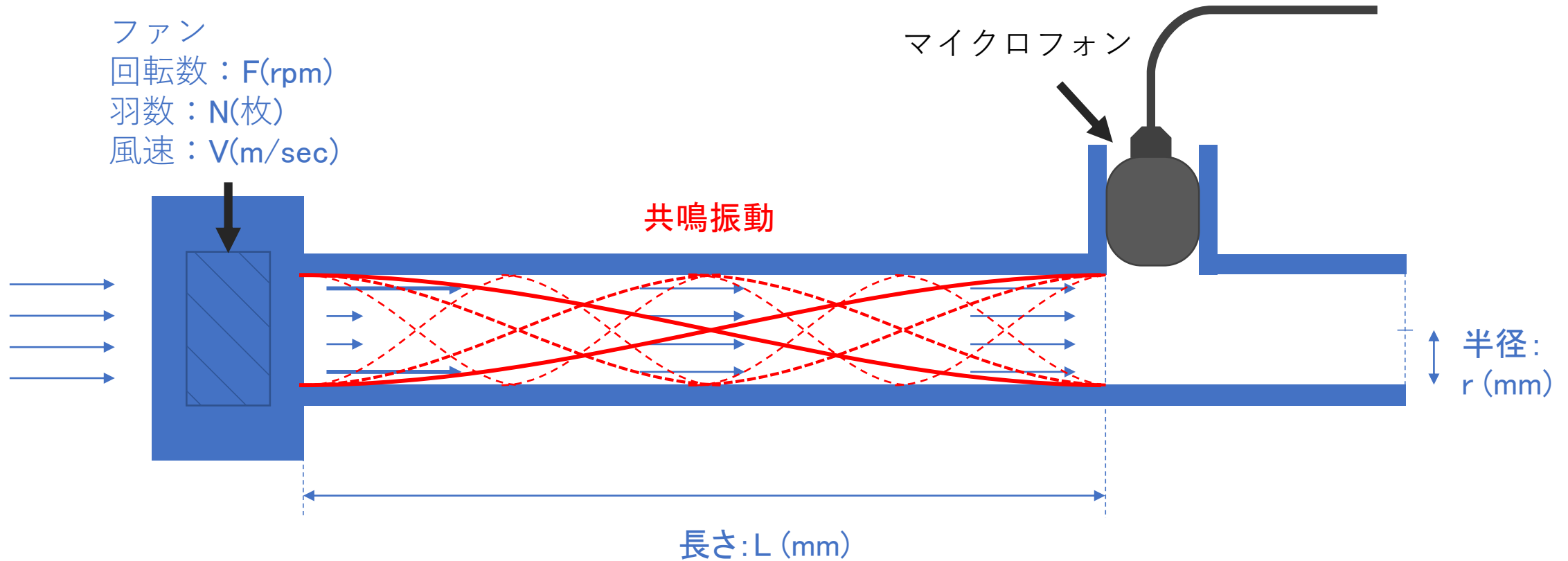


空気パイプモデルの考察



マイクが拾う音の要因：ファンの振動 ($F \times N$)

空気パイプモデルの考察

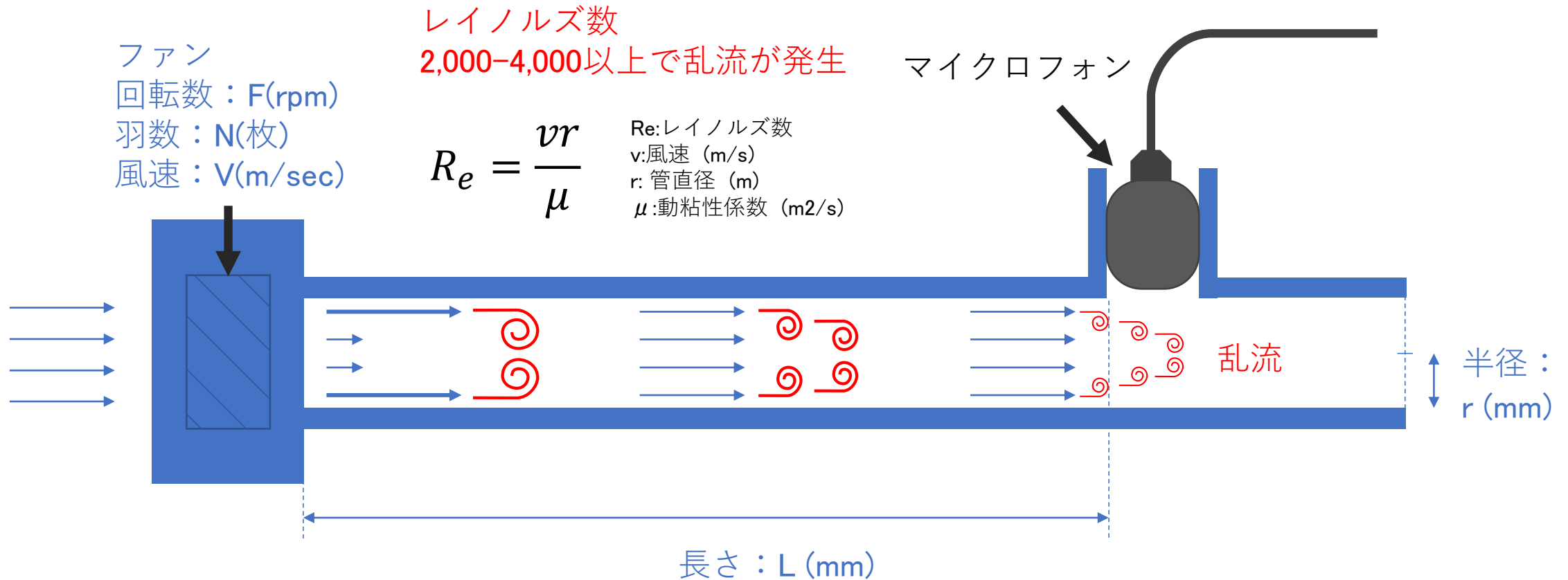


c:音速 340m/s, n: 倍数 1~6

マイクが拾う音の要因：共鳴振動($c/2L \times n$)



