

IVI 先進研究分科会ホワイトペーパーVol.08

AI 深層学習応用研究分科会

# センサーデータ活用の事例とその技術

高速プレス機の AE センサー波形解析を通して

2021 年 8 月 31 日

**IVI**

一般社団法人

インダストリアル・バリューチェーン・イニシアティブ



1. はじめに.....	1	5.3. 良否判定事例3.....	24
1.1. 本書の目的.....	1	5.4. 長期データ傾向分析事例1.....	26
1.2. 対象読者.....	1	5.5. 長期データ傾向分析事例2.....	31
1.3. 関連ドキュメント.....	1	5.6. 長期データ傾向分析事例3.....	37
1.4. 用語略称.....	2	5.7. 長期データ傾向分析事例4.....	41
2. 用語の説明.....	3	5.8. 長期データ傾向分析事例5.....	43
3. 活動の経緯.....	4	6. データ解析高速化技術.....	47
3.1. 活動方針.....	4	6.1. 目的と方針.....	47
3.2. 活動概要とスケジュール.....	4	6.2. 高速化手法の解説.....	49
4. データ解析概要.....	5	6.3. 処理時間比較処理時間比較.....	50
4.1. データ解析の目的と取り組み方針.....	5	7. 機械学習.....	53
4.2. 解析対象データと解析目標.....	6	7.1. CNN.....	53
4.3. データ解析の取り組みと成果概要.....	7	7.2. AE/CAE.....	55
5. データ解析取組結果.....	9	8. 課題と今後の展望.....	57
5.1. 良否判定事例1.....	9	参考文献.....	58
5.2. 良否判定事例2.....	13	著者およびメンバー.....	59

# 1. はじめに

## 1.1. 本書の目的

---

本書は、センサーデータ(時系列データ)の分析を通して得られた知見・ノウハウを分析事例に基づき整理したものである。本書では、実際の分析で利用した手法についても概要を紹介することで読者の理解を深められるようにし、センサーデータ分析のポイントを紹介することを目的とする。

センサーデータの一例として、精密機械部品向けの高速プレス機から採取した AE センサーの波形データを活用し、プレス製品の良否判定や金型の交換時期を予測することを目指した分析事例を紹介する。本書の内容は、IVI 先進研究分科会「AI 深層学習応用研究分科会」の直近約 1 年の活動成果をまとめたものである。

## 1.2. 対象読者

---

本書の対象読者は、データ分析者、機械設計技術者、機械設備によるプラント運営者、波形解析分野の研究者の方々、さらに、技術分野のアナリストやデータ標準化を推進する方々を対象読者とする。

## 1.3. 関連ドキュメント

---

なし。

## 1.4. 用語略称

---

### 略称

CAE: Convolutional Autoencoder

CIOF: Connected Industries Open Framework

CNN: Convolutional Neural Network

DWT: Discrete Wavelet Transform

GAF: Gramian Angular Field

Grad-CAM: Gradient-weighted Class Activation Mapping

MKL: Math Kernel Library

RP: Recurrence Plot

## 2. 用語の説明

**高速プレス機**:高い剛性と精度を備え、高速で加工を可能としたプレス機械。

**金型**:素材の塑性または流動性の性質を利用し成形加工して製品を得るための、主として金属素材を用いてつくった型 [1]。

**AE センサー**:アコースティック・エミッション (Acoustic Emission, AE) とは、材料が変形あるいは破壊する際に、内部に蓄えていた弾性エネルギーを音波 (弾性波、AE 波) として放出する現象。AE 波は主に超音波領域 (数 10kHz ~ 数 MHz) の高い周波数成分を持つ。AE センサーは、圧電素子によってこの AE 波を電気信号に変換して破壊や変形の様子を非破壊的に評価する [2]。

### 3. 活動の経緯

#### 3.1. 活動方針

IVI 先進研究分科会「AI 深層学習応用研究分科会」では、「全員が自ら手を動かし AI・深層学習を身につけると同時に、現実の課題解決にチャレンジする」方針の下で活動を行ってきた。本分科会では、IVI の業務シナリオ WG や他の先進研究分科会と連携し、分科会メンバーが自身の知見を活かしてデータ分析を実施して課題解決に生かすとともに、メンバー間での切磋琢磨の場としてきた。

#### 3.2. 活動概要とスケジュール

本書に記載の内容は、IVI 業務シナリオ WG にて採取した精密機械部品向け高速プレス機の AE センサー波形データの提供を受けて分科会メンバーがデータ分析を実施したものである。図 3.2-1 に示すようにデータ分析は、OK/NG 波形データに基づく良否判定から開始し、金型の交換時期予測などの予知保全に向けた長期データの傾向分析を推進してきた。

		19/10	20/4	21/3	状況
プレス機での異常判定①	OK/NGデータをもとにモデル化	→			完了(5モデル作成) 100%判定可
プレス機での異常判定②	長期データの傾向分析・予知保全	→			推進中
プレス機での異常判定③	画像データによる良否解析			→	計画中
CIOF連携	・作成モデル登録 ・API実装			→	計画中

 : 本書の記載範囲

図 3.2-1 分科会の活動概要と推進スケジュール

## 4. データ解析概要

### 4.1. データ解析の目的と取り組み方針

---

本書で紹介するデータ解析は、次の目的で取り組んだ成果である。

- 品質判定: プレス製品の不良品判定の AI による自動化。
- 予知保全: プレス機の金型摩耗による金型の適切な交換時期予測。

具体的には、波形特徴の時系列変化の解析を行いながら、OK/NG の自動判定、時系列変化の自動判定を実現する機械学習モデルを作成する、という流れで推進した。

## 4.2. 解析対象データと解析目標

データ収集システムの構成を図 4.2-1 に示す。高速プレス機 [1] に AE センサーを装着し、採取した波形データをエッジコンピュータで収集する<sup>1</sup>。データ収集のサンプリングレートは 1MHz とし、1 波形あたり 105000 ポイントのデータを収集した。波形取得のタイミングは、プレス機の加工タイミングトリガを装置より入手し実施した。

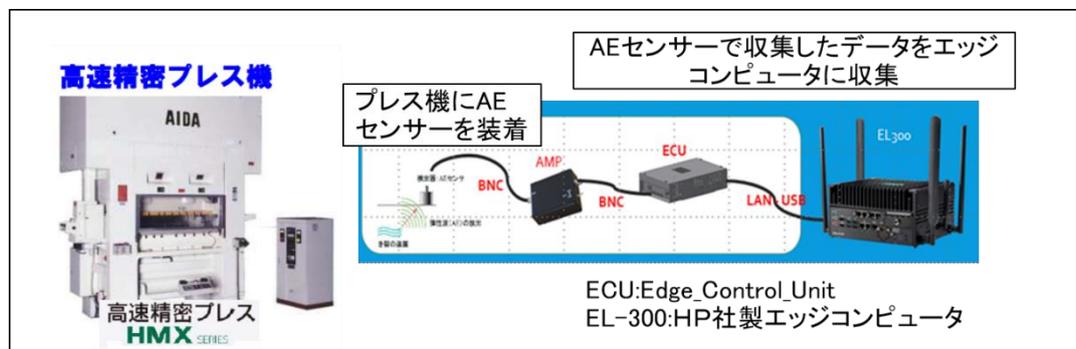


図 4.2-1 データ収集システムの構成

<sup>1</sup> プレス加工とは、プレス機に上下に分かれた金型を取り付け、型の中に材料を投入し上型下型で挟み込む事により、製品を成形する加工方法。

### 4.3. データ解析の取り組みと成果概要

#### (1) データ解析条件

図 4.3-1 にデータ解析に利用した採取データの概要と解析目標を示す。良否判定に用いた NG データは、意図的に作成したものである。LOT とはプレス機にかける原材料の単位であり、1LOT で約 10 万ショット(10 万個の生産物)の処理を行う。扱ったデータは各 LOT の先頭 200 ショット、中間 200 ショット、最後 200 ショットとした。#1~#3 は同一のプレス装置、金型、#4 は前記とは別のプレス装置、金型である(どちらも順送型<sup>2)</sup>)。

#	採取データ概要	データ解析に利用したデータ件数	解析目標
1	OKデータ、NGデータ*	OK:200件、NG:200件	OK/NG判定
2	94万ショットデータ(全OK)、 LOT1~LOT9(各10万件)	各LOTの先頭、中間、最後の200件	OK/NG判定、金型摩耗判定
3	200万ショットデータ(全OK)、 LOT9~LOT15(各10万件)	各LOTの先頭、中間、最後の200件	OK/NG判定、金型摩耗判定
4	40万ショットデータ(全OK)	有効な全データ(約14万件)	OK/NG判定、金型摩耗判定

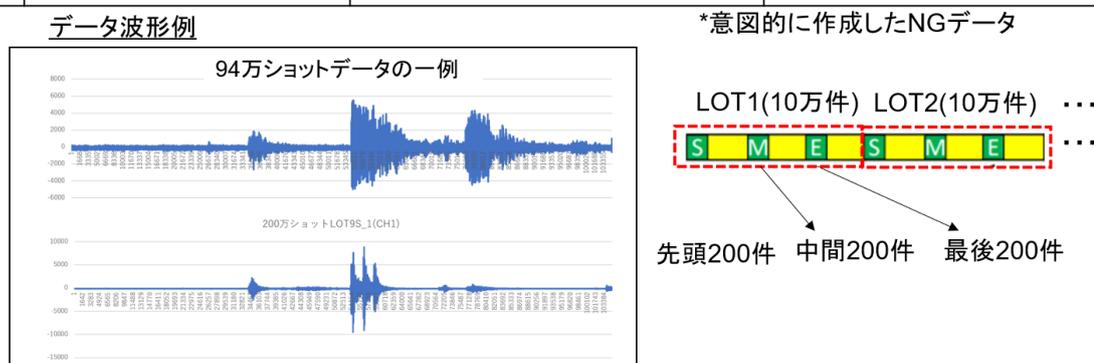


図 4.3-1 データ解析条件

#### (2) データ解析結果概要

表 4.3-1 に各データ解析結果の概要を示す。解析内容の詳細は次章以降に示す。

<sup>2</sup> 順送型は、複数の工程(例えば、外形抜き・曲げ・穴あけという複数の工程)が一つの金型に組み込まれており、自動で材料を順送りにすることによりプレス機 1 台で複数の工程を経て、製品を完成させる金型。

表 4.3-1 解析結果概要

#	データ解析概要	得られた成果
5.1	機械学習によるOK/NG判定	CNN:4画像分類ベース、OK/NG100%判定、学習過程、GradCamカラーマッピング
5.2	機械学習によるOK/NG判定、前処理の工夫によるOK/NG分離性能向上	前処理(GAF,PR+データ切り出し範囲)による分離性能向上、OK/NG100%判定
5.3	長期データ(94万ショット)OK/NG判定	OK/NGデータモデルにより100% OK判定
5.4	長期データ(94万ショット/200万ショット)傾向分析	DWTによる信号分離とトレンド分析
5.5	長期データ(94万ショット/200万ショット)傾向分析	数値解析のアプローチ:ロット毎にデータ傾向の変化を繰り返す、特定ロット間でデータ傾向変化にギャップあり
5.6	長期データ(200万ショット)傾向分析	#4の傾向変化を機械学習で自動判定
5.7 5.8	長期データ(40万ショット)傾向分析	材料の切れ目等で変化、ばらつきの多い区間と安定区間あり、データをグルーピング

## 5. データ解析取組結果

### 5.1. 良否判定事例1

#### (1) 対象データと解析目的

高速プレス機に設置した AE センサデータ（図 4.3-1#1）から得られる、センサ波形画像を深層学習させ、OK/NG 判定分類器を作成、その実現性を検証する。

#### (2) データ前処理

データ前処理の流れを図 5.1-1 に示す

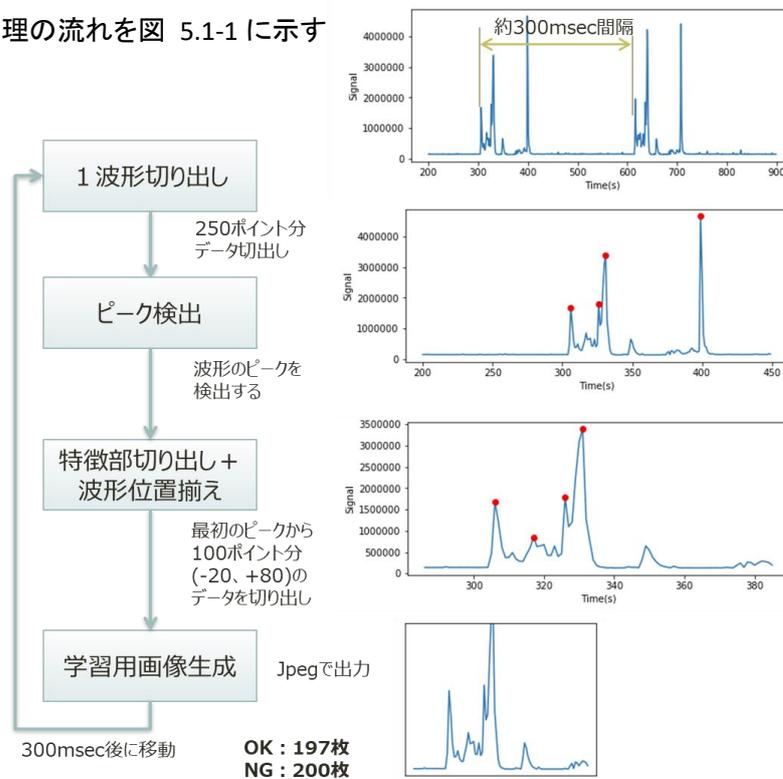


図 5.1-1 データ前処理の流れ

(3) 学習

学習環境、学習用方法について図 5.1-2 に示す。

- 学習環境
  - ✓ フレームワーク : Keras+TensorFlow
  - ✓ CNNモデル : 4画像分類サンプルを流用
- 学習方法
  - ✓ 学習用データ : 評価用データ = 80%:20% = 317枚:80枚
  - ✓ 画像サイズ : 50x50pixel
  - ✓ Epoch数 : 200/300/400

```

# 200epoch
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same', input_shape=(3, train_shape[1:-1]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('softmax'))
    
```

図 5.1-2 学習環境、学習方法

Epoch<sup>3</sup>毎の学習曲線について図 5.1-3 に示す。

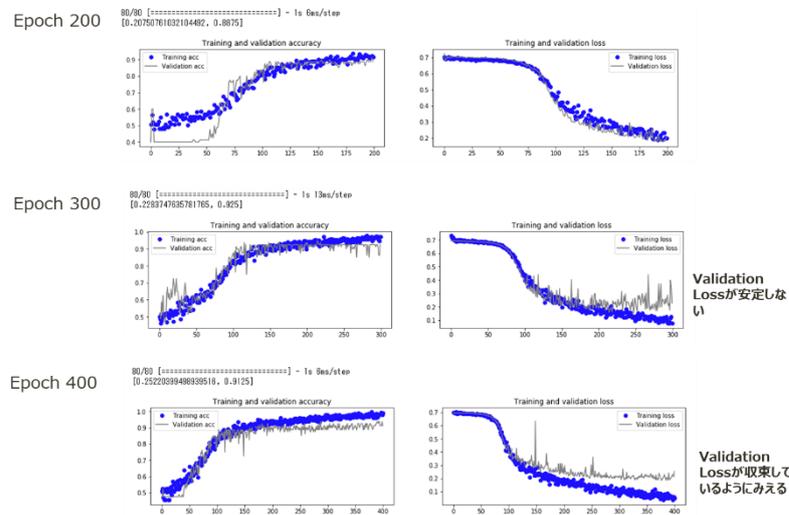


図 5.1-3 学習曲線

(4) 評価

学習済みモデルによる分類性能 及び 推論処理時間について図 5.1-4 に示す。

評価用データは前項で示した 20%のデータを用いた。

<sup>3</sup> 1つの訓練データを何回繰り返して学習させるかの数のこと。深層学習では epoch 数を多くすると訓練精度が上がるが、予測精度(未知の入力に対する精度)が下がる場合がある。これを過学習といい、学習過程では、両者を適切にバランスさせる最適な epoch 数を見つけることが重要なポイントである。

Epoch	結果\入力	OKデータ	NGデータ	推論処理時間
Epoch 200	OK判定	100.0% (20/20)	10.0% (2/20)	Found 40 images belonging to 1 classes. 40/40 [=====] - 0s 12ms/step <b>推論処理時間 12msec</b>
	NG判定	0.0% (0/20)	90.0% (18/20)	
	結果\入力	OKデータ	NGデータ	
Epoch 300	OK判定	100.0% (20/20)	5.0% (1/20)	Found 40 images belonging to 1 classes. 40/40 [=====] - 0s 11ms/step <b>推論処理時間 11msec</b>
	NG判定	0.0% (0/20)	95.0% (19/20)	
	結果\入力	OKデータ	NGデータ	
Epoch 400	OK判定	100.0% (20/20)	0.0% (0/20)	Found 40 images belonging to 1 classes. 40/40 [=====] - 0s 12ms/step <b>推論処理時間 12msec</b>  Epoch数を増やすと、誤認識が減っている。 推論処理時間には差がない。
	NG判定	0.0% (0/20)	100.0% (20/20)	
	結果\入力	OKデータ	NGデータ	

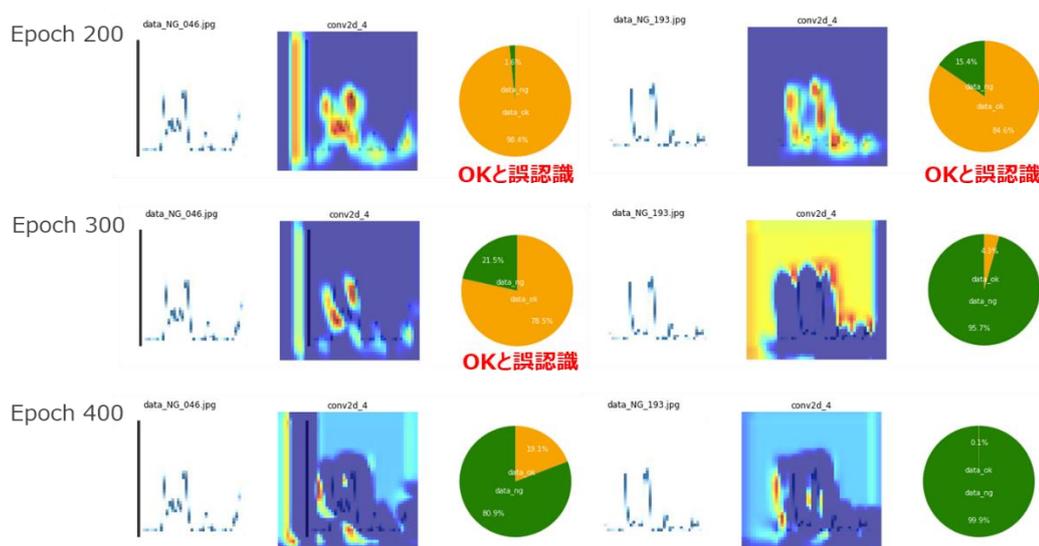
図 5.1-4 分類精度及び推論処理時間

Epoch400 で正答率 100%の結果となった。

(5) 深層学習における OK/NG 判断基準

一般的に深層学習のロジックはブラックボックスと言われているが、推論の過程で画像のどの部分を判断に用いているかを確認できる Grad-CAM という手法を用い、本解析データでの見える化を行った。