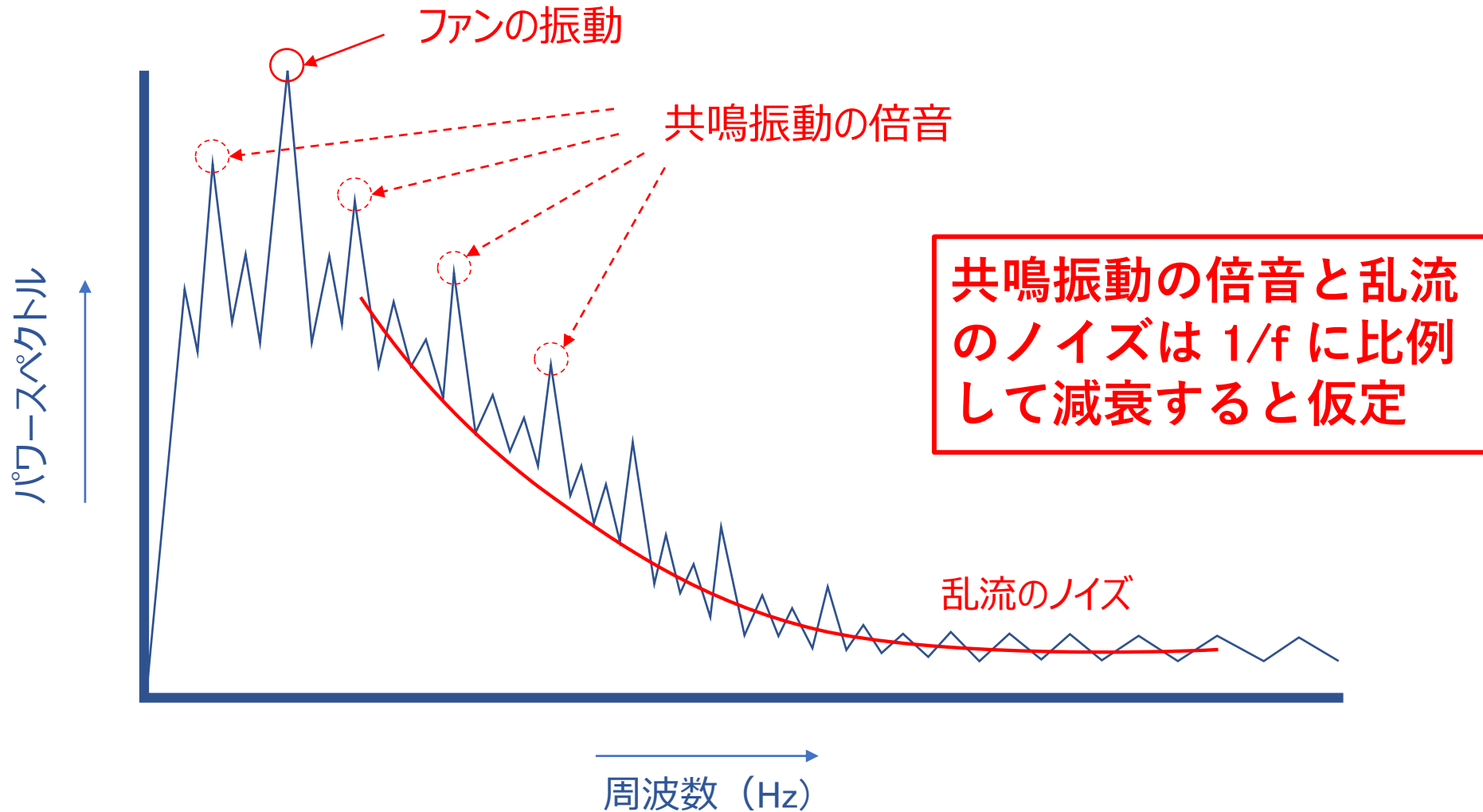
空気パイプモデルの考察



マイクが拾う音の要因：乱流



空気パイプモデルの考察



空気パイプモデルの考察

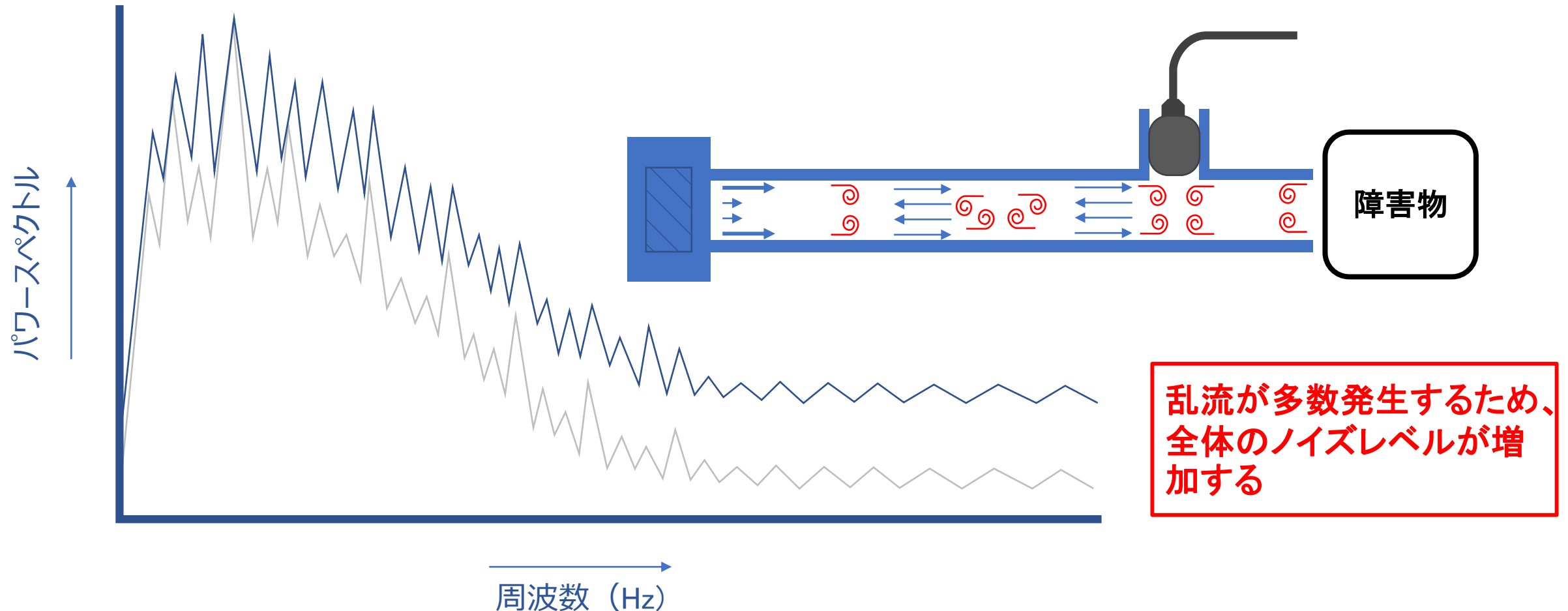
障害のケース1：ファンの入口が詰まる





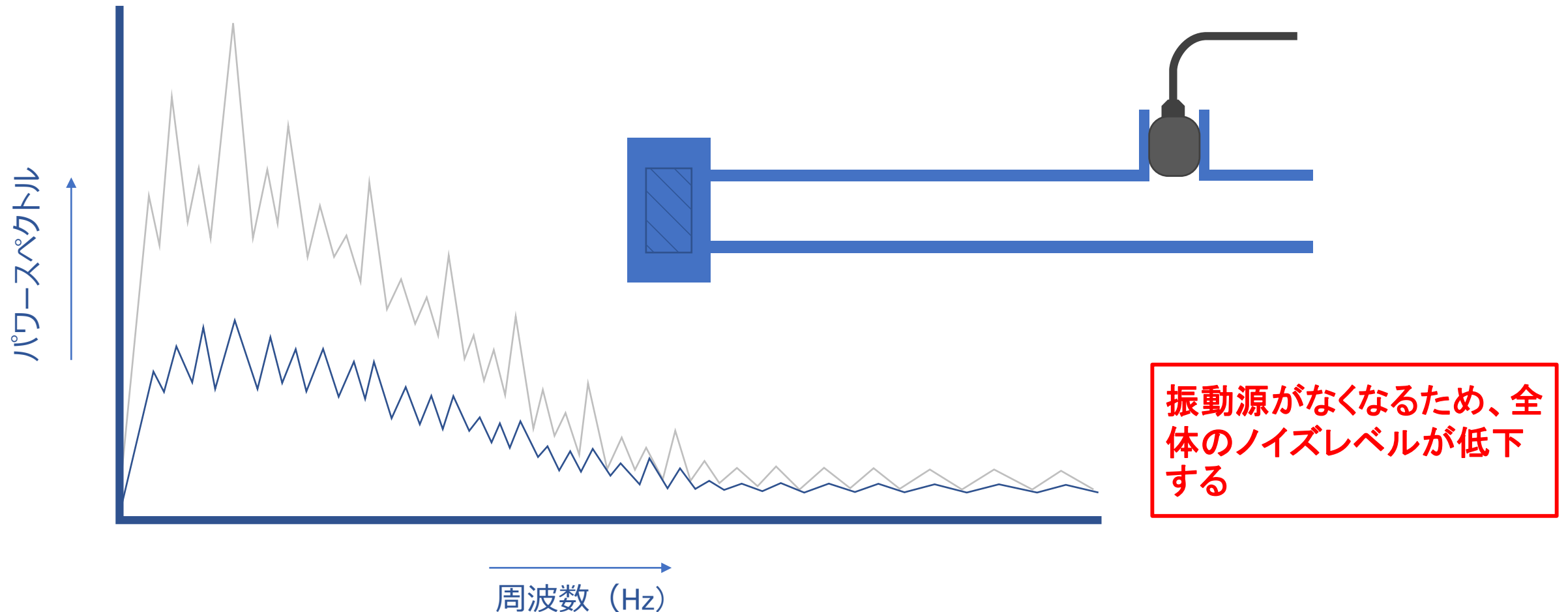
空気パイプモデルの考察

障害のケース2:パイプの出口が詰まる