Epoch200/300 で誤認識したショットのデータを用い、Grad-CAM 表示をしたものが図 5.1-5 である。



直前のConv層をカラーマッピング表示  
赤い部分を重要視している、青い部分は利用していない。

図 5.1-5 GRAD-CAM 表示例

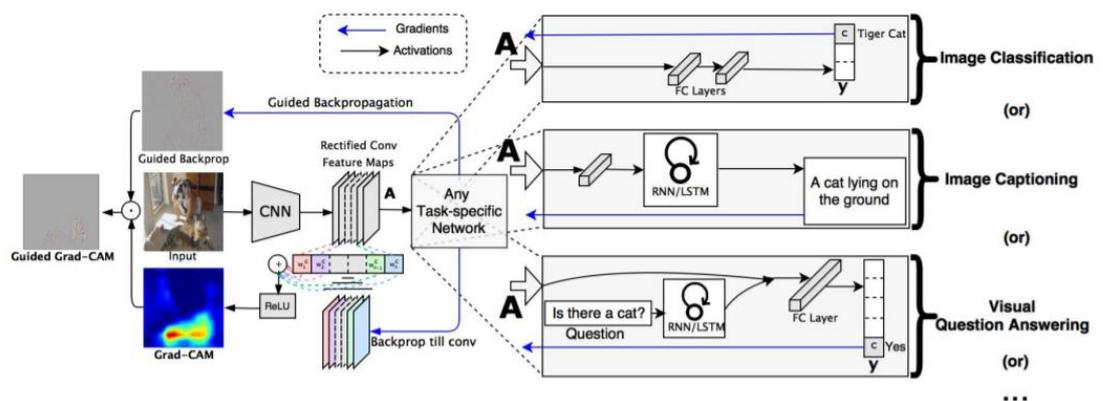
波形の山の部分を重要視していると OK と誤認識し、波形の立ち上り部分を重要視していると正しく判断していることが見て取れる。

(6) まとめ

少ないデータ数での評価ではあるが、波形画像による OK/NG 判定の実現性の可能性は十分にあると考えられる。また判定精度向上のために、Grad-CAM が有効であることも確認できた。

Grad-CAM

Grad-CAM とは、Gradient-weighted Class Activation Mapping の略で、CNN が物体を認識する際に、画像のどこに着目しているかを可視化する方法の一つである。最終出力直前の畳み込み層 (Conv 層) が出力した特徴量マップから、ニューラルネットワークの出力値の間の勾配を利用して、判断根拠を可視化している。



出典 : Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra

<https://arxiv.org/abs/1610.02391>

## 5.2. 良否判定事例2

### (1) 解析方法概要

- (a) AE センサ波形 Ch3 に注目し、波形の切出し起点を波形の最大ピーク [Max-Peak] と 波形の最初のピーク [First-Impact] の 2 つの検出方法で試行。また波形の切出し時間も 300, 1200, 2000, 33000 point と長さを変えて試行。

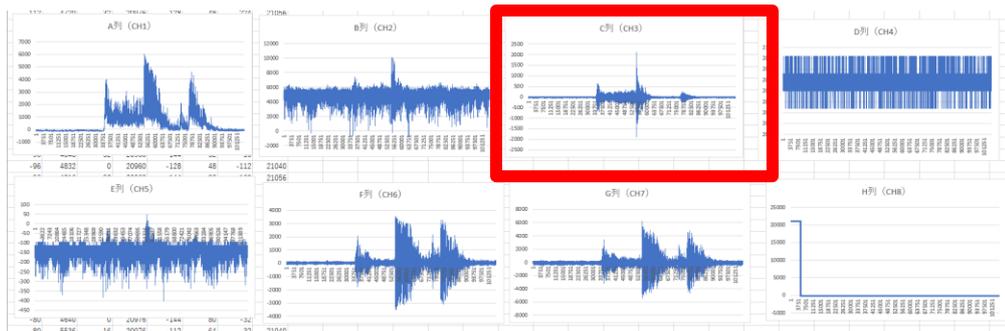


図 5.2-1 AE センサーの波形

- (b) 前処理は時系列データであるセンサ波形をビジュアル化(画像化)する方法として時系列データのエンコーディング手法である Recurrence Plot (以下 RP と省略) と Gramian Angular Field (以下 GAF と省略) の 2 つの方法を試した。
- (c) 上述 b.でビジュアル化した画像を Deep Autoencoder (以下 DAE と省略) で OK データのみで学習させた後に、OK データと NG データを画像データとして判別できるか検証を行った。また AI モデルの比較の為に、Variational Autoencoder (VAE) も試行した。

### (2) データ前処理方法と AI 解析手法の説明

#### (a) Recurrence Plot (RP)

$\vec{x}(i) = \vec{x}(j)$  の時にのみ、描画されるペア  $(i, j)$  を持つ。  
例えば値が周期Tを持つならば、そのような時間のペア全体はTの倍数毎に分けられ、対角線と並行な線として描かれる。

$$R(i, j) = \begin{cases} 1 & \text{if } \|\vec{x}(i) - \vec{x}(j)\| \leq \epsilon \\ 0 & \text{otherwise,} \end{cases}$$

$(i, j)$  が  $R(i, j) = 1$  の場合のみ点を描画することでリレンスプロットを描画する。

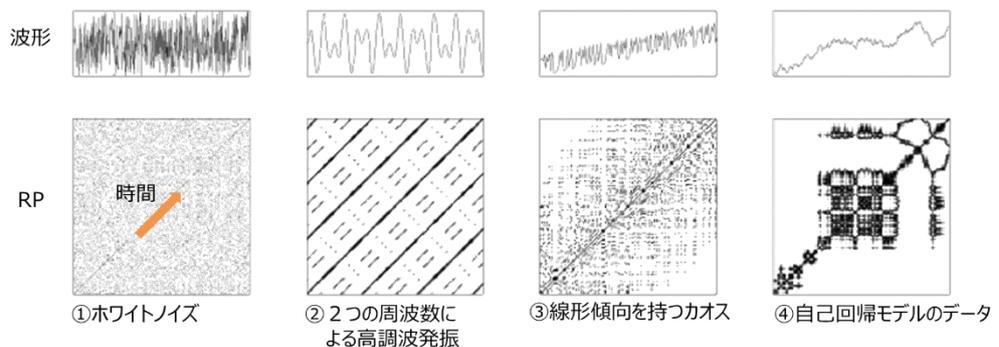


図 5.2-2 RECURRENCE PLOT

(b) Gramian Angular Field (GAF)

ベクトル  $u$  と  $v$  の内積は

$$\langle u, v \rangle = u_1 \cdot v_1 + u_2 \cdot v_2$$

$$\langle u, v \rangle = \|u\| \cdot \|v\| \cdot \cos(\theta)$$

$u$  と  $v$  のノルムが 1 の時は

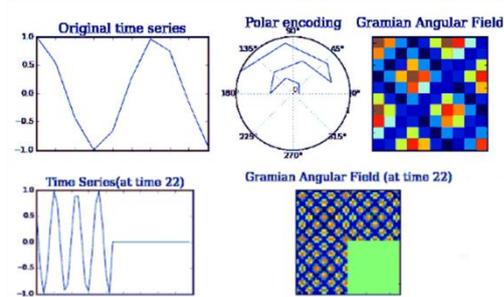
$$\langle u, v \rangle = \cos(\theta)$$

$$\langle u, v \rangle = \sum_{i=1}^n u_i \cdot v_i$$

時間方法に  $n$  個のベクトルの時

$$G = \begin{pmatrix} \langle v_1, v_1 \rangle & \langle v_1, v_2 \rangle & \dots & \langle v_1, v_n \rangle \\ \langle v_2, v_1 \rangle & \langle v_2, v_2 \rangle & \dots & \langle v_2, v_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_n, v_1 \rangle & \langle v_n, v_2 \rangle & \dots & \langle v_n, v_n \rangle \end{pmatrix}$$

$$G = \begin{pmatrix} \cos(\phi_{1,1}) & \cos(\phi_{1,2}) & \dots & \cos(\phi_{1,n}) \\ \cos(\phi_{2,1}) & \cos(\phi_{2,2}) & \dots & \cos(\phi_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_{n,1}) & \cos(\phi_{n,2}) & \dots & \cos(\phi_{n,n}) \end{pmatrix}$$



$$G = \begin{pmatrix} \cos(\phi_1 + \phi_1) & \cos(\phi_1 + \phi_2) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cos(\phi_2 + \phi_2) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cos(\phi_n + \phi_2) & \dots & \cos(\phi_n + \phi_n) \end{pmatrix}$$

<https://medium.com/analytics-vidhya/encoding-time-series-as-images-b043becbdfb3>

図 5.2-3 GRAMIAN ANGULAR FIELD

(c) Deep Auto Encoder (DAE)

図 5.2-4 に示す AI モデルで OK データの RP 画像及び GAF 画像が復元できるようにモデル学習。OK データ画像は元画像と復元画像の乖離度が小さくなるが、NG データ画像は復元が上手く行かない為に元画像と復元画像との乖離が大きくなる特性を利用して、OK/NG の判別を行った。

(3) RP によるデータ前処理結果

以下に RP でデータ前処理を行った結果の OK データの画像と NG データの画像を下に示す。OK 画像と NG 画像の特徴が異なっており、人間の眼でも判別が可能であることがわかる。特に RP300 では特徴差異が大きい。

(a) RP300: Max-Peak を起点として後に 300point 波形を切り出し

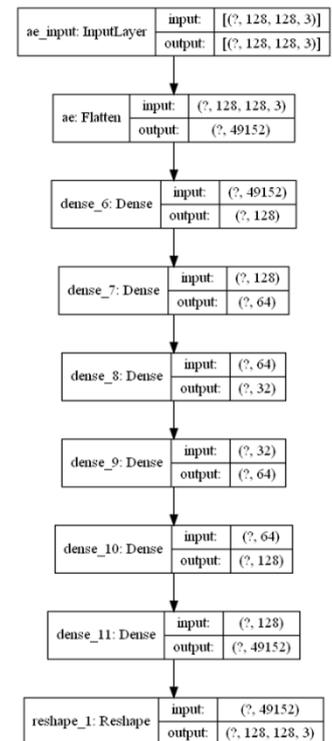
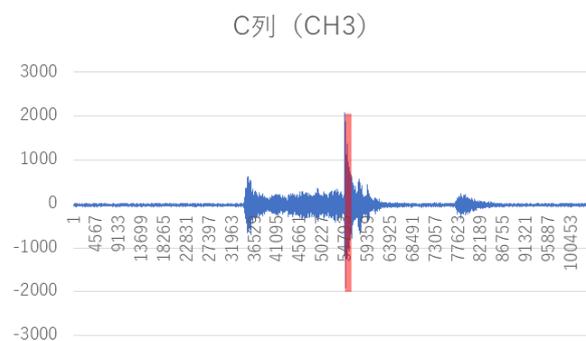
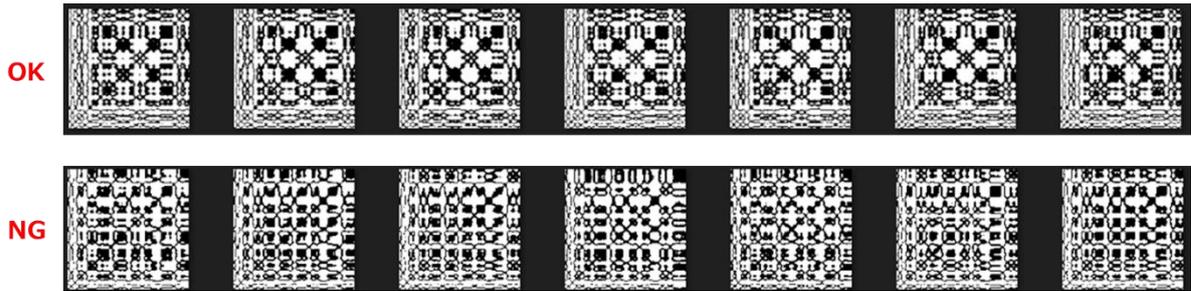
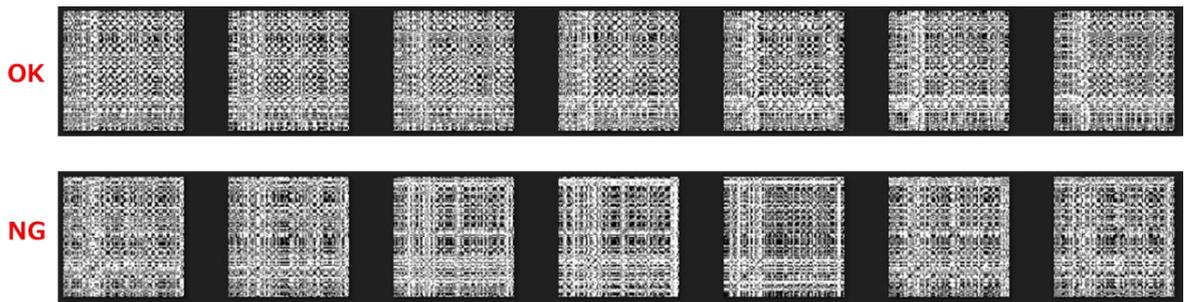
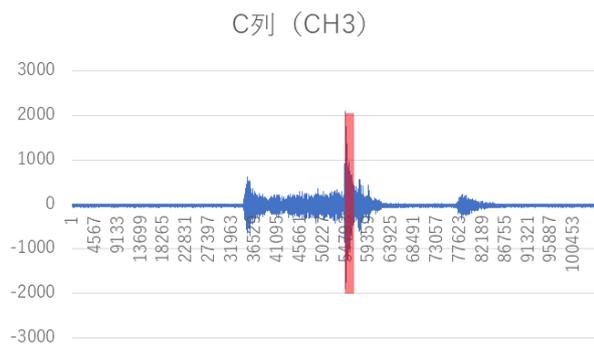


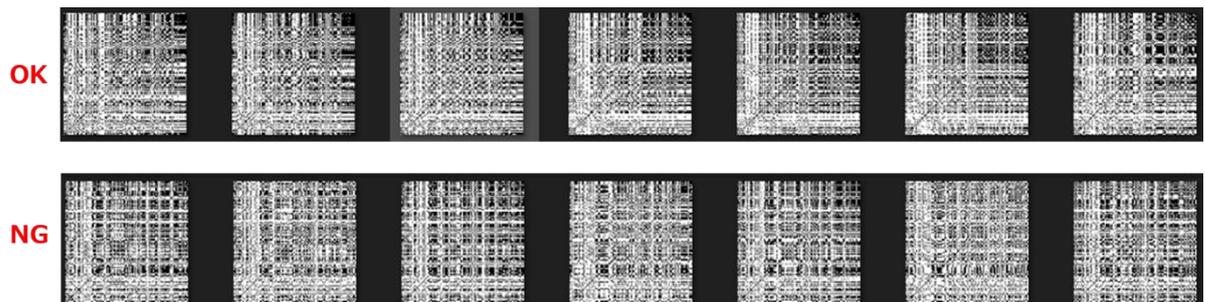
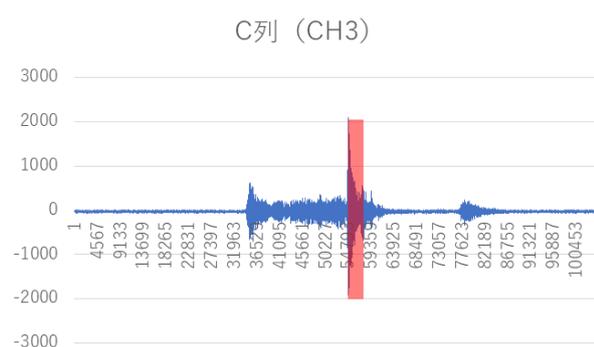
図 5.2-4 DAE の AI モデル



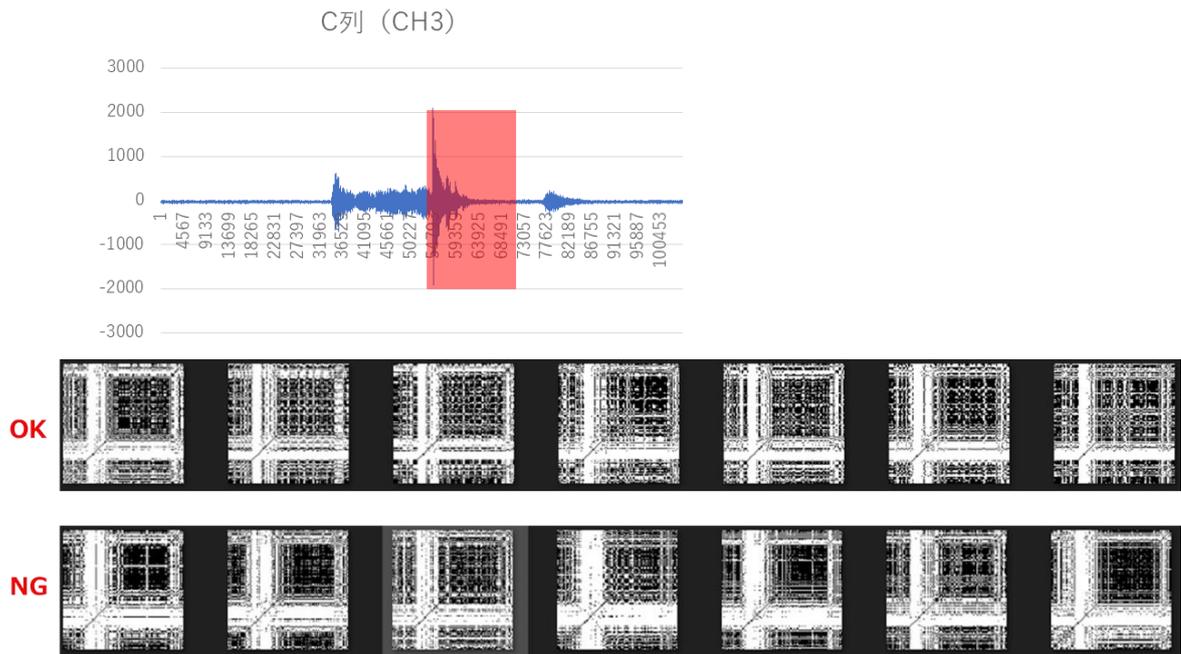
(b) RP1200:Max-Peak を起点として後に 1200point 波形を切り出し



(c) RP2000:Max-Peak を起点として後に 2000point 波形を切り出し



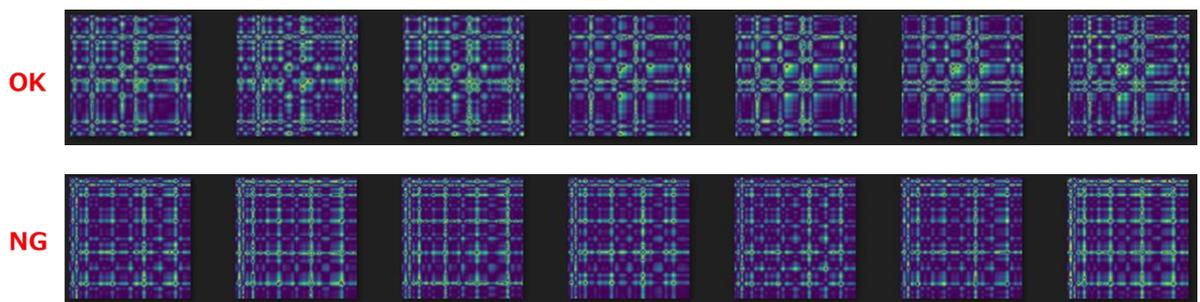
(d) RP33000: Max-Peak を起点として前に 6600point + 後に 26400point 合計 33000point 波形を切り出し



(4) GAF によるデータ前処理結果

以下に GAF でデータ前処理を行った結果の OK データの GAF 画像と NG データの GAF 画像を下に示す。GAF300 では人間の眼でも OK 画像と NG 画像の判別が可能だが、GAF33000 では画像が似通っており、判別が難しいことがわかる。

(a) GAF300: Max-Peak を起点として後に 300point 波形を切り出し



(b) GAF33000: Max-Peak を起点として前に 6600point + 後に 26400point 合計 33000point 波形を切り出し



(5) AI モデルによる学習・検証結果

(a) AI モデル学習及び検証方法

OK データ 200 サンプルの内、半分の 100 サンプルを学習データとして利用。学習用 100 サンプルの内、90 サンプルをトレーニング用に 10 サンプルをバリデーション用に使用した。トレーニング 45 サンプル/ミニバッチで 200epoch 行った。

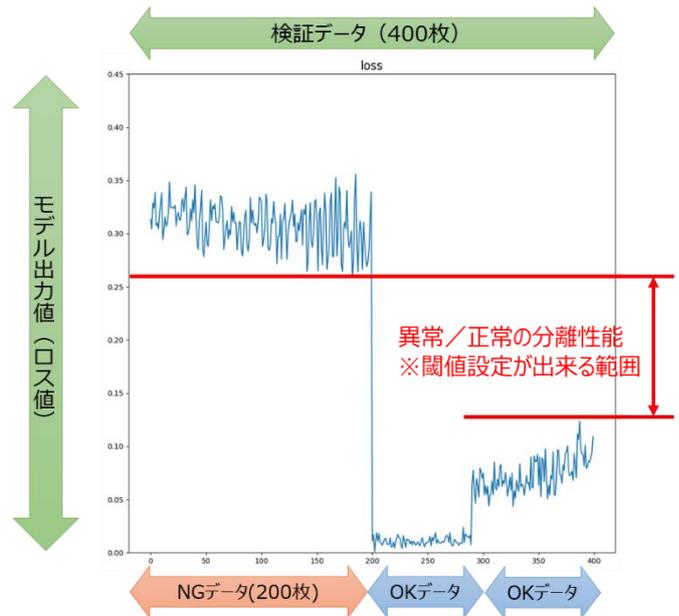


図 5.2-5 学習・検証結果

(b) グラフの見方 [図 5.2-5]

横軸にサンプル数 (NG データ 200 サンプル+OK データ 200 サンプル)を、縦軸に DAE の出力値 (ロス値)を示す。DAE の出力は OK データでは元画像との差分(ロス値)が小さく、NG データでは画像が復元できない為に元画像との差分が大きくなる。OK データと NG データでロス値の差が大きければ、閾値を設けることによって判別が可能となる。横軸の最初の NG データ 200 サンプルは検証に、次の OK データ 100 サンプルは学習に、最後の OK データ 100 サンプルは検証に使用した。学習に使用した OK サンプルは当然ながらロス値はほぼ 0 近傍となる。

(c) RP と GAF の DAE 結果比較 [図 5.2-6]

最大ピーク [Max-Peak] を起点として波形を切り出した時の RP300, RP33000 と GAF300, GAF33000 の DAE 結果を図 5.2-6 に示す。図 5.2-6 はそれぞれトレーニングカーブで、太い青線はトレーニングデータ(90 サンプル)のロス値、細い青線はバリデーションデータ(10 サンプル)のロス値を、下図は OK/NG 判別に用いる検証結果のロス値を示す。

RP300: OK/NG の分離性能は4つの中では最も良く判別は可能だが、OK を NG と誤検知している点がある。これは最大ピーク位置にバラツキがある為、波形切り出し位置がズレたことによる誤検知と思われる。

RP33000: データ間のバラツキが大きいため学習が正常に進んでおらず、過学習を起している可能性もある。OK/NG の分離性能が悪く、閾値による判別は困難。

GAF300: OK/NG の分離性能は概ね良好で判別は可能だが、OK サンプルを NG と誤検知している点があるのは RP300 と同じ。同じサンプルを誤検知しており、原因も同様と考えられる。

GAF33000: 学習は正常に進んでいるようだが、OK/NG の分離性能が悪く、閾値による判別は困難。

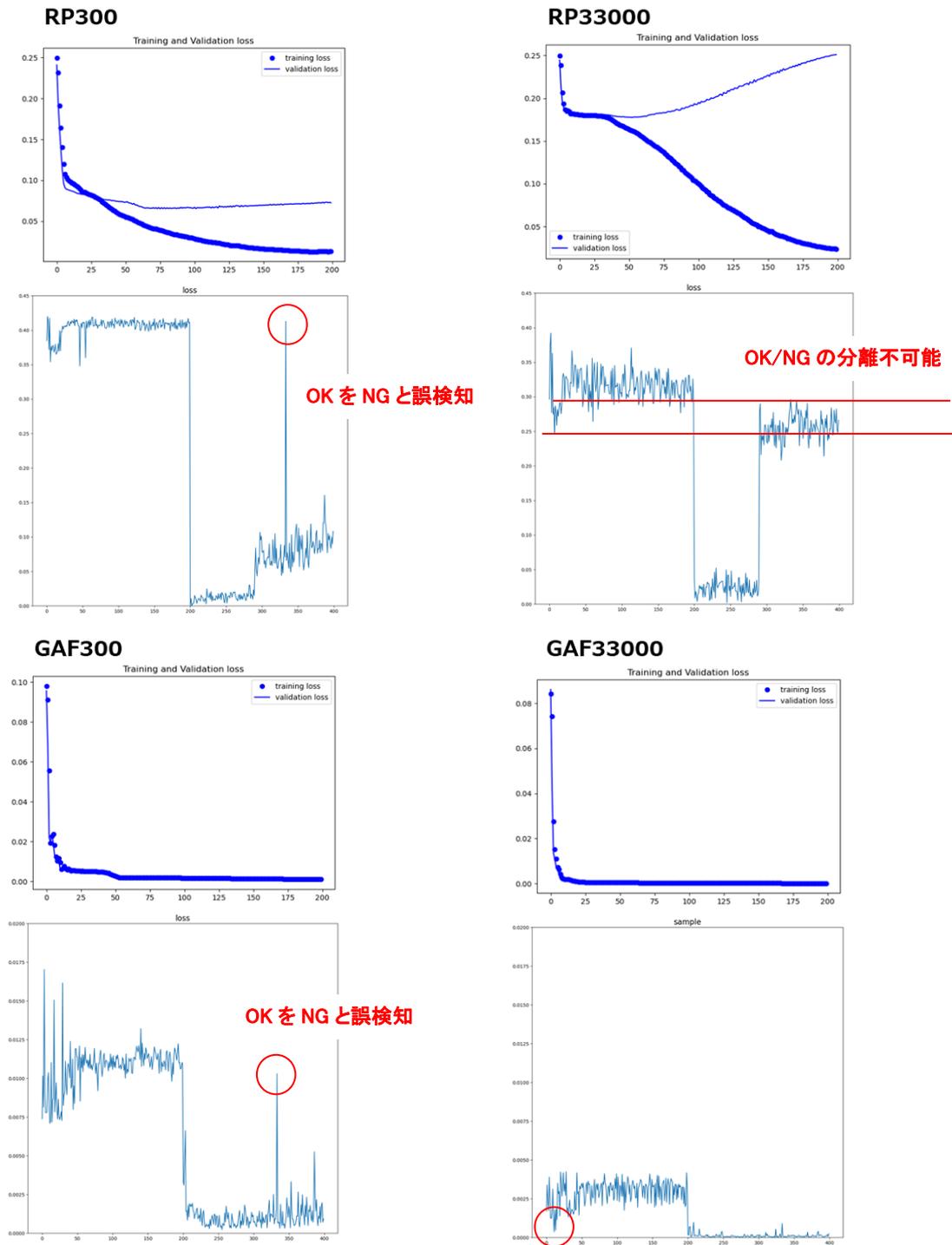


図 5.2-6 RP と GAF の DAE 結果比較

(d) 誤検知と分離性能が悪い原因 [図 5.2-7]

GAF300 で OK データを NG と誤検知した原因: CH3 の最大ピーク位置のバラツキが大きい為、その位置ズレにより OK データでも特徴データ (GAF 画像) の差が大きくなり、未学習範囲では NG

データと誤検知されてしまう。GAF 画像を見ても外れ値の OK 画像はその他の OK 画像と明らかに特徴が異なっていることがわかる。

GAF33000 で分離性能が悪い原因: 出力される画像の特徴量が小さく、OK/NG の特徴量差が小さい為。NG 画像と OK 画像を見ても非常に似通っていることがわかる。さらに NG データは特に最大ピーク位置のバラツキが大きく、NG データのピーク位置がたまたま OK データのピーク位置と一致してしまった場合には OK/NG の判別ができなくなる。

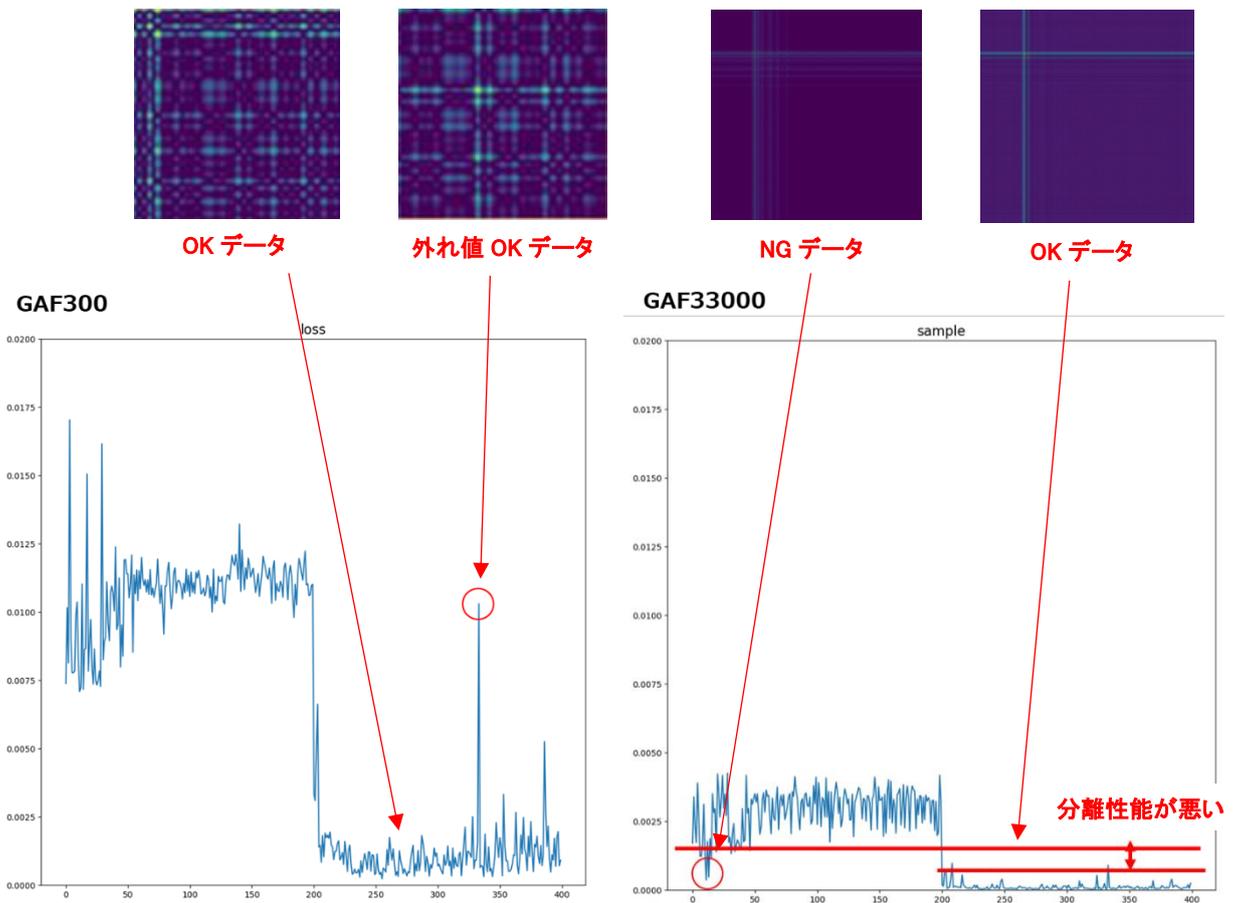


図 5.2-7 誤検知と分離性能が悪い原因

(e) 波形切り出し起点の改善 [図 5.2-8]

最大ピーク位置はバラツキが大きい為に波形切り出し位置がバラついてしまい、検出精度が落ちてしまった。そこでより位置が安定している First-Impact (最初のピーク位置)を基準に波形を切り出す方式を検討した。相互相関関数を利用して First-Impact を検出し、そこから 131point 後の位置から波形を切り出すことにした。

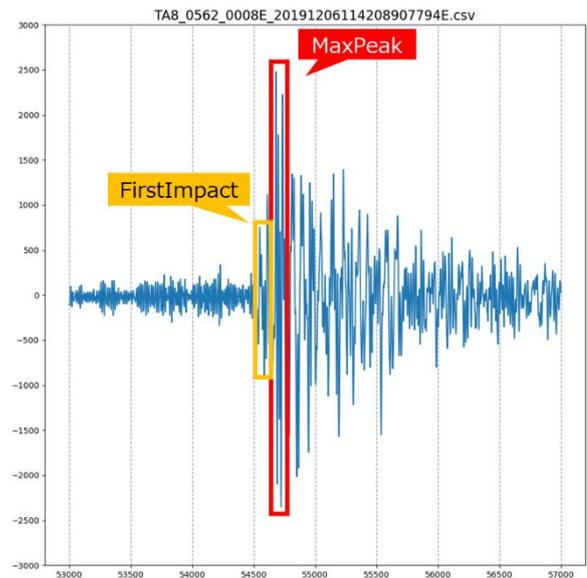
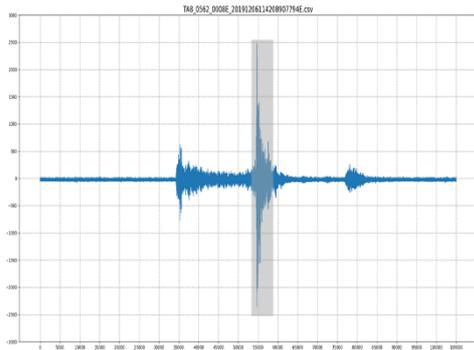
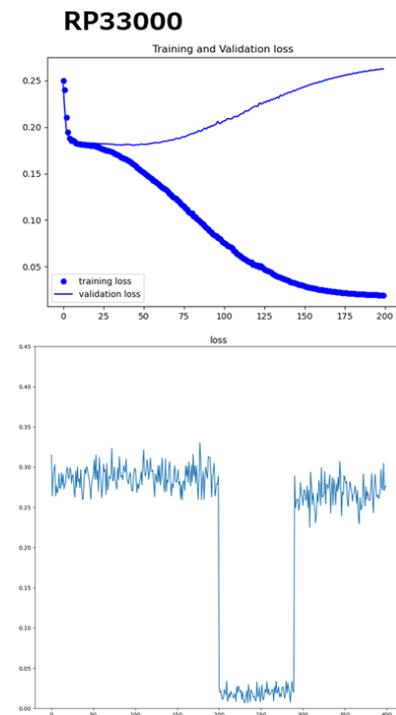
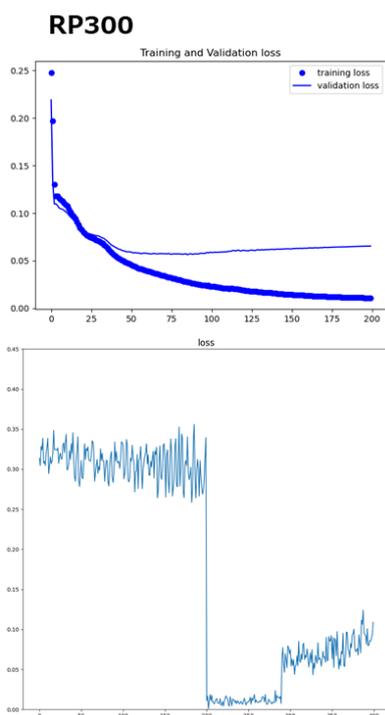


図 5.2-8 波形切り出し起点の改善

(f) First-Impact を起点とした時の DAE 結果 [図 5.2-9]