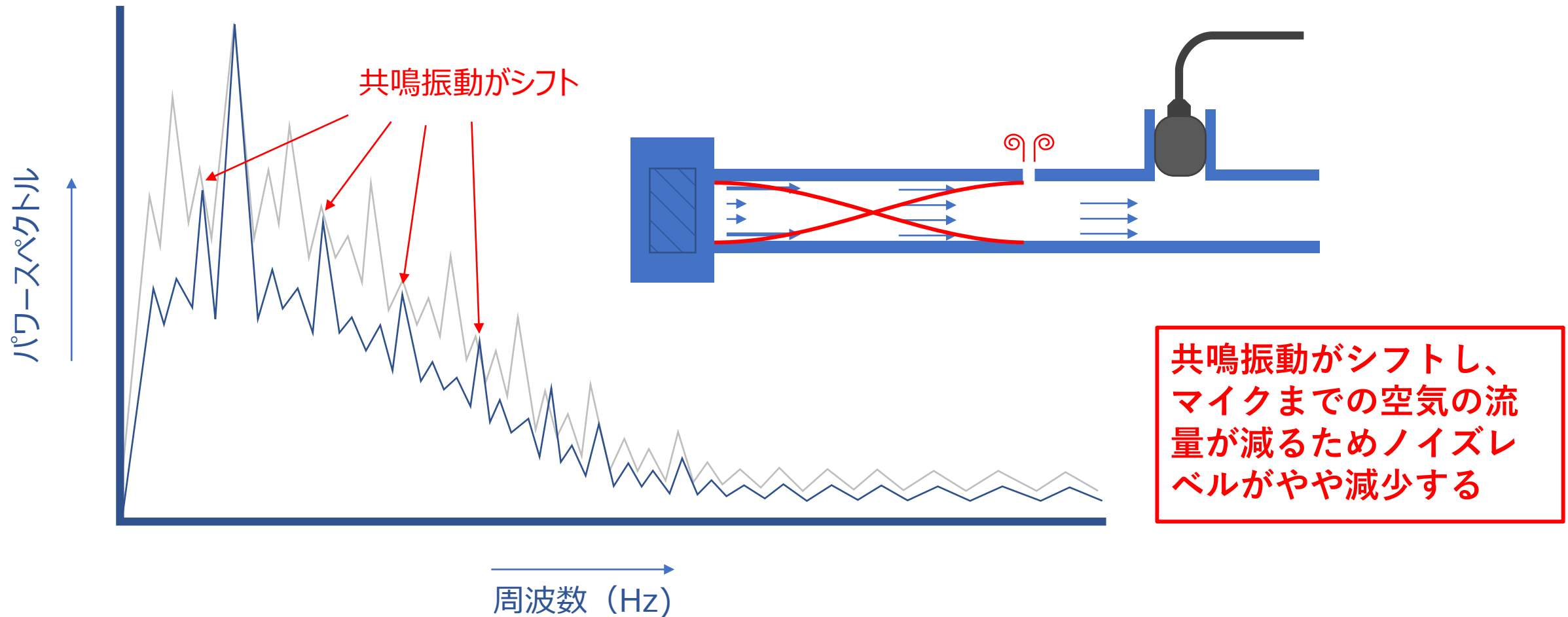
空気パイプモデルの考察

障害のケース3:ファンが止まる



空気パイプモデルの考察

障害のケース4：パイプに穴が開く





パイプの異常を検出するAI

- パイプの異常は常に起こるものではない
- 異常時の雑音パターンは障害物やクラックの状態で変化する
- 複合的な異常雑音の場合はパターンが複雑に絡み合う
- 乱流が発生させる雑音は複雑で発生パターンが特定できない
- 平常時のパイプ雑音は定常的であり安定している

正常時のパイプ雑音を学習、
それから外れる雑音を異常と見なす

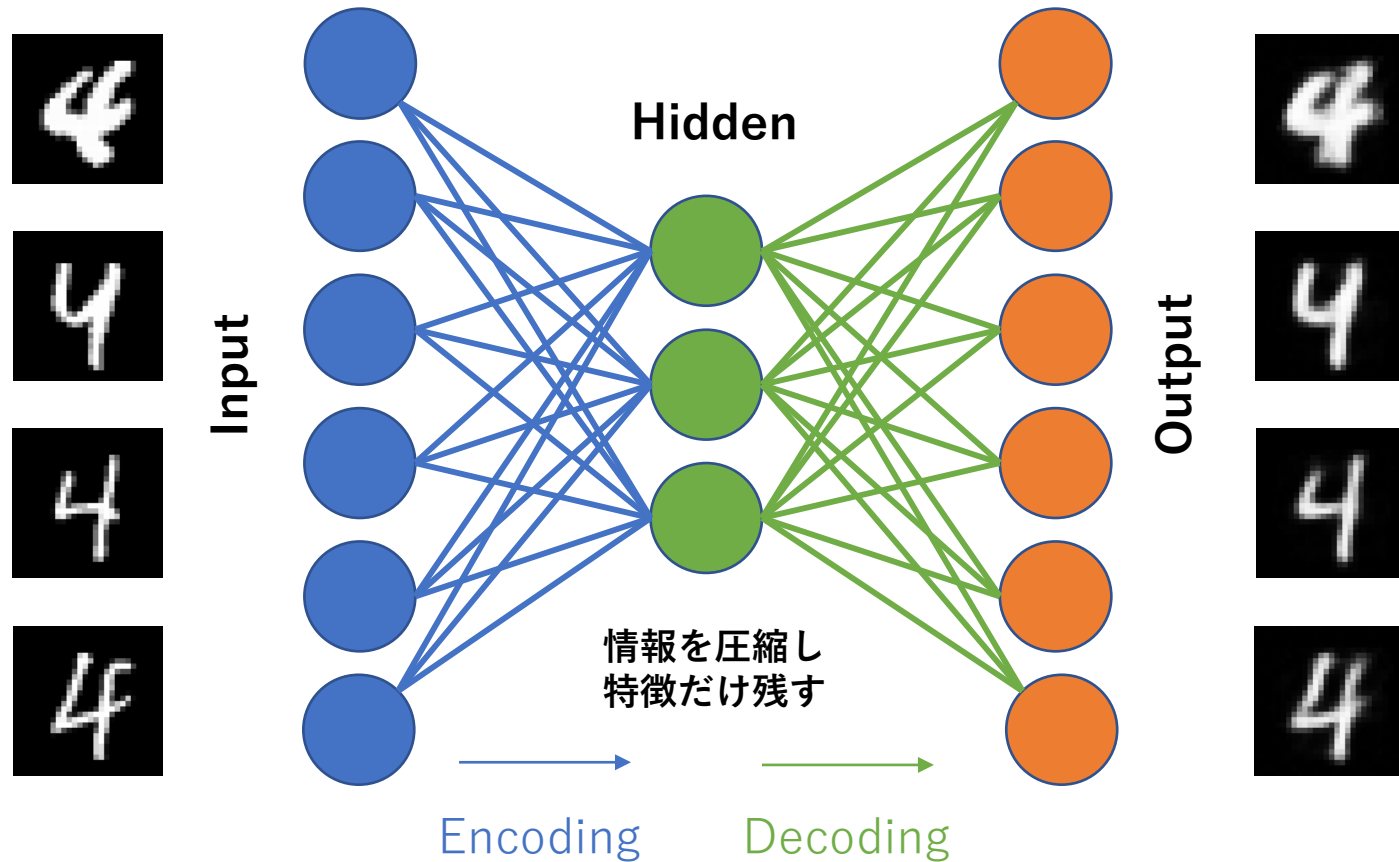


教師なし学習

パイプの異常を検出するAI

教師なし学習を実現するオートエンコーダ（学習時）

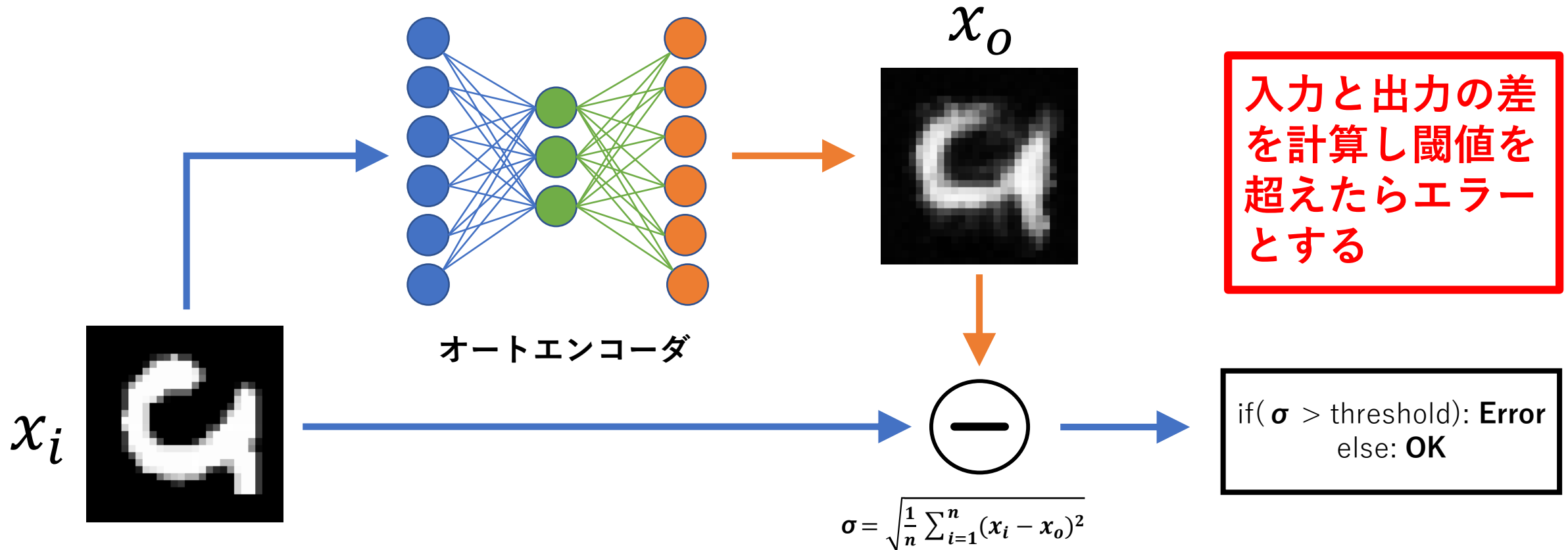
正解データ
を用意して
学習



特徴を残し
た形を出力
する

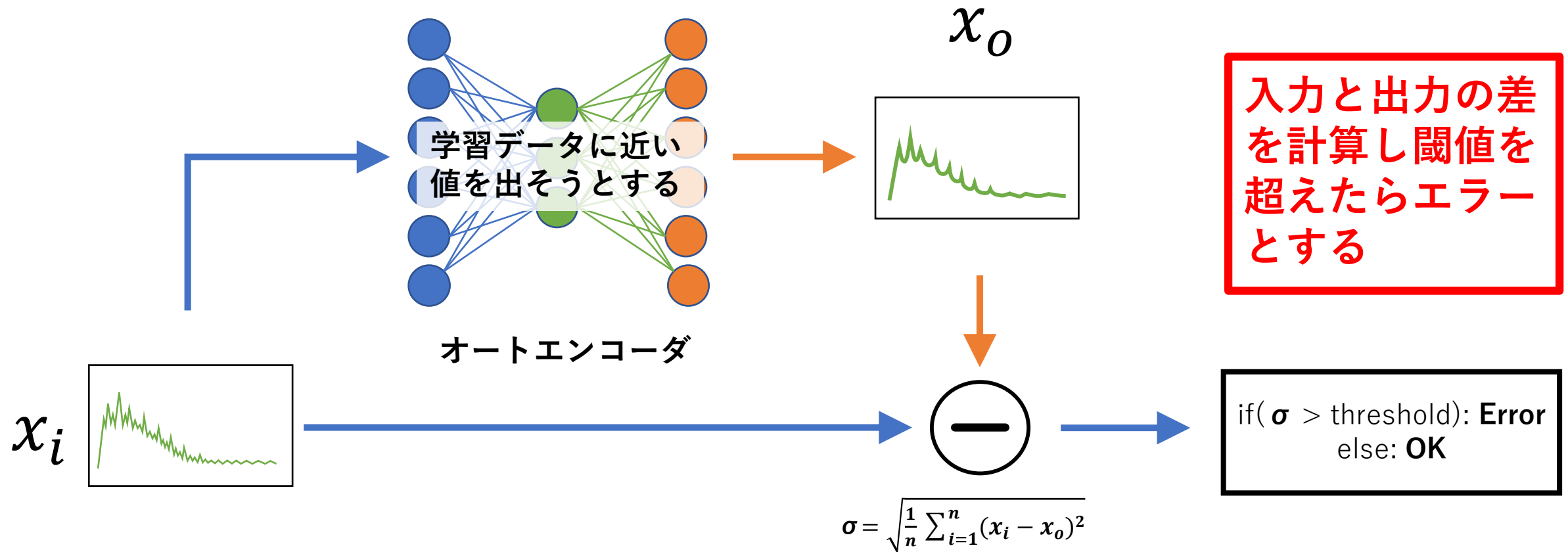
パイプの異常を検出するAI

異常データを入力すると、オートエンコーダは正常データに近い値を出そうとするため、誤差が拡散する⇒異常を検出しやすい



パイプの異常を検出するAI

音響データの周波数スペクトルデータを使って異常を検出



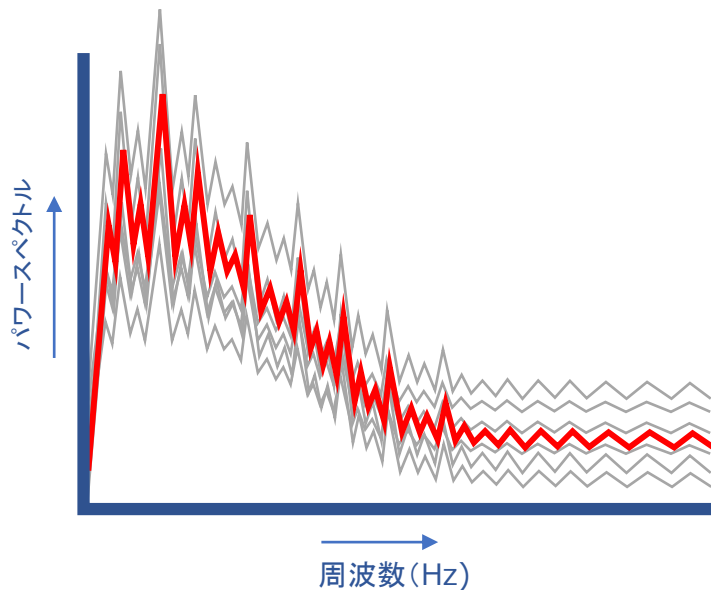


パイプの異常を検出するAI

平均値との差分ではなくオートエンコーダを使う理由

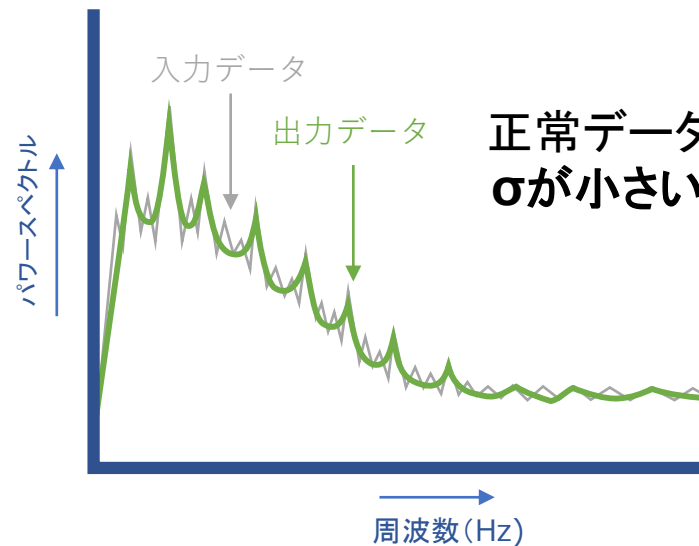
平均との差分

同じ正常データでも、変動が激しいため平均値の差分では正常データと異常データの判別が難しい

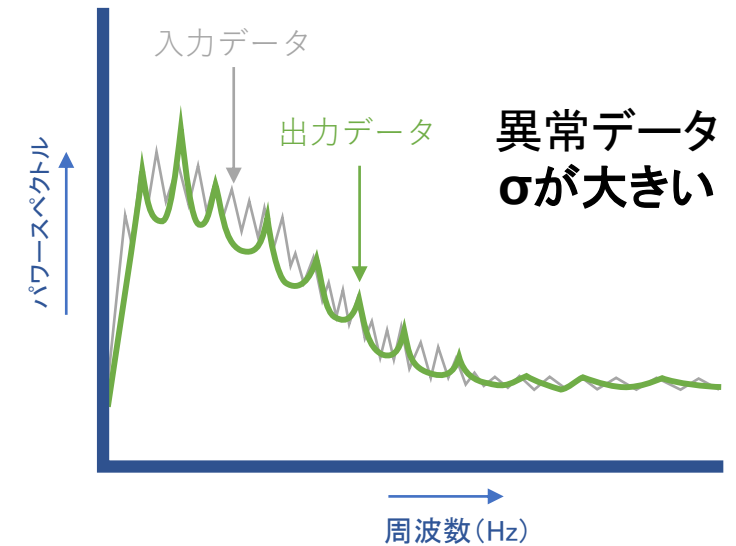


オートエンコーダによる検出