最大ピーク [Max-Peak] を起点として波形を切り出した時の RP300, RP33000 と GAF300, GAF33000 の DAE 結果を図 5.2-9 に示す。それぞれ上図はトレーニングカーブで、太い青線はトレーニングデータ(90 サンプル)のロス値、細い青線はバリデーションデータ(10 サンプル)のロス値を、下図は OK/NG 判別に用いる検証結果のロス値を示す。

RP300: 波形の切出し位置が安定したため、図 5.2.3 で見られた OK を NG と誤検知する点は改善された。また OK/NG の分離性能はほぼ変わっていない。



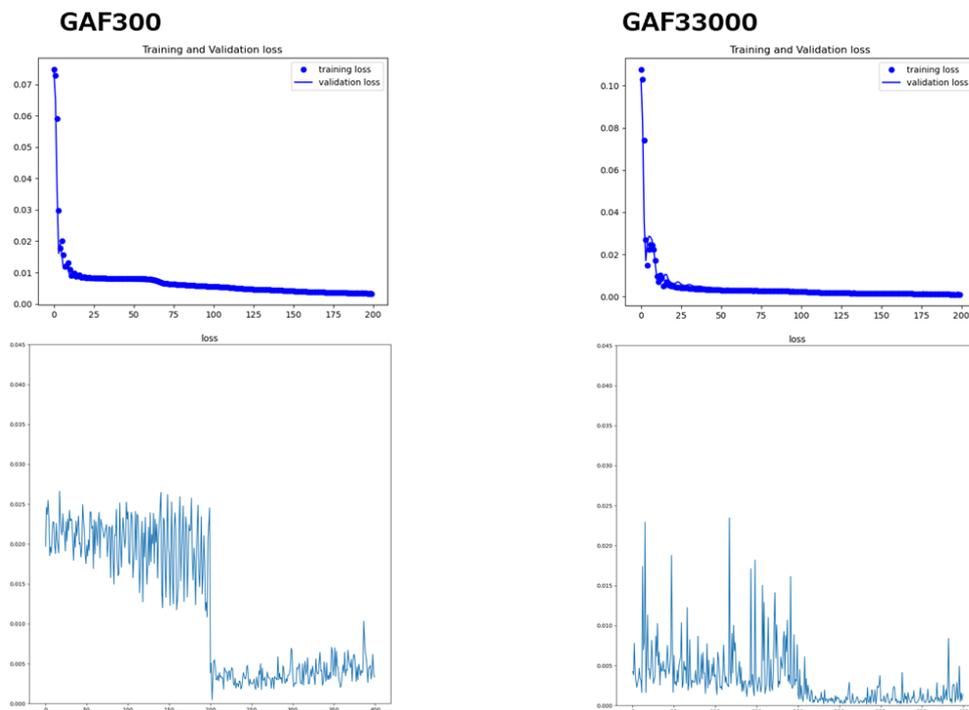


図 5.2-9 FIRST-IMPACT を起点とした時の DAE 結果

RP33000: 図 5.2.3 とほぼ同様で、OK/NG の分離性能が悪く判別不可能。

GAF300: 波形の切り出し位置が安定したため、図 5.2-6 で見られた OK を NG と誤検知する点は改善されたが、NG データと OK データの差が小さくなることもあり、分離性能が若干悪化した。

GAF33000: 波形の切り出し基準が揃った為に、OK/NG の差が小さくなり分離性能は悪化した。

全体的には安定した基準である First-Impact を起点とすることで、誤検知は無くなるが、OK/NG の分離性能は若干悪くなる傾向が見られた。

(g) 波形切り出し範囲の最適化 [図 5.2-10]

OK/NG の分離性能が良い RP をベースに、波形切り出し範囲を変えて分離性能を比較し、最適な切り出し範囲を検討した結果を図 5.2-10 に示す。それぞれ上図はトレーニングカーブで、太い青線はトレーニングデータ(90 サンプル)のロス値、細い青線はバリデーションデータ(10 サンプル)のロス値を、下図は OK/NG 判別に用いる検証結果のロス値を示す。尚、波形切り出し起点は First-Impact に統一した。

結果は RP300, RP1200, RP2000 で大きな差異は見られず、どれも良好な分離性能を示しているが、この中では RP1200 が最も安定しているように見える。

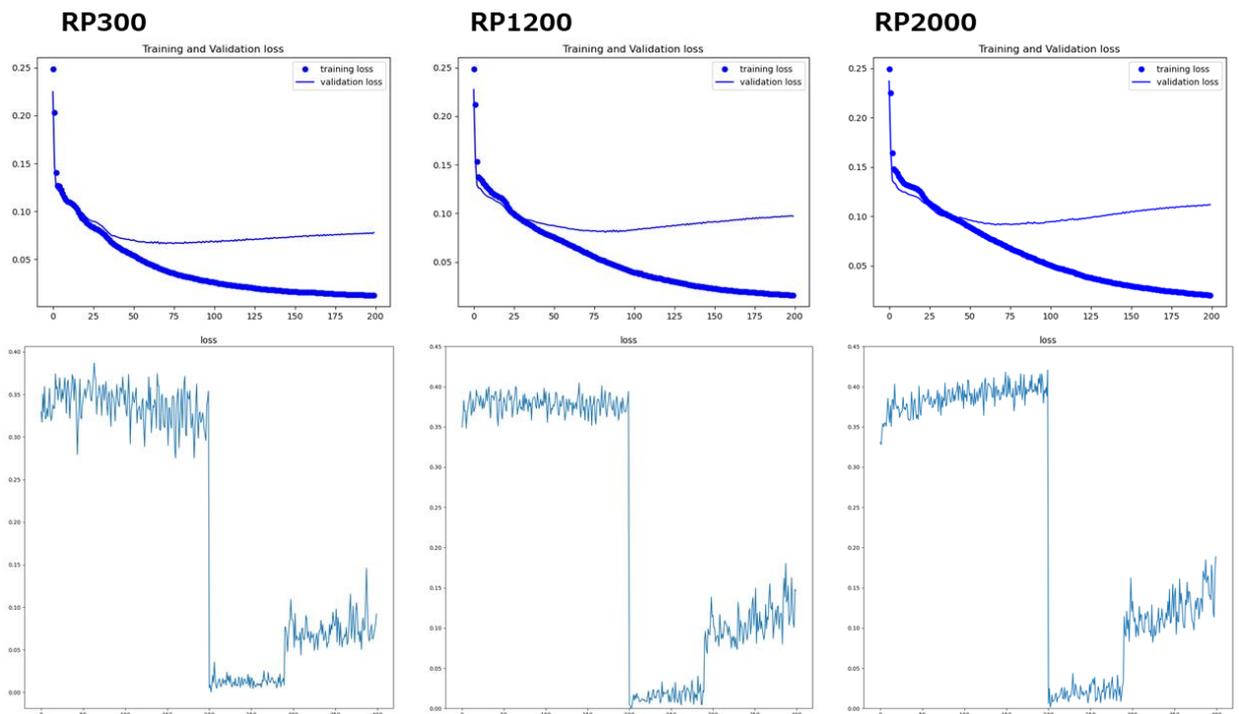


図 5.2-11 波形切り出し範囲の最適化

(h) DAE モデルと VAE モデルの比較 [図 5.2-11]

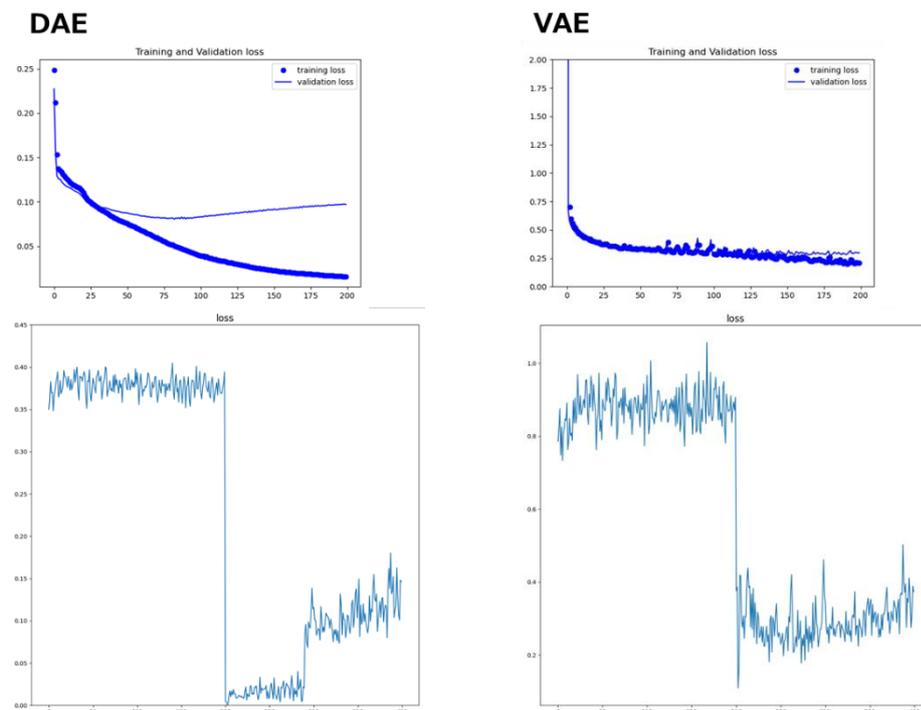


図 5.2-10 DAE モデルと VAE モデルの比較

OK/NG 分離性能が安定している RP1200 をベースに、OK/NG を判別する AI モデルについて DAE (Deep AutoEncoder) モデルと VAE (Variational AutoEncoder) モデルで比較した結果を図 5.2-11 に示す。それぞれ上図はトレーニングカーブで、太い青線はトレーニングデータ(90 サンプル)のロ

ス値、細い青線はバリデーションデータ(10 サンプル)のロス値を、下図は OK/NG 判別に用いる検証結果のロス値を示す。尚、波形切り出し起点は First-Impact に統一した。

VAE では上図のトレーニングデータとバリデーションデータのロス値がほぼ一致している事から、同一学習条件でもランダムサンプリング層により、過学習が起これ辛くなっていることがわかる。また、下図の検証結果でも学習に用いた OK データと検証に用いた OK データのロス値の差が小さい。DAE は学習対象の OK データでの過学習が若干見られるが、分離性能としては VAE より安定しているように見える。

(i) 前処理 & AI モデル性能評価 [図 5.2-12]

Huawei 社のエッジ AI ステーション Atlas500 を用いて、前処理の GAF と AI モデルの DAE の性能 (Latency) を評価した結果を図 5.2.9 に示す。

GAF の前処理時間 1.35ms, DAE の処理時間 4ms @Grayscale, 8ms @RGB と十分高速に処理できることが実証できた。

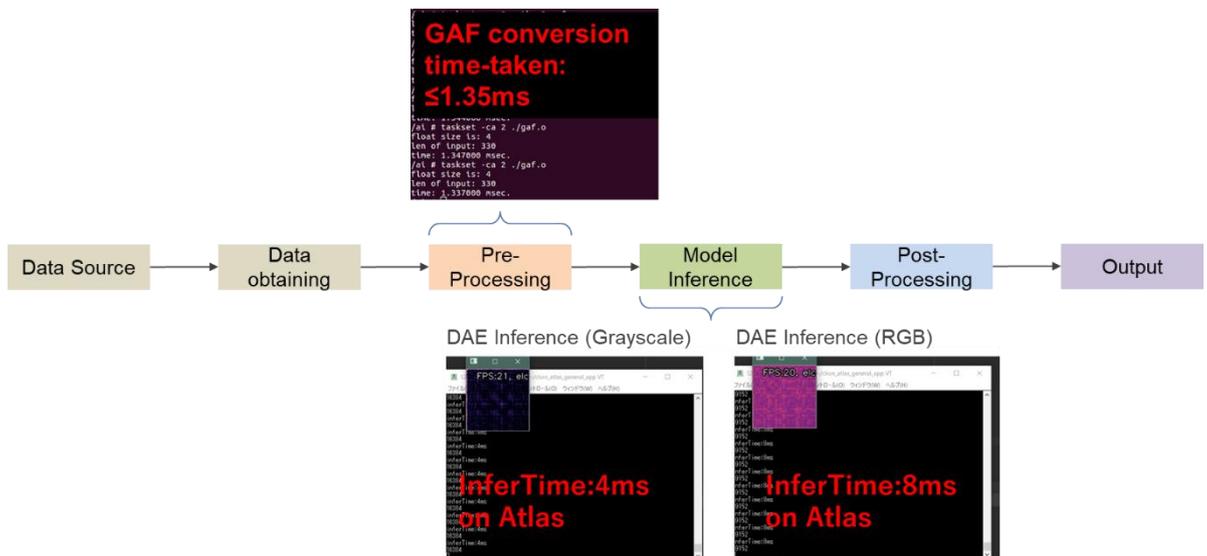


図 5.2-12 前処理 & AI モデル性能評価

(6) 結論

- (a) OK データ 100shot を使用して (OK データのみ) 学習した教師無しモデルで、残りの OK データ 100shot, NG データ 200shot を 100%の精度で判別できた。
- (b) GAF (Gramian Angular Field) と RP (Recurrence Plot) では RP の方が特徴量が多く、OK/NG の分離性能が良い。但し、RP の方が計算量大きい。
- (c) Ch3 波形の Max-Peak 起点の切り出しでは、Max-Peak 位置のバラツキにより特徴量のバラツキが大きくなり、結果として OK/NG の誤検知が発生した。対して First-Impact 起点の切り出しでは特徴量が安定し、結果として OK/NG の誤検知が改善される。但し、First-Impact 起点の方が OK/NG の分離性能が若干落ちる傾向がある。

考察として、OK/NG の波形は Max-Peak 位置の違いだけで判別できているだけの可能性もあり、その場合 OK データでもたまたま Max-Peak 位置がズレていると NG と誤検知する可能性がある。

- (d) 波形切り出し範囲としては、300~2000point が特徴量を良く表現できている。33000point は範囲が広過ぎる為、特徴量が薄まってしまい、OK/NG の分離性能も悪くなる。

考察として、金型がプレスした瞬間の最初の範囲(300~2000point)にセンサーデータの特徴量が凝縮されていると考えらる。

- (e) VAE は入力データのバラツキに対してロバストになるモデルではあるが、OK/NG 分離性能としては DAE の方が良好である。
- (f) Atlas500 上での DAE の処理時間は~4msec と非常に高速であることが実証できた。

### 5.3. 良否判定事例3

#### (1)対象データと解析目的

OK/NG データ 320 枚(OK=NG=160)(図 4.3-1#1)を用いて学習モデルを作成、これを 94 万ショット長期データで良否判定を実施した。

#### (2)データ前処理

図 5.3-1 に示すように 3ch, 6ch, 7ch を切り出し 元データに時間平均を施し、各 1/10,1/100 に圧縮した。次に各チャンネルを 4096, 8192, 16384 で割り正規化した。



#### (3)学習

学習データとして、OK/NG データ 320 枚(OK=NG=160)を、評価データとして 94 万ショット長期データ 1800 枚(図 4.3-1#2)を使用した。学習アルゴリズムは CNN、学習環境は Sony Neural Network Console (NNC) [3]を利用した。

図 5.3-1 データ前処理

#### (4)推論結果

推論結果を表 5.3-1 に、使用したネットワーク構成を図 5.3-2 に示す。

表 5.3-1 推論結果

条件\結果		OKデータ && OK判定
構成	圧縮	
(A)	1/10	50.0% → 99.3% (*)
	1/100	97.2%
(B)	1/10	99.2%
	1/100	82.9%
(C')	1/10	53.4% → 99.8% (**)
	1/100	97.8%
(C)	1/10	98.6%
	1/100	100%

- ・構成 (B') は、構成(B) から ResNet 風分岐を除いた構成。
- ・1 学習はデータをシャッフルさせ 3 回行い、平均値を出力。
- ・すべて 200 epoch。
- (\*) 全結合層前に、GAP 層を追加した。
- (\*\*) (\*)に加え、各層後に Dropout を追加した。

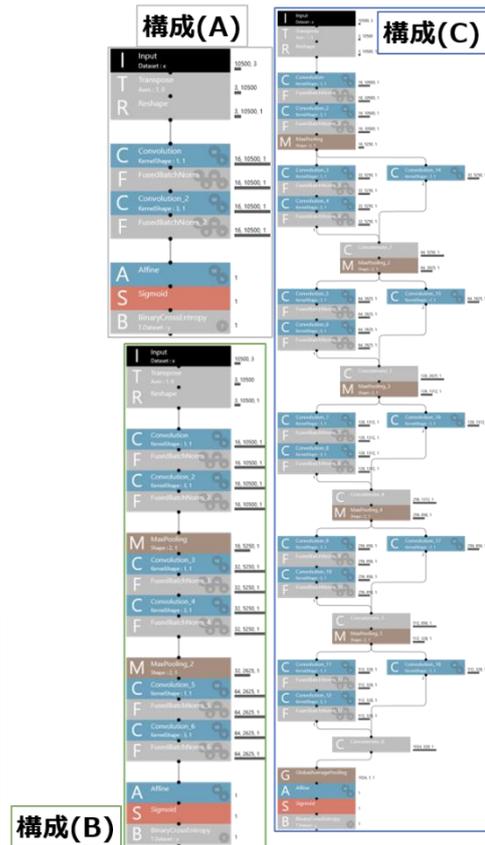


図 5.3-2 CNN ネットワーク構成

#### (5)まとめ

ネットワーク構成(C)が最も良い性能が得られた。OK/NG データと 94 万ショット長期データはデータ収集条件が異なるため同一の学習モデルの適用は困難と思われたが、大規模なネットワーク構成を用いることで高精度を達成した。OK データと NG データの差が大きいことが推定される。

## 5.4. 長期データ傾向分析事例1

---

### (1) 対象データと解析目的

プレス機の波形は複数の金型からの振動波が合成された波形と考えられる。長期的傾向を見る場合、金型に起因する波形のみを取り出すことが肝要と考える。本事例では DWT(離散型ウェーブレット変換)を用い分離した信号での長期的傾向を確認し、その有効性を検証する。