オートエンコーダは入力データに対し、記憶した特徴を強調するように出力する



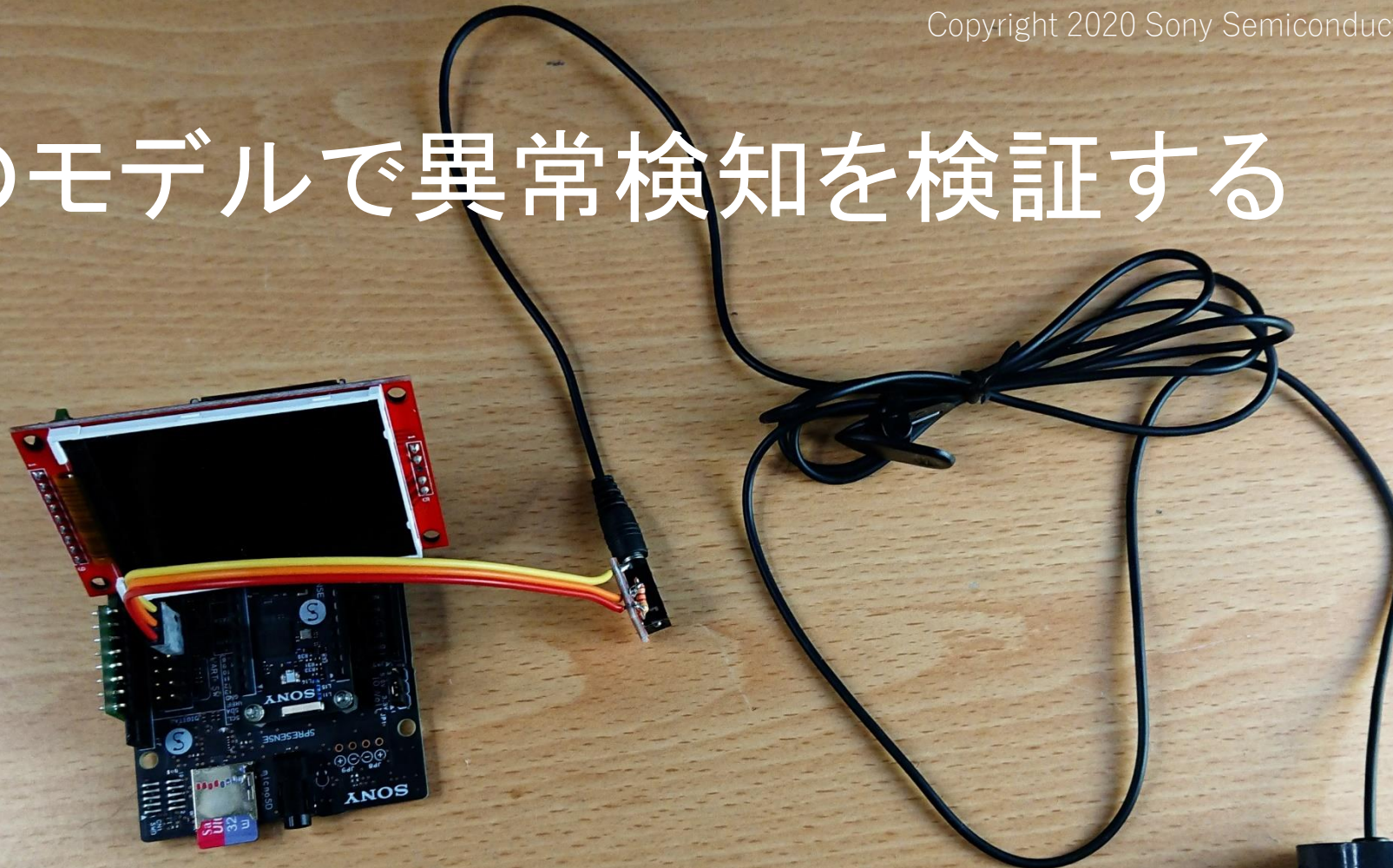
異常データの場合、特徴を強調するように出力するため、入力との違いが顕著となる





実際のモデルで異常検知を検証する

ファン
回転数: 8000(rpm)
羽数: 5 (枚)
風速: 10 (m/s)



アナログ
マイクروفोन



長さ: 300 (mm)

半径:
6.5 (mm)





実際のモデルで異常検知を検証する

検証用システムの概要

ファン

回転数: 133(Hz)

羽数: 5 (枚)

風速: 10 (m/s)

レイノルズ数

$$Re = 10 \times 6.5 \times 10^{-3} \times 2 / (1.512 \times 10^{-5}) = 8,598$$

空気の動粘性係数(20°C): 1.512×10^{-5}

ファンの振動数

$$F = 133 \text{ (Hz)} \times 5 \text{ (枚)} = 665 \text{ Hz}$$

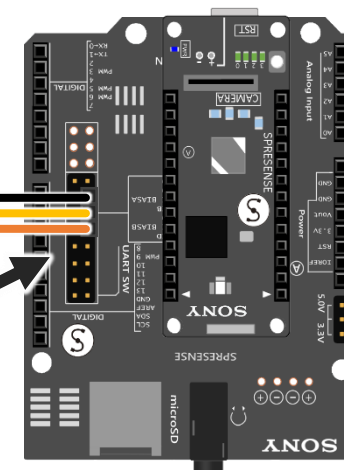
SPRESENSE
マイク入力

半径:
6.5 (mm)

長さ: 300 (mm)