対象データは、94 万ショットデータ(図 4.3-1 #2) 及び 200 万ショットデータ(図 4.3-1#3)とする。

### (2) データ処理

データ処理の流れは図 5.4-1 の通りとなる。

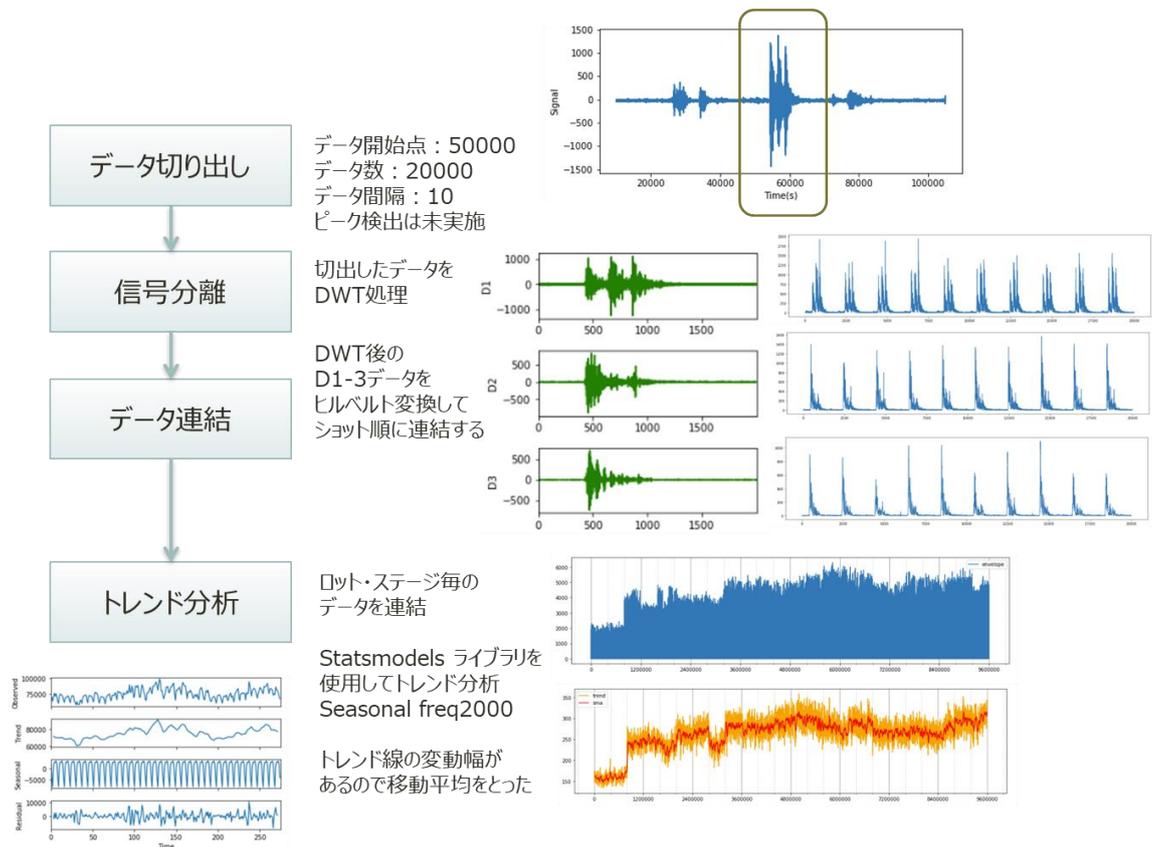
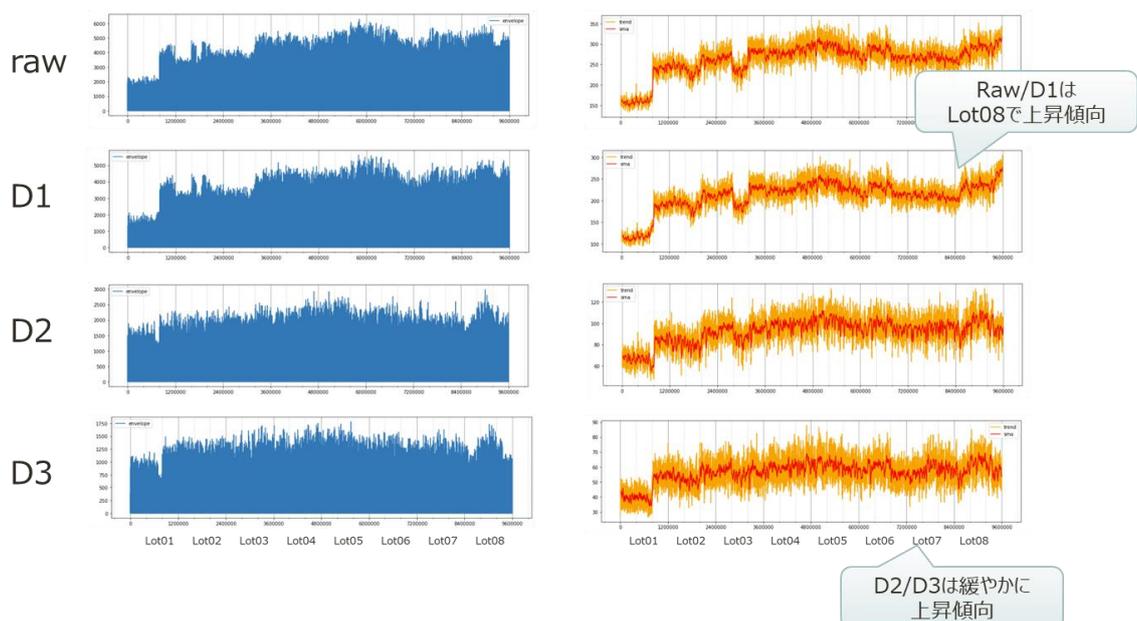


図 5.4-1 データ処理の流れ

(3) トレンド分析データ

トレンド分析データを図 5.4-2 に示す。DWT 処理前の raw データ 及び D1,D2,D3 を比較している。

90万ショットデータ (Lot01~08 ; CH03)



## 200万ショットデータ (Lot09~15 ; CH02)

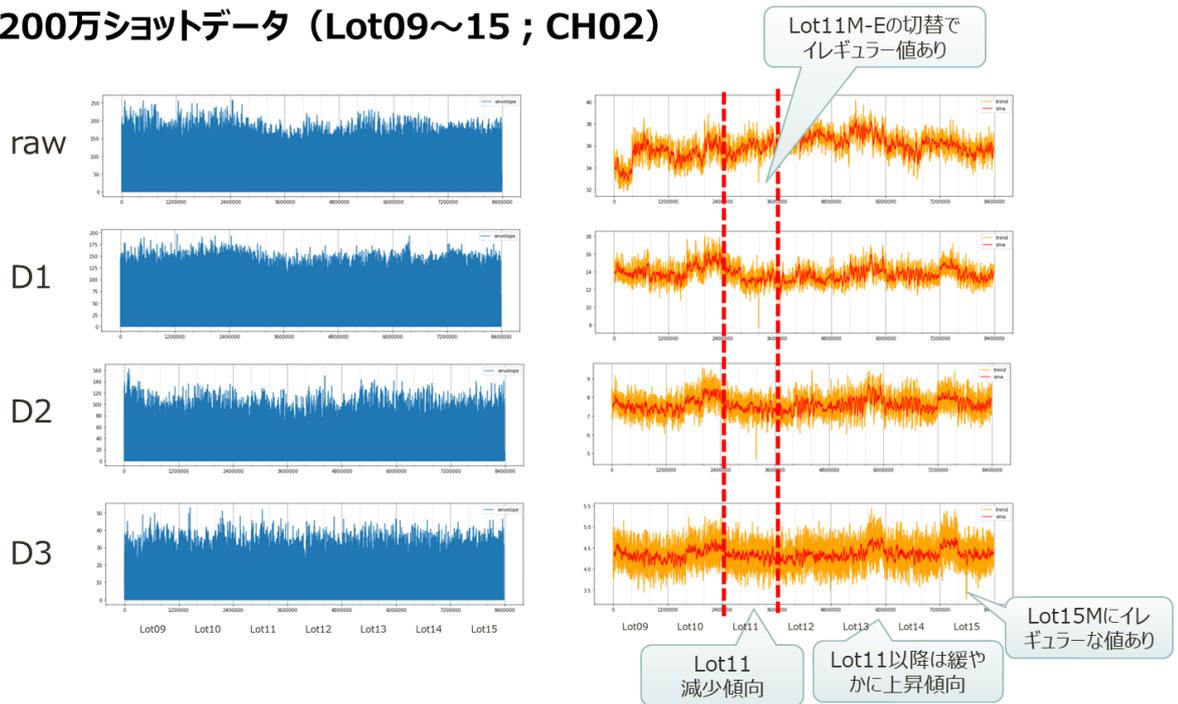


図 5.4-2 トレンド分析データ(90 万ショット、200 万ショット)

90 万ショットデータでは、全期間を通して右肩上がりの傾向がみられる。200 万ショットデータでは、Lot 間で大きく傾向が変わる点があることが見て取れる。また、DWT で分離した信号については、それぞれ傾向が異なり、この意味付けができれば長期データ分析の要素として活用できる可能性もある。ただし、どちらも本データだけでは、装置の傾向なのか金型の傾向なのかは判別がつかない。現場のデータと照らし合わせての解析が必要と考える。

#### (4) 時間経過の影響

取得したデータのタイムスタンプを確認すると、同一ロットであっても数日空いているケースが散見された。連続運転中の傾向と、間欠運転時の傾向とは当然異なると考えられる。

図 5.4-3 上部の折れ線グラフは Lot 間の経過時間を示している。段差が大きいところは経過時間が大きいことを示す。この中で 12 時間以上のポイントを細線、48 時間以上のポイントを太線で示した。

## トレンド線と時間経過の関連性

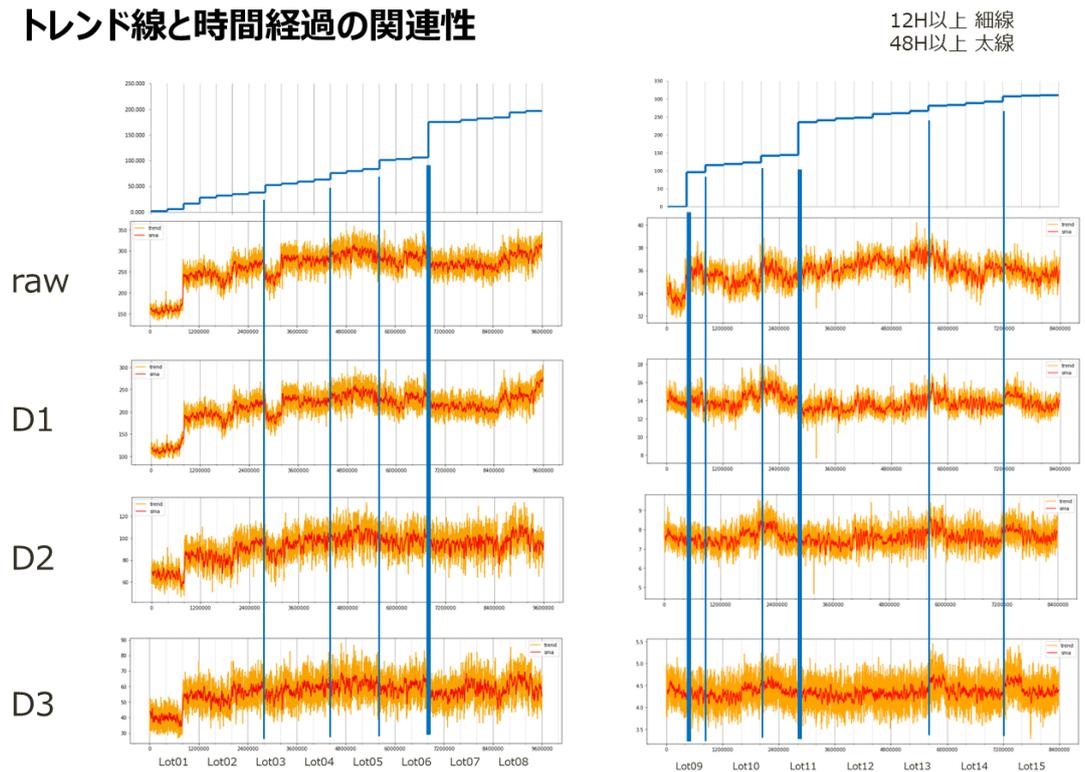


図 5.4-3 トレンド線と時間経過の関連性

Lot 間の経過時間が長いポイントで期間を区切ると、トレンド線の傾向には何かしらの規則性があるようにも見受けられる。時間経過とトレンドの影響分析に関しては、今後の課題と考える。

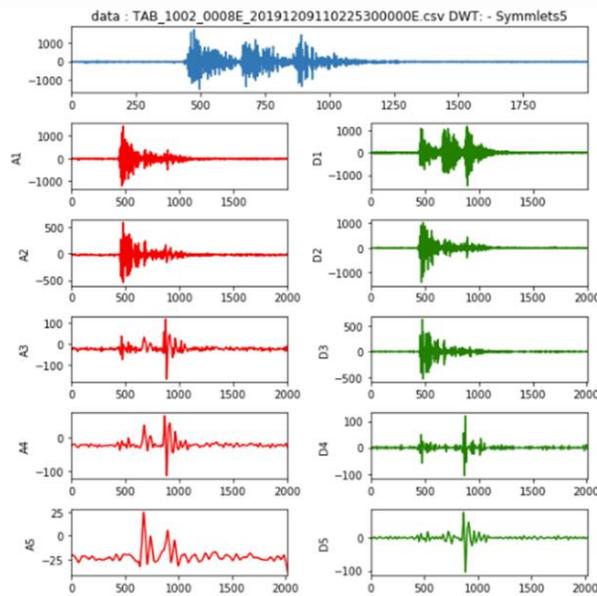
### (5) まとめ

長期データの傾向を把握するためのデータ分析手法について確認することができた。今度は本手法を用い、実際の加工現場での事象変化と照らし合わせた解析を行いたい。

DWT (離散ウェーブレット変換)

DWT とは、Discrete Wavelet Transform の略で、連続ウェーブレット変換 (Continuous Wavelet Transform) と異なり、スケール及びシフトを離散値として扱う。CWT が実信号データを詳細にモニタするために用いられるのに対し、DWT はデータを簡素化し扱いやすくする目的 (圧縮、ノイズ除去等) に用いられる。

今回の事例では、Python で使用できる PyWavelets ライブラリを用い、多重解像度解析による DWT にて、信号成分の抽出・分離を試みている。



$$S = A_n + D_n + D_{n-1} + \dots + D_1$$

参考: <https://pywavelets.readthedocs.io/en/latest/>

## 5.5. 長期データ傾向分析事例2

全ロットを通しての変化の傾向を見るために、ロット内の変化、ロットをまたいだ変化の可視化を試みた。対象データは、200万ショットデータ(図 4.3-1#3)とした。図 5.5-1 のように、S, M, E各ブロックの前半、後半各 100 ショット波形データを連続ウェーブレット変換したものを平均化し、Lot9-S に対する変化率をカラーマップ化した。

各ロットのS,M,Eをウェーブレット変換し、それぞれ平均化  
各ロットにおいてSは199ショット M,Eは各200ショット

Ch 1, 2, 3, 4, 5

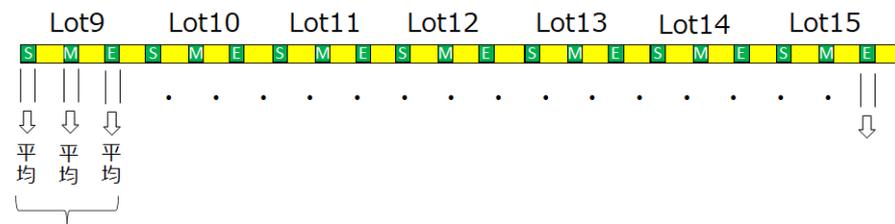


図 5.5-1 各ロットのS, M, E各ブロックを平均化

カラーマップは、変化率に対し、10 の対数を取ったものを-1から1(0.1倍~10倍)の範囲とし、赤が振幅の増加(最大10倍)、青が振幅の減少(最小0.1倍)を表している。

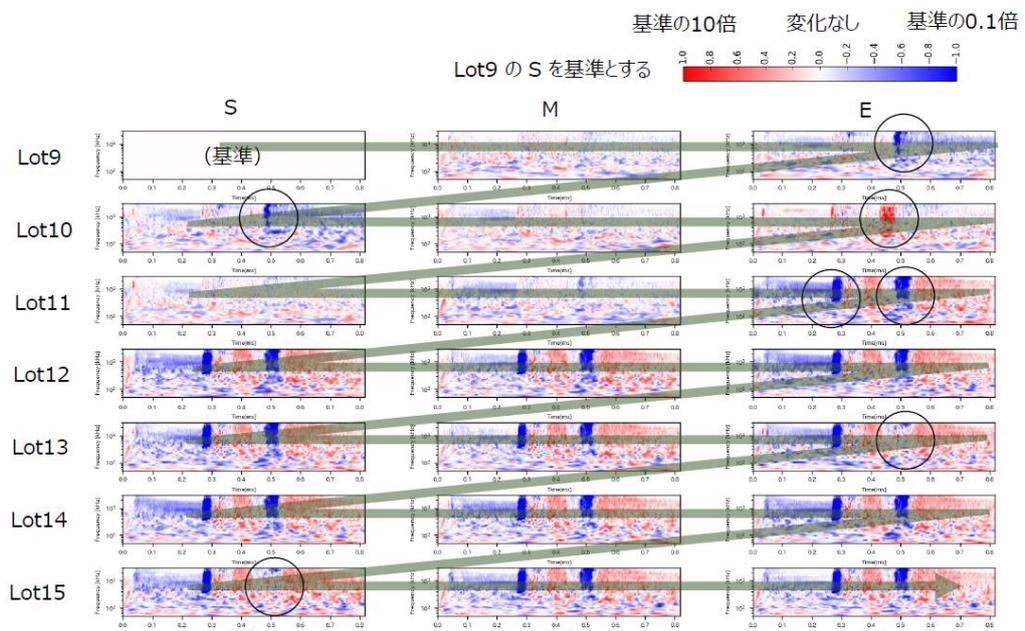


図 5.5-2 Lot9-S に対する変化率

図 5.5-2 は、Ch1 の Lot9 から Lot15 の変化の様子であるが、丸で示した個所に大きな変化が見られ、また、同一個所(時間-周波数)で増加、減少が繰り返されていることが解った。この可視化方法では、Lot

が進むことで徐々に変化している個所を見出せることを期待したが、このカラーマップを人の目で見た限りにおいては、連続した増加傾向、連続した減少傾向を見出すことはできなかった。

次に、ロットの進行に対して、相関の強い個所を数学的に見つけることを試みた。時間-周波数の全ポイントについて時間軸との相関係数を求め、相関係数の大きさにより、時間に対して相関の強い個所を抽出した。その結果、ch1 においては、相関係数  $R=0.93$  の強い相関を示すポイントが抽出できた (freq=590.9kHz, time= 0.3184) (図 5.5-3)。

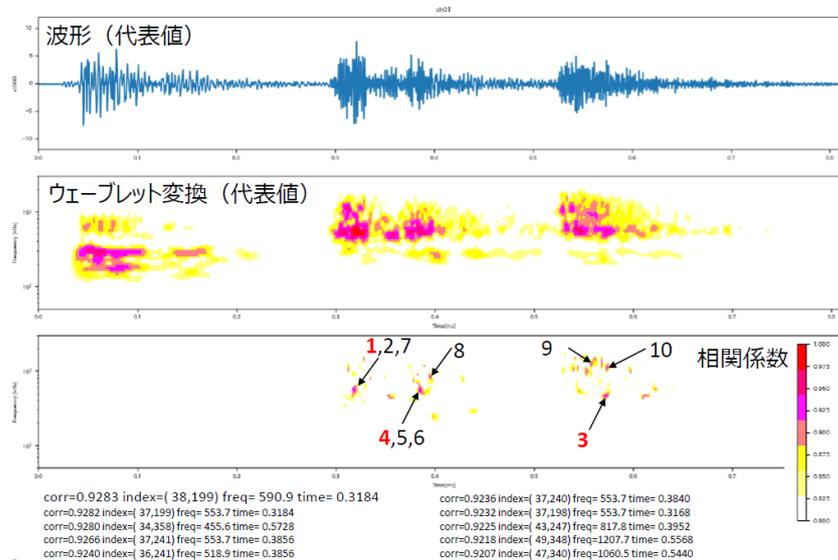


図 5.5-3 時間軸との相関の強いポイント(時間、周波数)の抽出

このポイントの波形の周波数成分の全ロットを通しての変化を下図 5.5-4 に示す。上が、S,M,E 各ブロックの平均値の変化の傾向であり、下は1ショットごとにプロットしたものである。平均値の変化の傾向は、直線に近い形で変化しており、1ショットごとにプロットしたものについても、一定のばらつきはあるものの、明らかに直線的に変化していることが解る。

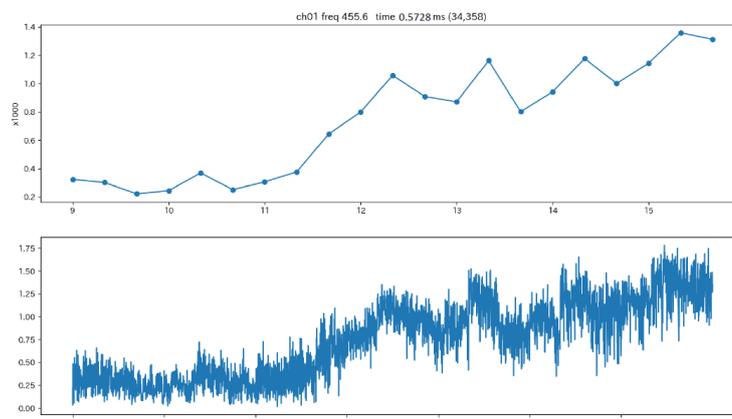


図 5.5-4 相関の強いポイントの振幅の推移 (Lot9~Lot15) 上:ブロック平均、下:1ショットごと

この変化が型の劣化に起因するものか、その他の物理的な変化に起因するものかについての検証は今後の課題であるが、時間-周波数領域の膨大なデータから時間経過とともに一様に変化するポイントを見つける手法として、今回の手法は有効な手段のひとつであることが解った。

S, M, E間、ロット間での変化について、何らかのパターンかどうかをクラスタリングの手法を用いて分析した。特徴量としては、波形を以下の図に示すようにA, B, Cの3つのパートに分け、その面積(絶対値の総和)と重心位置(時間方向)をショットごとに求めた(図 5.5-5)。

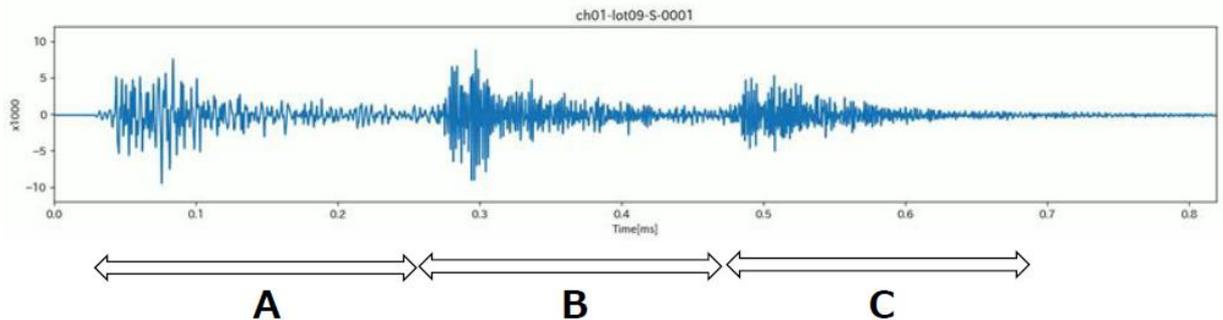


図 5.5-5 波形の3パートへの分割

図 5.5-6 に示すように、正逆の変化が繰り返され、丸で示したように階段状に変化している個所も見られた。それ以外の個所では大きな変化は見られなかった。したがって、面積、重心を特徴量としたとき、全波形はいくつかのグループに分けられることが解った。

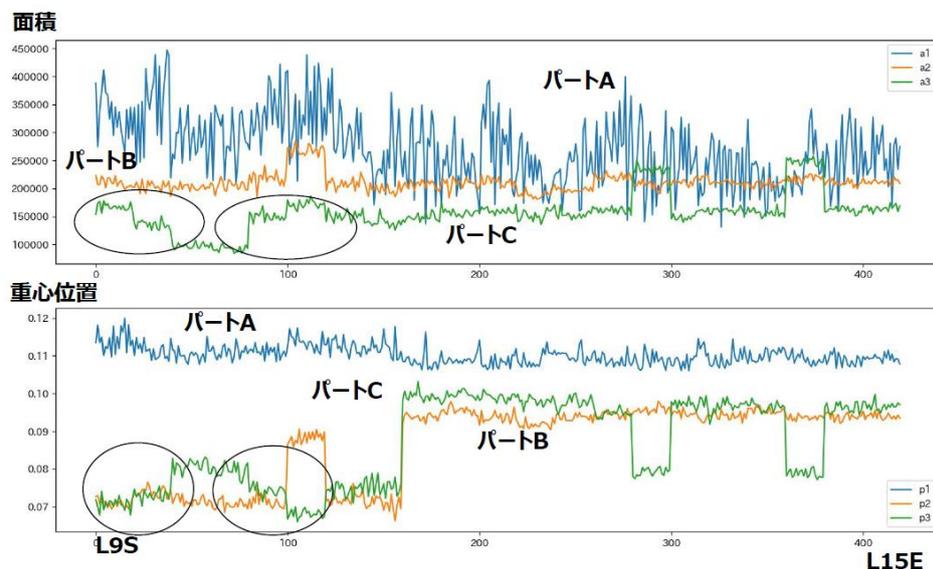


図 5.5-6 各パートの面積、重心位置の推移(Lot9~Lot15)

特徴量をグループ化する手法として、k-means 法によるクラスタリングの適用を試みた。k-means 法は、クラスタ数を最初に与えることが必要あり、結果を見ながら試行錯誤することが必要である。Ch1 のパ

ートCの場合は、クラスタ数を5個とすることで、人の直観に近いクラスタリング結果となった。各ロットのS,M,Eが、どのクラスタに属するか、および、変化の方向を矢印で示したものが図 5.5-7 である。

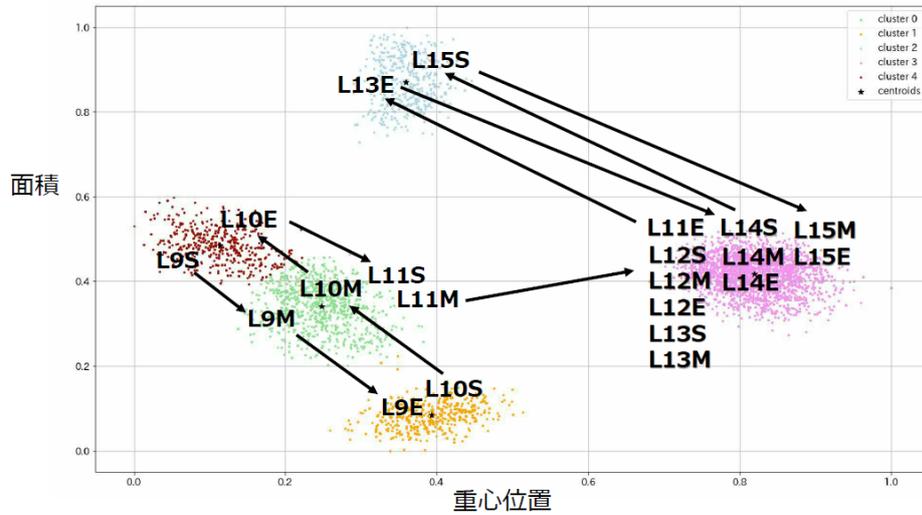


図 5.5-7 面積と重心位置を特徴量としたときのクラスタ分析

本解析により波形のグループ化する手法として、面積と重心を特徴量とすることが、有効な方法のひとつであることが解った。

連続ウェーブレット変換(CWT)

最小単位の波形(ウェーブレット)との相関度を見ることで、波形の周波数に適した区間でその波形の振幅、位相が得られる。何山のウェーブレット使うかで周波数、時間のどちらの分解能を優先するかを選ぶことができる。

短時間フーリエ変換(STFT)では、高周波の分解能を良くすると低周波数の分解能が悪くなるというトレードオフがあるが、CWT では、周波数に適したウェーブレットを使用するため、周波数によらず良好な周波数解析ができる。

Python 用ライブラリがあり、今回の分析では "swan" パッケージを使用した。

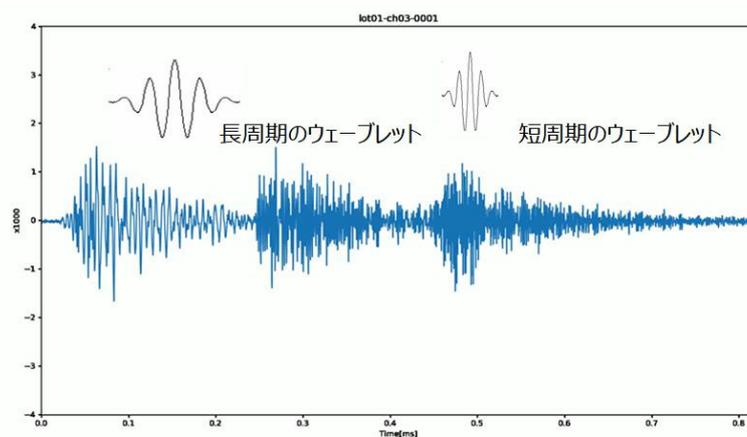


図 5.5-8 ウェーブレットと元波形

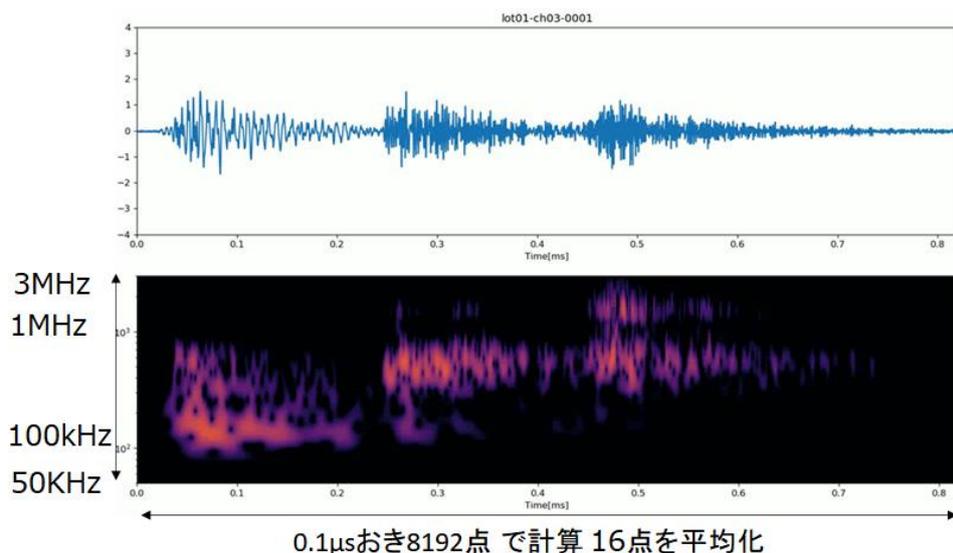


図 5.5-9 ウェーブレット変換結果の例(振幅の対数をカラーマップ化)

連続ウェーブレット変換(CWT)

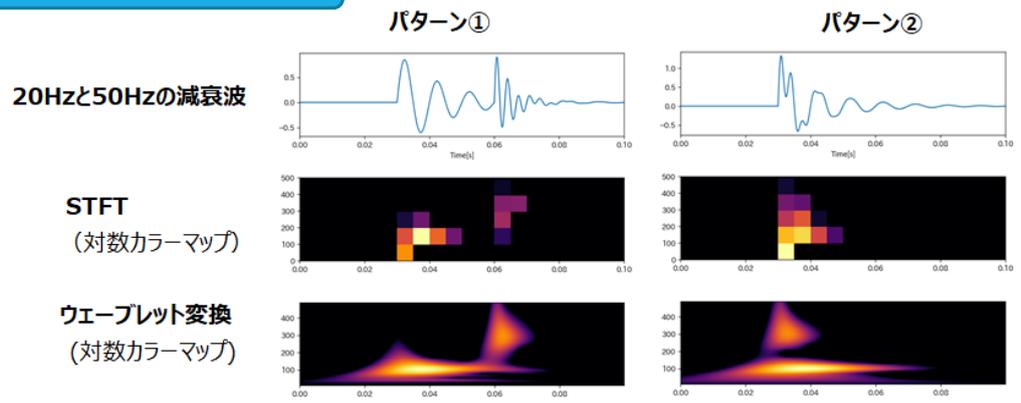


図 5.5-10 減衰波サンプルによる STFT、ウェーブレット変換比較

## 5.6. 長期データ傾向分析事例3

### (1) データ解析の目的、概要

本節では、長期データ(94万ショット(図 4.3-1#2)、200万ショット(図 4.3-1#3))の時系列の傾向変化を機械学習で解析し、数値解析(ウエーブレット等の周波数解析)との比較を行う。図 5.6-1 に評価対象データと解析手法を示す。

#### 《評価対象データ》

使用チャンネル	前処理	学習データ	テストデータ
94万ショット (CH3:AE(ピコ)超小型Φ4mm のAEセンサー)	RSM	Lot01_F (200ショット)	Lot01_F, Lot01_M, Lot01_E ~ Lot07_F, Lot07_M, Lot07_E
200万ショット (CH1:AE小型Φ15mmのAEセ ンサー)	RSM	LOT9_S (200ショット)	LOT9_S, LOT9_M, LOT9_E ~ LOT15_S, LOT9_M, LOT9_E

#### 《データ解析手法》

アルゴリズム	モデル	評価値
CAE	畳み込み5層+逆畳み込み5層	入力値と出力値の平均二乗誤差(復元エラー) ※200個の平均値、異常判定で増加、評価値は出力値に適宜係数を乗じた

図 5.6-1 評価対象データとデータ解析手法

### (2) データの前処理

図 5.6-2 に示すように、元データのうち注目する波形(中央付近の3つの山)に着目してデータ解析を行う。また、注目波形は3つの波形から構成され、波形ごとに分離した解析を実施する。

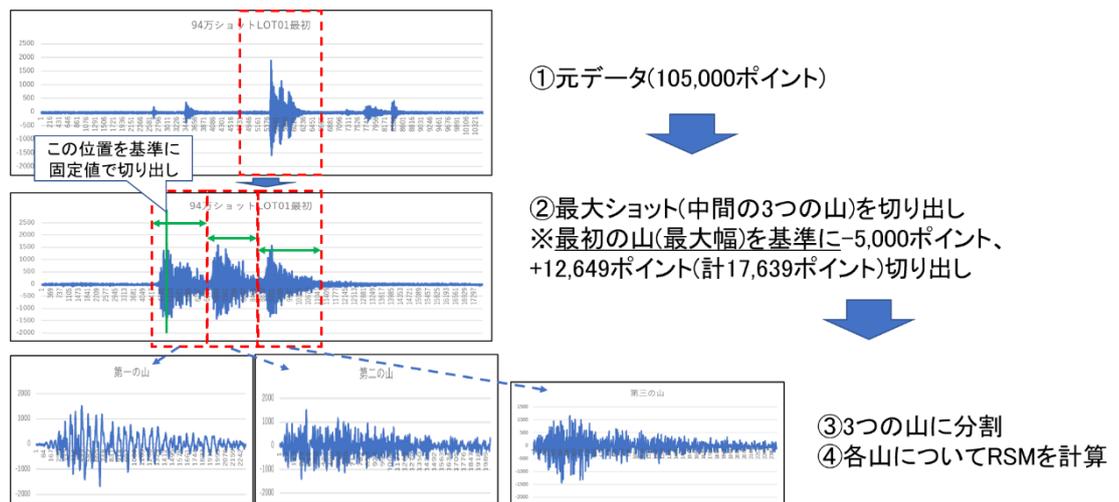


図 5.6-2 データの前処理手順

(3) データ解析結果(200 万ショット)

200 ショットデータに対して、3 山全体、第 1 の山、第 2 の山、第 3 の山のそれぞれのショットごとの解析結果を図 5.6-3 に示す。LOT9S のデータを学習データ(基準データ)として設定した。



図 5.6-3 200 万ショットデータの CAE 解析結果

ロットごとにデータ傾向の変化があることが確認できた。3山のうち、第3の山が傾向変化に寄与していることが確認できた。また、図 5.6-4 に 5.5 節で示したウェーブレット解析の結果を上記①～⑤との対比で示す。CAE 解析結果とウェーブレット解析の結果が①～⑤の変化点と対応していることが分かる。特に LOT11M から LOT11E の間で大きな傾向変化がみられる。