共鳴振動数

$$\lambda = 340 \text{ (m/s)} / (2 \times 0.3 \text{ m}) = 567 \text{ Hz}$$





実際のモデルで異常検知を検証する

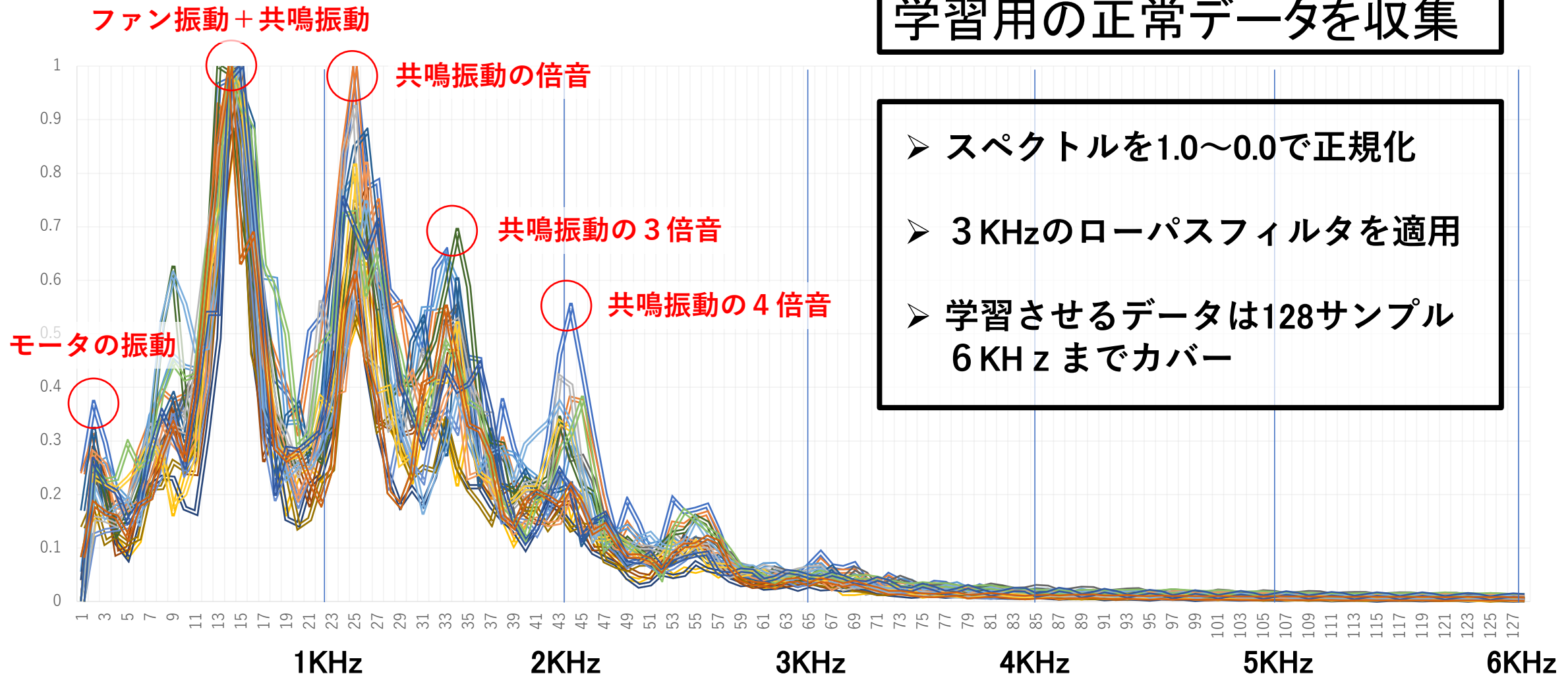
検証用システムの概要





実際のモデルで異常検知を検証する

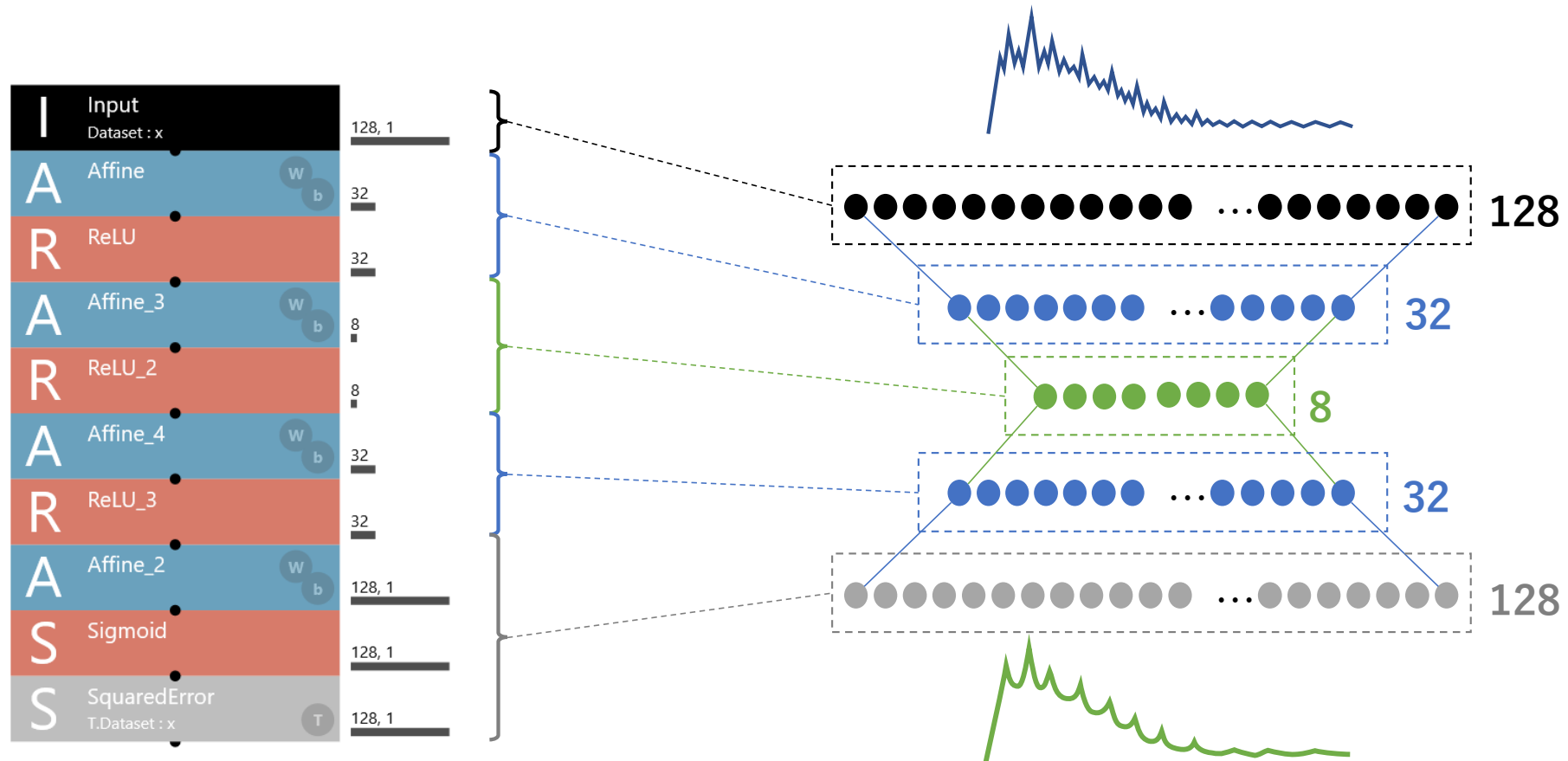
学習用の正常データを収集





実際のモデルで異常検知を検証する

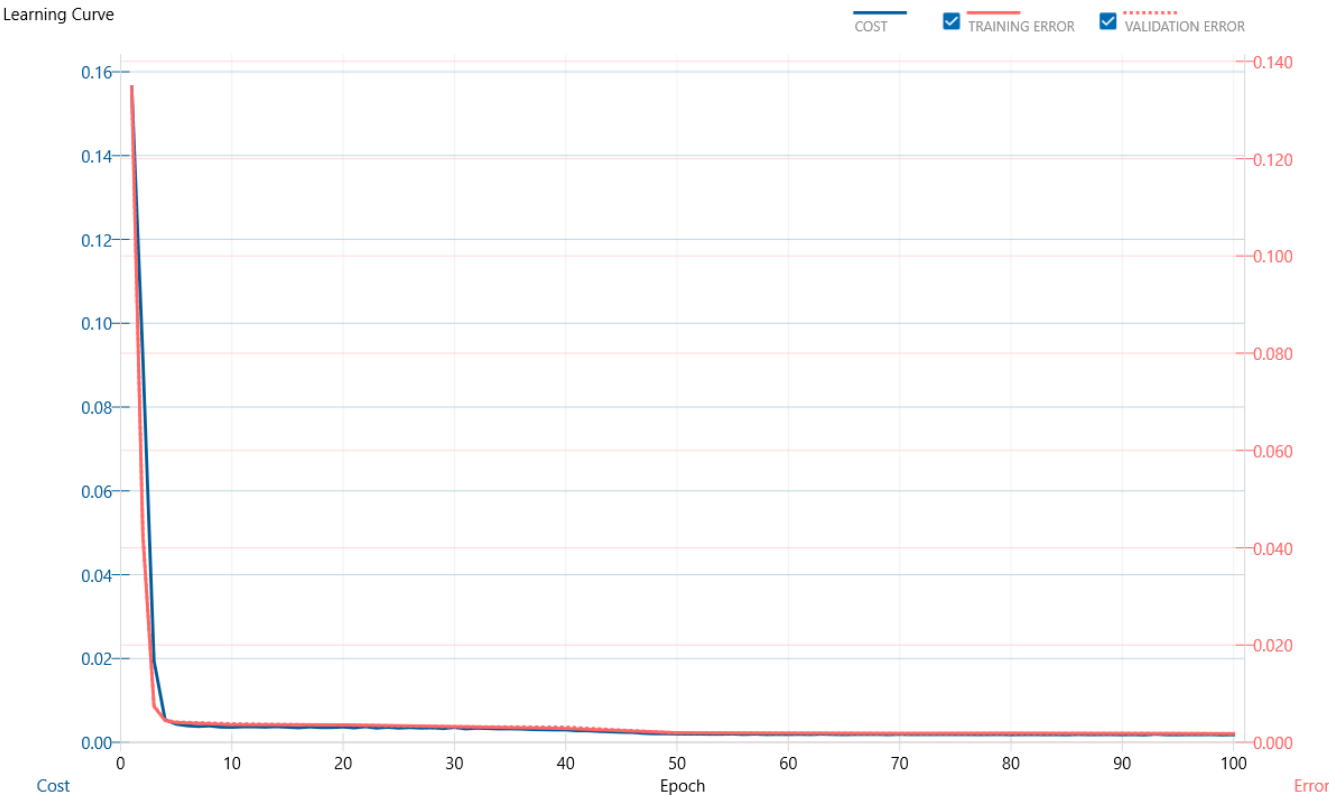
ニューラルネットワークモデルの生成



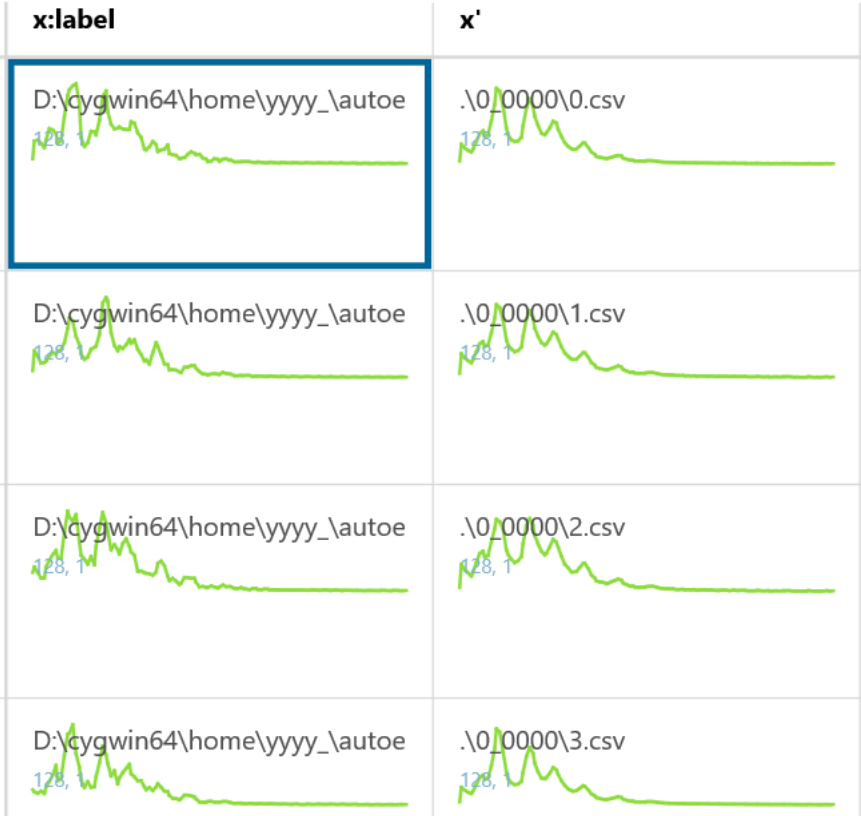


実際のモデルで異常検知を検証する

学習曲線



学習結果



入力データ

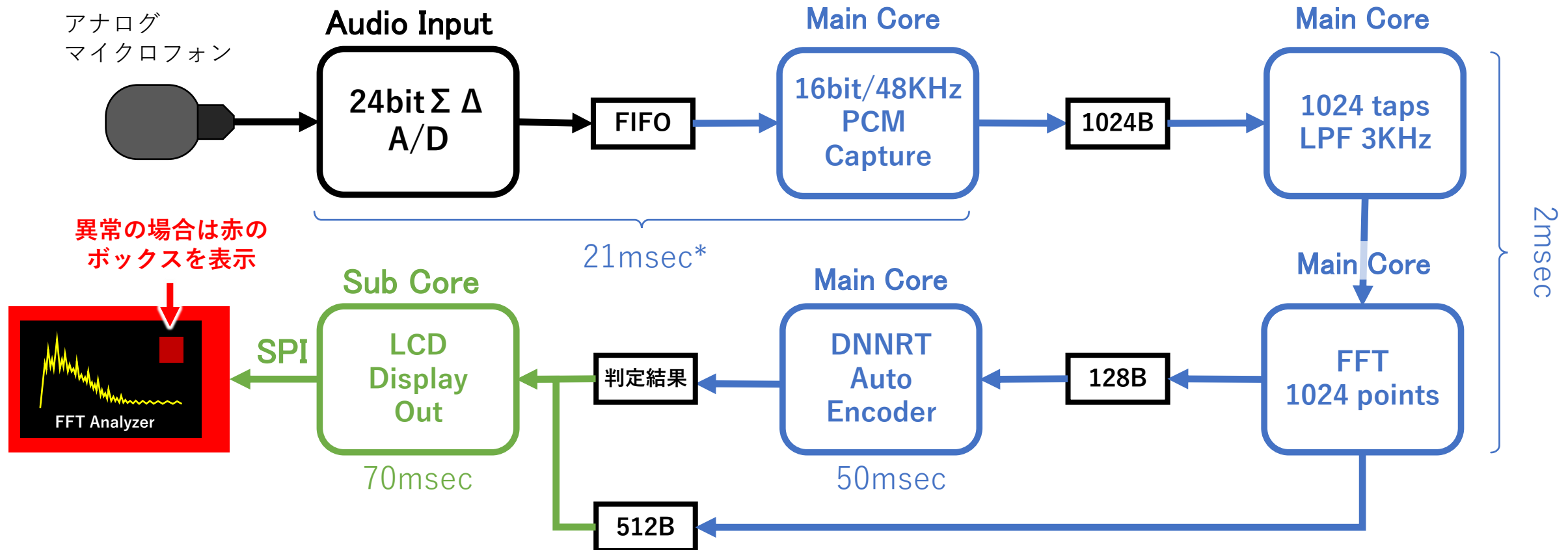
出力データ



実際のモデルで異常検知を検証する

検証用システムの概要

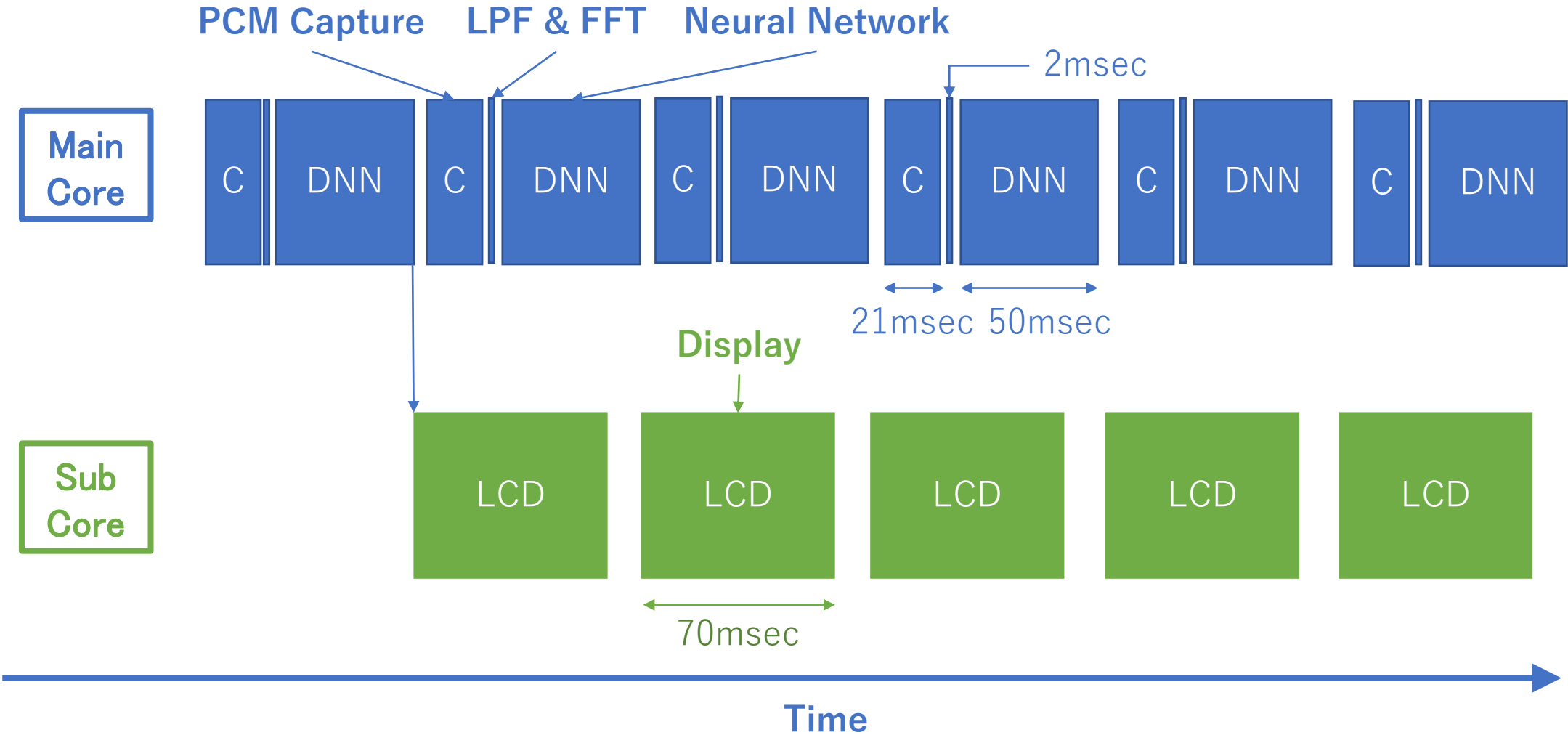
マルチコアによる並列処理を活用



* $1024\text{samples} \div 48000 (\text{samples/sec}) = 0.021 = 21\text{msec}$



実際のモデルで異常検知を検証する





実際のモデルで異常検知を検証する

オートエンコーダをSpresenseに組み込む

```
File nnbfile("model.nnb");
dnnrt.begin(nnbfile);

theAudio->readFrames(buff, buffer_size, &read_size);
LPF.put((q15_t*)buff, FFT_LEN);
LPF.get(pLPFSig, 0);
FFT.put(pLPFSig, FFT_LEN);
FFT.get(pData, 0);

DNNVariable input(FFT_LEN/8);
float *dnnbuf = input.data();
for (int i = 0; i < FFT_LEN/8; ++i) {
    dnnbuf[i] = pData[i];
}
dnnrt.inputVariable(input, 0);
dnnrt.forward();
DNNVariable output = dnnrt.outputVariable(0);
float sqr_err = 0.0;
for (int i = 0; i < FFT_LEN/8; ++i) {
    float err = pData[i] - output[i];
    sqr_err += sqrt(err*err);
}
sqr_err > 0.6 ? bNG = true : bNG = false;
```

Neural Network Consoleで出力した学習データの読み込み

PCMキャプチャされたサウンドデータのFIFOからの取り込み
とローパスフィルタ、FFTの処理

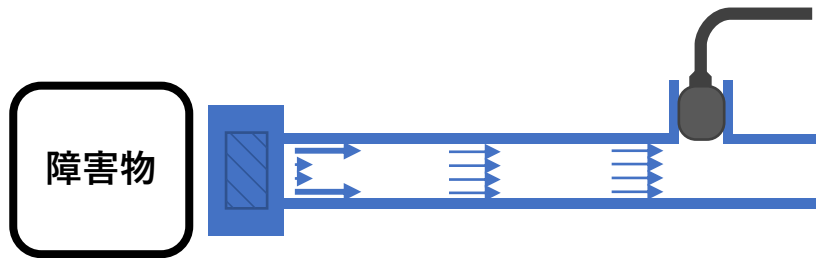
FFTによる周波数スペクトルデータをニューラルネットワー
ク（オートエンコーダ）に入力し、出力データを取得

入力データとオートエンコーダの出力の標準偏差を算出し、
閾値をもって、正常・異常を判定

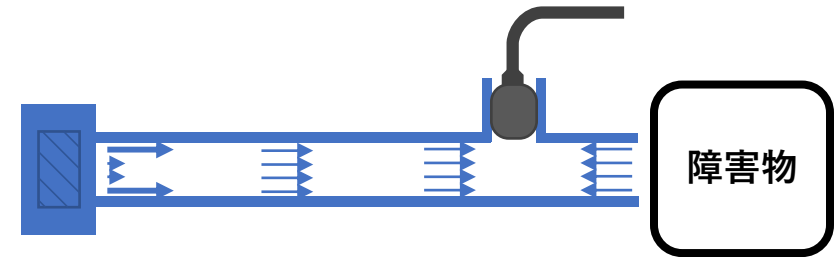
実際のモデルで異常検知を検証する

Spresenseによるパイプ異常の検知

ケース 1 : ファンの入り口が詰まる



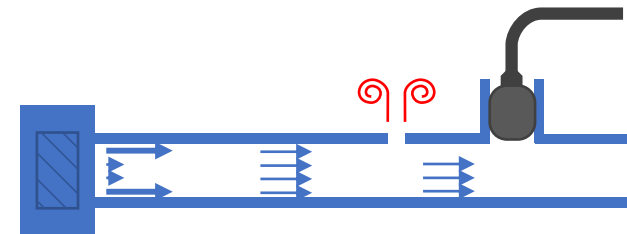
ケース 2 : パイプの出口が詰まる

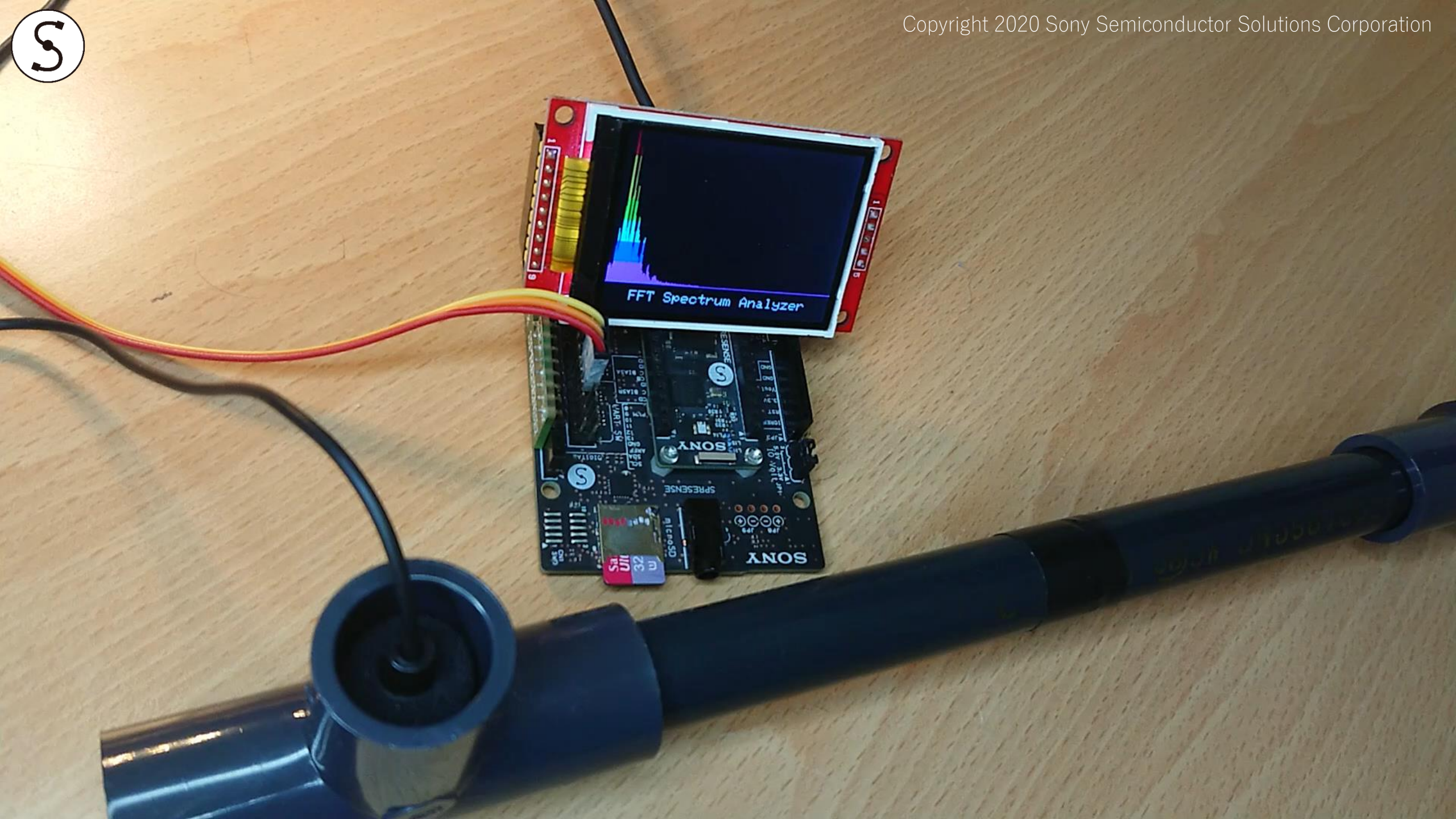


ケース 3 : ファンが止まる



ケース 4 : パイプに穴が開く







まとめ

- 👉 パイプの音響モデルの異常時の振る舞いを考察しました
- 👉 パイプの異常検知にオートエンコーダを検討しました
- 👉 SpresenseとNeural Network Console を使って、パイプの異常検知を検証しました

**SpresenseとNeural Network Console で
エッジAIを現場で活用してみましょう！**

SPRESENSEでエッジAIを現場で活用