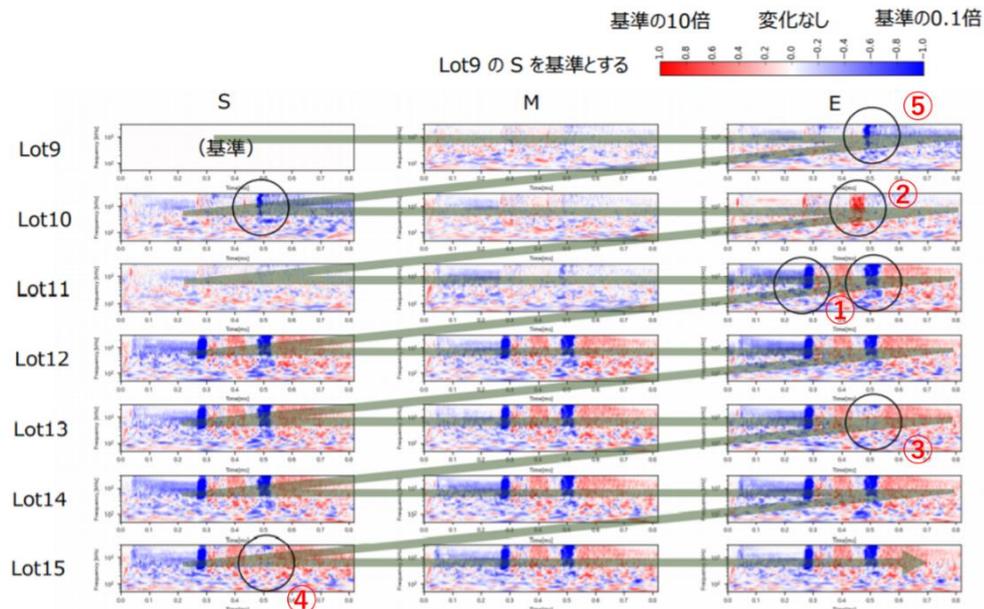


図 5.6-4 200 万ショットデータのウェーブレット解析結果

(4) データ解析結果(94 万ショット)

94 万ショットデータについても(3)と同様に図 5.6-5 に CAE 解析結果を、図 5.6-6 にウェーブレットの解析結果を示す(5.5 節では図示なし)。LOT1F のデータを学習データ(基準データ)として設定した。

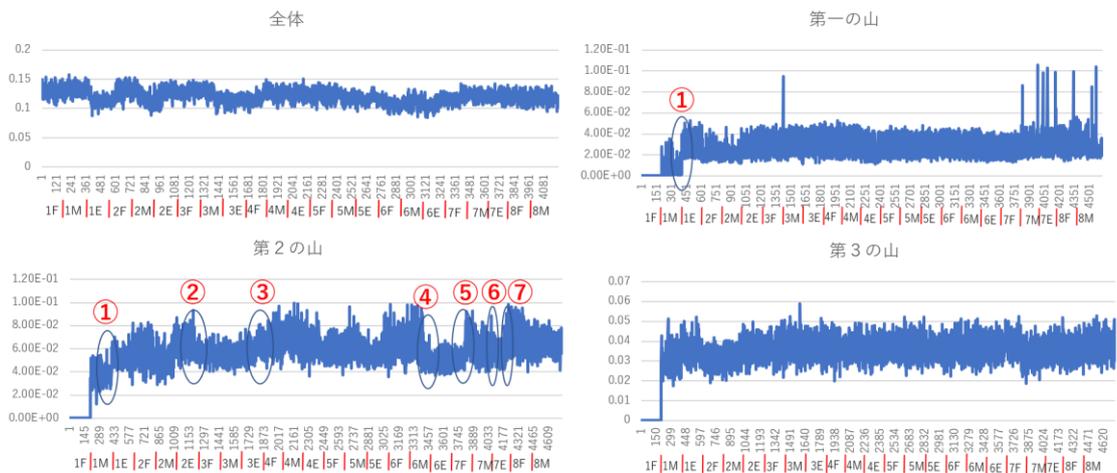


図 5.6-5 94 万ショットデータの CAE 解析結果

200 万ショットデータと変化のパターンは異なるが、CAE 解析結果とウェーブレットの解析結果は対応しており、主に第 2 の山が傾向変化に寄与していることが確認できた。

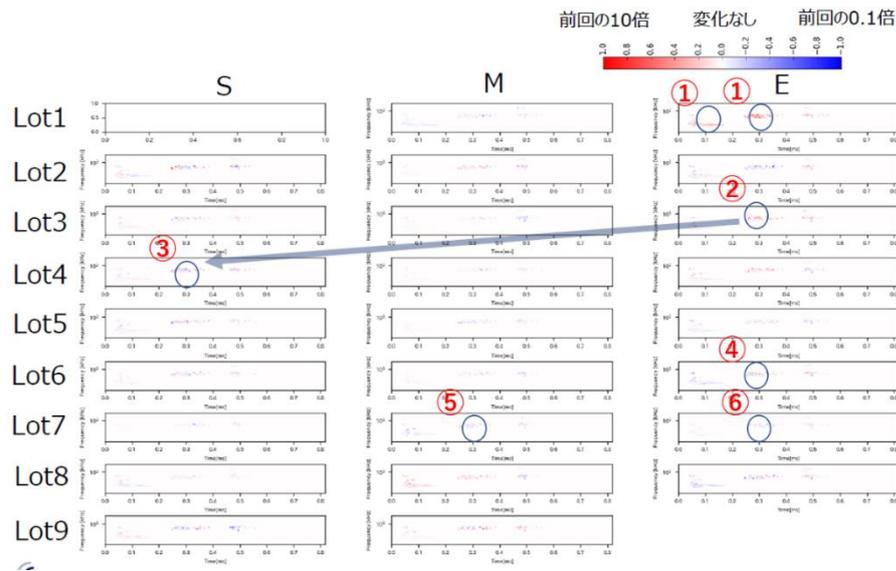


図 5.6-6 94 万ショットデータのウェーブレット解析結果

(5) まとめ

200 万ショットデータ、94 万ショットデータともに CAE 解析結果によるロット毎の変化はウェーブレット解析による傾向変化と概ね一致した。200 万ショットデータでは、第1の山はほとんど変化がなく、第2の山、第3の山に変化があること、94 万ショットデータでは第1の山、第3の山はほとんど変化がなく、第2の山、第3の山に変化あることが確認できた。これらの変化の要因は現時点では明らかになっておらず、今後の課題である。

## 5.7. 長期データ傾向分析事例4

本節では、図 4.3-1#4 のデータ解析結果を示す。図 5.7-1 に示すように周波数領域に明確なピークが存在せず、ウェーブレット変換は有効でない。また、波形位置が大きく変動するため、波形をパートに分けるクラスタリングも効果が期待できそうになかったため、本評価においては、全体波形を包絡線処理し、波形の差分から変化の傾向分析を行った。

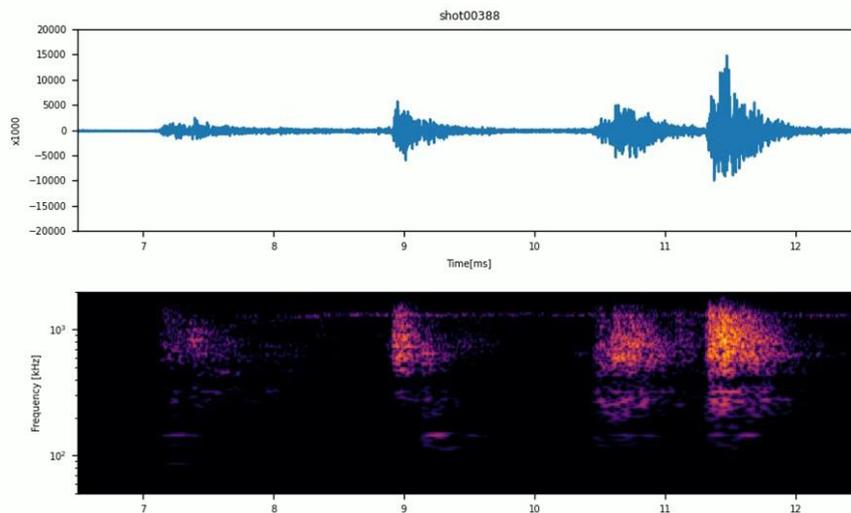


図 5.7-1 波形およびウェーブレット変換結果

ヒルベルト変換を用いた包絡線処理は、振幅の短時間の変動も忠実に表現できる反面、複数周波数の合成波形において、各ショットの周波数の位相の関係により振幅が重畳される場合と打ち消し合う場合があり、ショットごとに大きな変化を示す場合がある。したがって、ヒルベルト変換後、部分平均をとるスムージング処理を行うことで、波形の概略形状を表現することとした(図 5.7-2)。

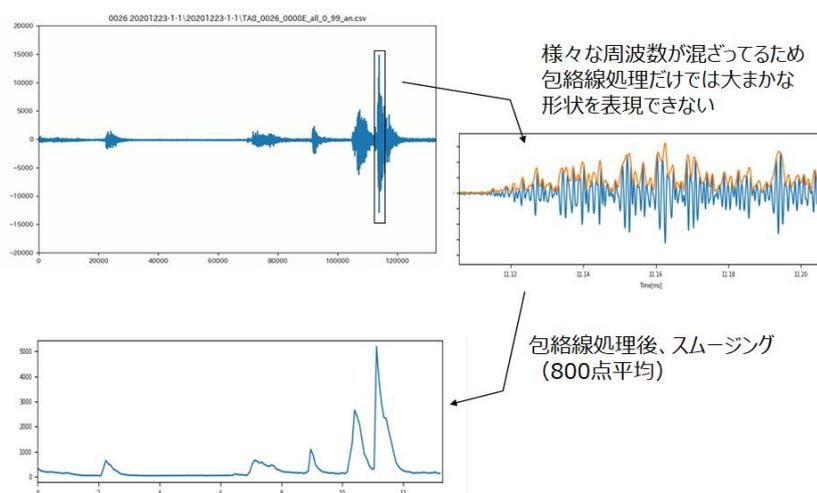


図 5.7-2 包絡線処理+スムージング処理

全データの平均を基準とし、各ショットの形状の差をユークリッド距離にて算出し、プロットしたものが下の図である(図 5.7-3)。

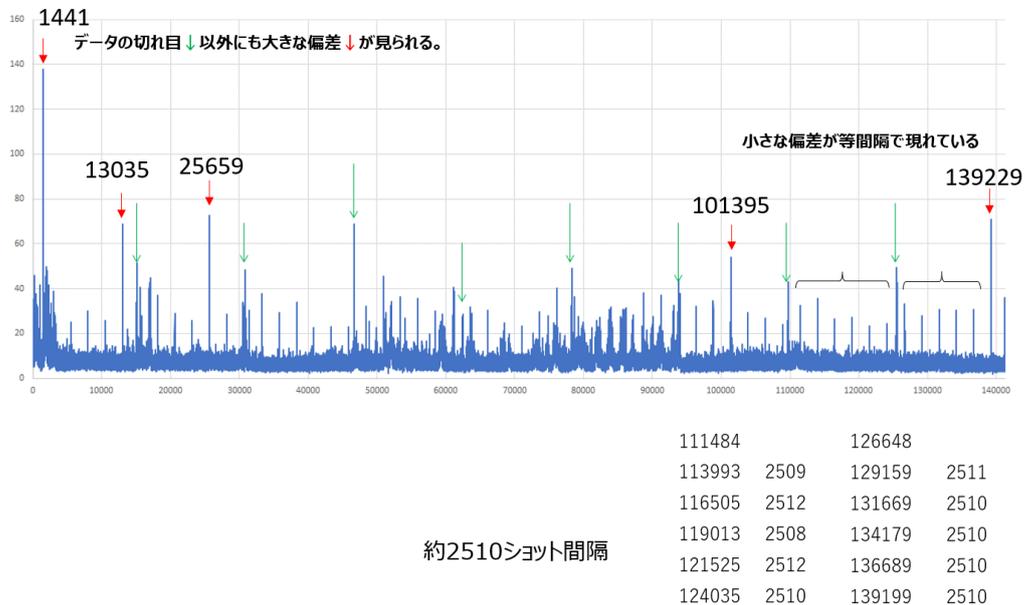


図 5.7-3 各ショットの平均値からの距離の差の推移

一定周期で突発的な変化が現れる傾向が見られ、その周期は約 17500 ショットごと、および、約 2500 ショットごとであった。この傾向を生産現場での作業と照らし合わせたところ、17500 ショットごとは、材料切れ、2500 ショットは、プレス装置から出てきた製品を入れるコンテナがいっぱいになり、それぞれ装置を一旦、止めたことに起因することが判った。

コンテナ入れ替え時の再起動では、プレスの回転速度が一定になるまでに、波形が時間方向に変化し、速度がふらつく傾向が見られた(図 5.7-4)。もし定常運転時にプレスの回転速度がなんらかの原因により変化した場合、これを異常として抽出できることが判った。

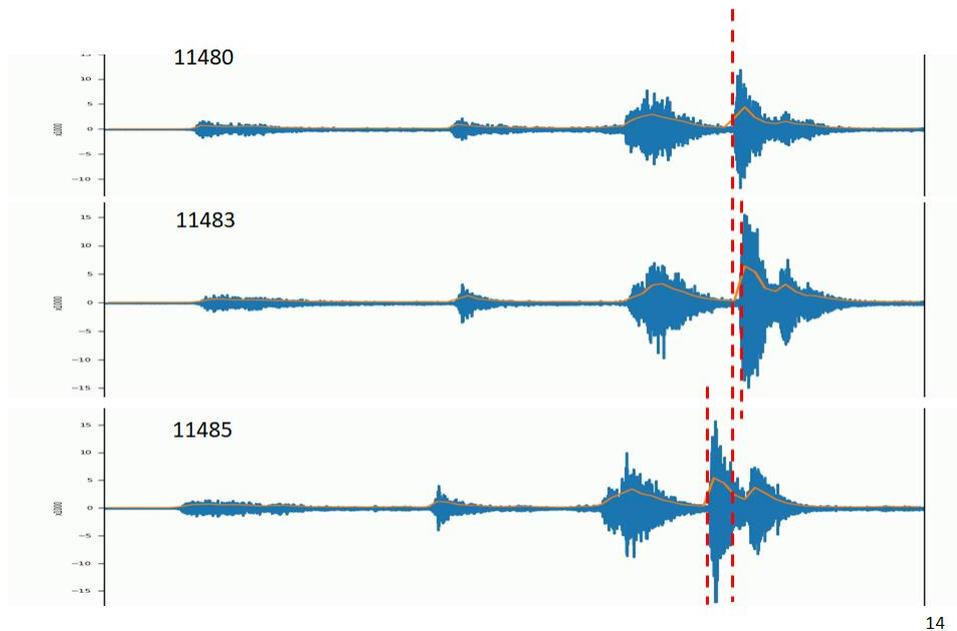


図 5.7-4 コンテナ入替時の再起動での波形の変化

## 5.8. 長期データ傾向分析事例5

本節では、図 4.3-1#4 のケースを 5.7 節とは別のアプローチで解析した結果を示す。本節では、6 つの波形特徴量を用いて解析する [4] [5] [6] [7]。

<p><b>実効値(RMS):</b></p> $V_{rms} = \sqrt{\frac{1}{T} \int_0^T v(t)^2 dt}$ <p>波の形をv(t)、周期をT</p>	<p><b>波高率:</b></p> <p>波高率 = 最大値/実効値 ※実効値に対するピーク値の比率</p>
<p><b>歪度:</b></p> <p>分布の非対称性を示す指標。値が正であると右に伸びている分布、逆に負であると左に伸びている分布である。</p>	<p><b>尖度:</b></p> <p>正規分布を基準とした分布の鋭さの指標。基準である正規分布の尖度よりも大きな値であれば分布は尖っており、小さな値であれば緩やかな分布である。</p>

図 5.8-1 解析に利用した特徴量

(1) 初期データ(27000 ショット)

図 5.8-2、図 5.8-3 にショット単位での各特徴量の推移を示す。運転開始から 2500 ショット程度までは不安定、13000 ショット付近、25000 ショット付近で突発的な変化がみられる(5.7 節と同様の結果)。

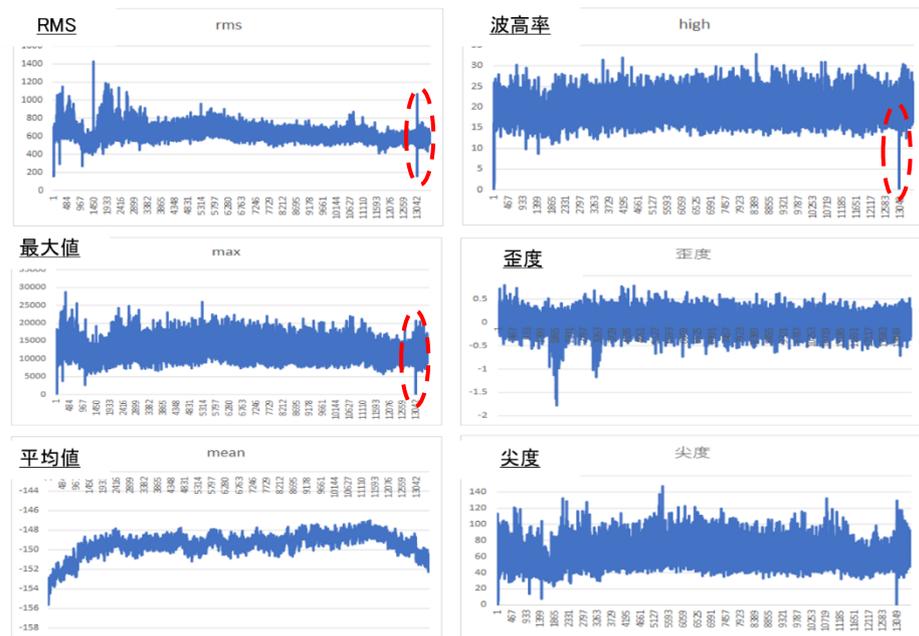


図 5.8-2 ショット単位の推移(1 ショット～13500 ショット)

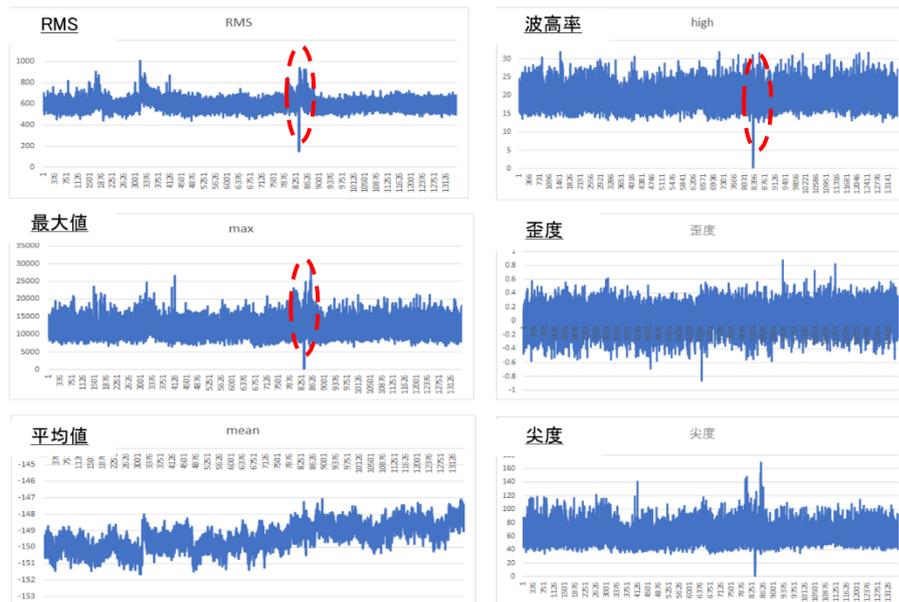


図 5.8-3 ショット単位の推移(22501 ショット～36000 ショット)

図 5.8-4、図 5.8-7、図 5.8-6、図 5.8-5 に t-SNE [5]によりショットごとの各特徴量の傾向変化を可視化した結果を示す。時間経過とともに一定の塊を作りながら傾向が変化していく様子が見えてくる(特に RMS、波高率が分かりやすい)。これらは、波形の変化点と対応している(詳細は各図参照)。なお、塊が大きく分かれるケースは、波形パターンの変化に対応していると考えられるが、変化の切れ目は明確には分からない。本解析結果は 5.7 節の解析結果を概ね裏付けられることが分かった。

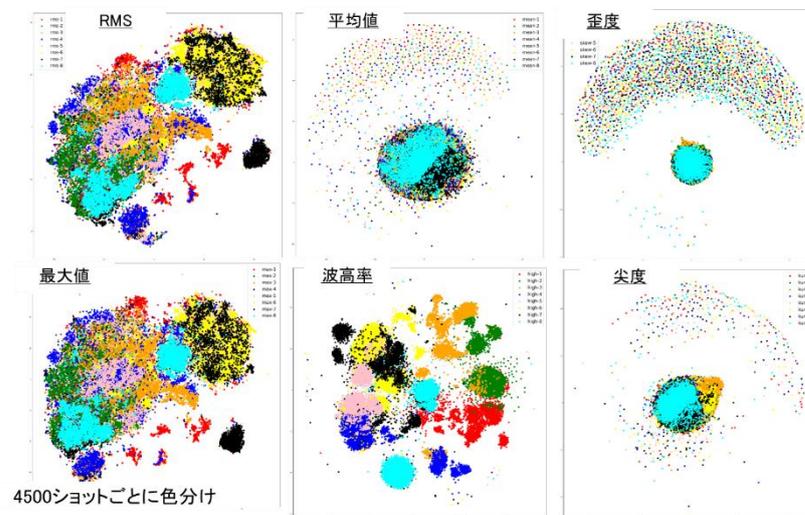


図 5.8-4 ショットごとの t-SNE による可視化(1 ショット～36000 ショット)

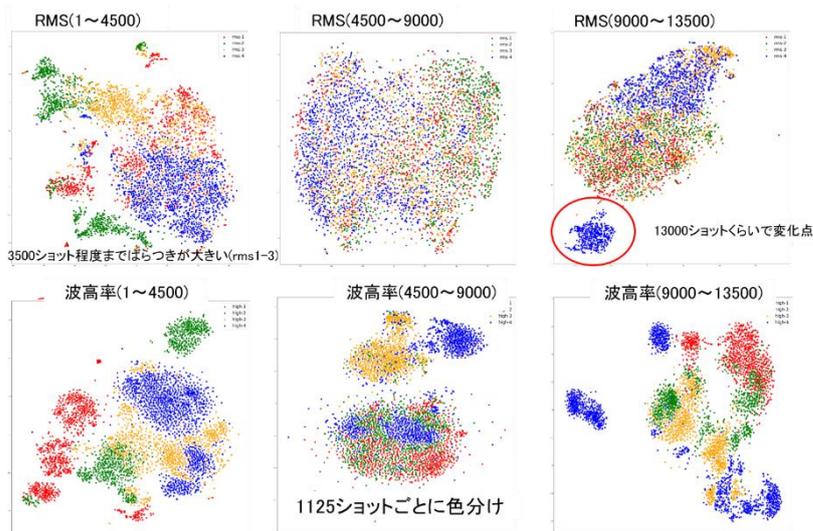


図 5.8-5 ショットごとの t-SNE による可視化(RSM,波高率:1 ショット~13500 ショット)

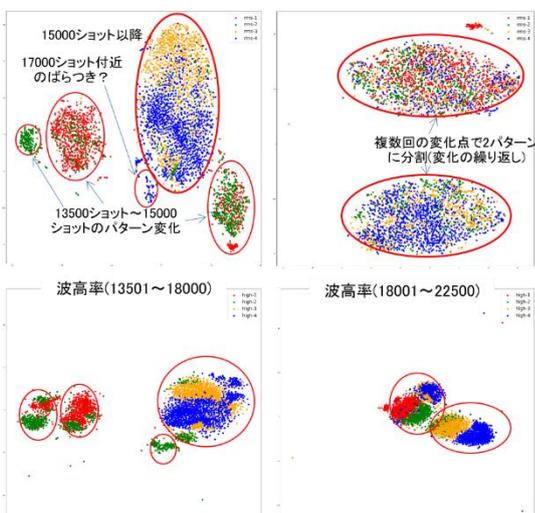


図 5.8-6 ショットごとの T-SNE による可視化(RSM,波高率:13500 ショット~18000 ショット)

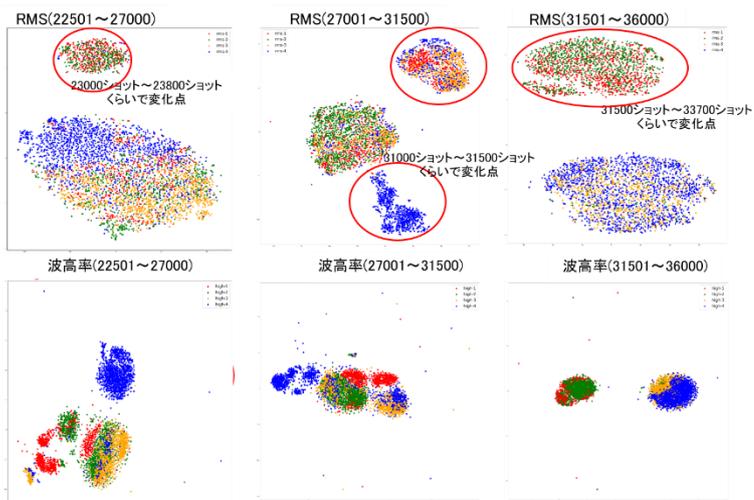


図 5.8-7 ショットごとの T-SNE による可視化(RSM,波高率:225001 ショット~36000 ショット)

## 6. データ解析高速化技術

### 6.1. 目的と方針

振動センサー、AE センサーなど周期的な信号をもとに良否判定、変化の予兆検知などを行う場合、生のデータをそのまま入力として扱くとデータ量が膨大となり処理時間が長くなる。また、少しの時間のずれによって波形の位相が変わってしまうため、特定時間の信号レベルを入力とするような処理では、正しい判定を行えない、または、あらゆる位相に対応するために多くのサンプルが必要となるなどの課題がある。

そのため、前処理として時間領域の信号を周波数領域に変化するフーリエ変換などを用いることが多いが、計算量が多くなるためリアルタイム性でのボトルネックになる場合や、深層学習など膨大なデータを扱う際に時間がかかるなどの課題がある。本章では、周波数変換を用いた代表的な手法をPC上で実行する際の処理時間を高速化する手法について評価を行った結果を報告する。

なお評価の範囲は、汎用性を重視し windows OS 環境において python のみを用いて実行できるものに限定した。さらに学習データなど多数のデータを扱う場合において、同一の処理を多数のデータに対して行う場合が多く、並列処理手法が有効であるため、複数データ同一処理の場合の効果についても評価を行った。

周波数変換を用いた前処理手法として評価したものは

- (1) 最も一般的な手法である高速フーリエ変換 (FFT)
- (2) 波形の包絡線処理手法のひとつであるヒルベルト変換
- (3) 任意周波数、任意時間の周波数分析を行うことができるウェーブレット変換

の3つである。

また、高速化の手法としては、

- (1) numpy+MKL, mkl\_fft (Intel 社が提供している数値演算ライブラリを利用)

(2) cupy(NVIDIA 社が提供している GPU を使用した高速演算ライブラリを利用)

(3) multiprocessing パッケージによるマルチコア動作

の3つである。また、一部、python 添付のパッケージを分析し冗長な処理を削除することで、より高速化を図る改良も行った。

本評価を行った際のマシンスペック、python 関連のバージョンについて表 6.1-1 にまとめる。

表 6.1-1 評価システムの構成

項目	メーカー	商品名、形式等	備考(おもな仕様など)
OS	マイクロソフト	Windows10pro 64bit	Ver 1903
CPU	インテル	Core i5-10400	2.9GHz、6 コア、12 スレッド
マザーボード	ASUS	TUF Gaming B460-PLUS	
メモリ	CFD	W4U2666CX1-8GB	2 枚で計 16GB
GPU	MSI/NVIDIA	GEFORCE GTX1650 AERO ITX 4G	1655MHz、メモリ GDDR5/4GB
ストレージ	Western Digital	WDS100T2B0C SN550 NVMe	m.2、SSD 1TB

項目	バージョン
python	3.6.8
numpy	1.19.5
numpy+MKL	1.19.5 ※1
mkl_fft	1.2.0 ※1
cupy	8.3.0
swan	0.7.1

※1: これらは、pip ではインストールできない。Unofficial Windows Binaries for Python Extension Packages として、インターネット上に存在するダウンロードサイトから入手

参考サイト: <https://www.lfd.uci.edu/~gohlke/pythonlibs/>

## 6.2. 高速化手法の解説

### (1) numpy + MKL, mkl\_fft

MKL は、Intel が提供する高速数値演算ライブラリ(ベクトル・行列演算、回帰分析、FFT 等が含まれる)

- ・Intel CPU が持つ SIMD (single instruction multi data)命令系列を使うことで処理を高速化
- ・独自の計算アルゴリズム適用(たとえば複数データの FFT で計算を最小化する初期化処理の分離)
- ・マルチコアによる高速化の適用

Python で使用する場合は、numpy 互換の numpy+mkl パッケージと mkl\_fft パッケージを大学等の Unofficial サイトよりダウンロードする。なお、numpy には、fft が含まれるが、numpy+mkl は、ベクトル・行列演算の高速化のみであり、fft は高速化されないため、別途 mkl\_fft が必要となる。

SIMD 命令系列は、一つの命令を CPU のレジスタ上の複数データに対して同時に適用するものであり、使用できる命令は CPU によって異なる。たとえば core-i5 は、レジスタ長が 256 ビットであり、32 ビット浮動小数点データを8個持つことができ、たとえば mm256\_add\_ps 命令で、2つのレジスタの8つの浮動小数点データがそれぞれ加算される(図 6.2-1)。

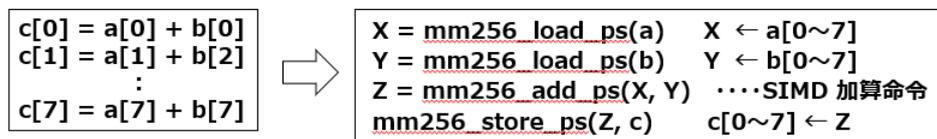


図 6.2-1 SIMD 命令を使用し8個の浮動小数点同士の加算を行う例

### (2) cupy

NVIDIA 社製 GPU を使うことで、データの並列処理を可能としたパッケージであり、関数インターフェースは、基本的には numpy 互換となっている。ただし、CPU 上のメモリと GPU 上のメモリは異なる領域にあるため、cupy.asarray, cupy.asnumpy 関数を使用してメモリの転送が必要となる。このメモリの転送は、PCI バスを通して行われるためオーバーヘッド大きく、単純な処理では高速化が図れない。GPU 側にデータを転送し、複数の処理を行ったのちに、最終結果を取り出すような場合において、処理が複雑になればなるほど大きな効果が得られる。

cupy を使用する場合は、通常は GPU のアーキテクチャは意識する必要はないが、高速化の限界を知るうえでは、使用している CPU の構成を理解しておくことが望ましい。図 6.2-2 は、今回使用した GTX1650 の内部構造である。

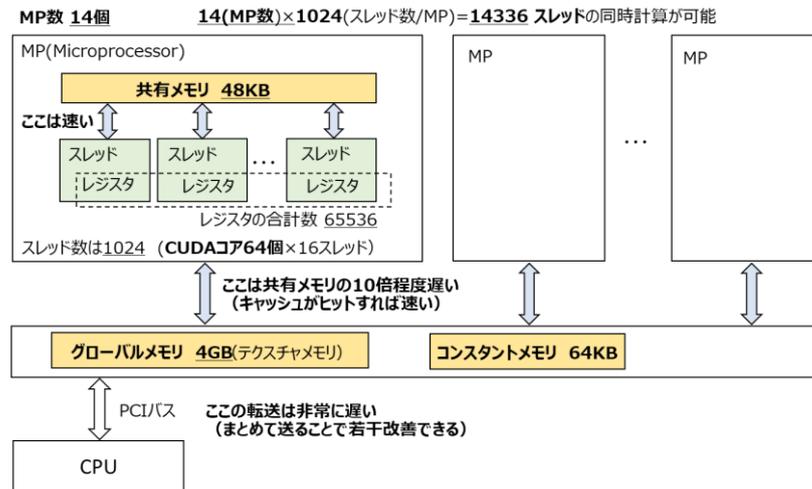


図 6.2-2 NVIDIA 社 GPU GTX1650 の内部構造

### 6.3. 処理時間比較処理時間比較

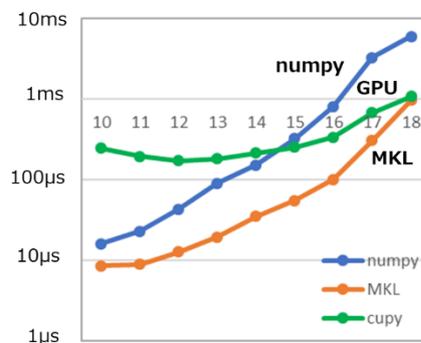
#### (1) FFT の速度比較

$2^{10} \sim 2^{18}$  のデータサイズで、numpy, MKL(mkl\_fft), GPU(cupy) を比較する(図 6.3-1)。

(以下の評価で「約 x 倍」は numpy に対する速度)

##### fft 1データ系列単位

MKL : 安定して高速化が可能  $2^{18}$  サイズで約6倍速い  
GPU : サイズが $2^{14}$ 以下では、逆に遅くなる  
 $2^{18}$  サイズでMKLと同等



##### fft 100データ系列単位

MKL : 安定して高速化が可能  $2^{18}$  サイズで約5倍速い  
GPU : MKL をわずかに劣る  $2^{18}$  サイズで約4倍速い

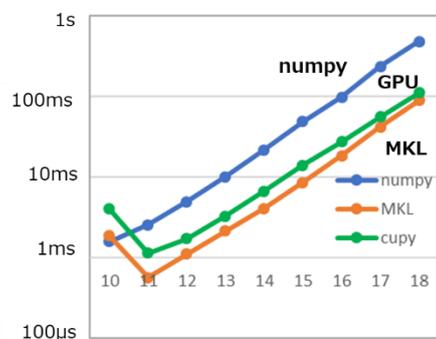


図 6.3-1 FFT 処理時間の比較(標準 NUMPY—MKL 使用—GPU 使用)

(2) ヒルベルト変換の速度比較

FFT と同様の比較を実施する(図 6.3-2)。

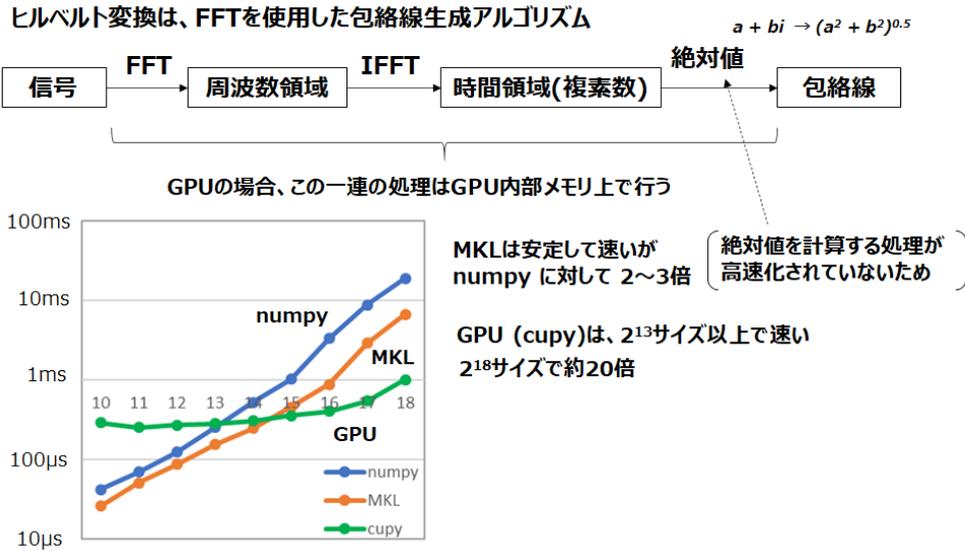


図 6.3-2 ヒルベルト変換処理時間の比較(標準 NUMPY—MKL 使用—GPU 使用)

(3) ウェーブレット変換の速度比較

Python のライブラリでインストールできるものとして swan がある。ウェーブレット変換の処理には、同サイズ、周波数の変換においては、1回で済む処理があり、複数データ系列の処理を行う場合は、この部分が無駄となる。

今回の評価は、swan ライブラリを参考に、初回のみで済む処理を外出しにする改善を行い、毎回必要となる処理時間の比較とした(図 6.3-3、図 6.3-4)。

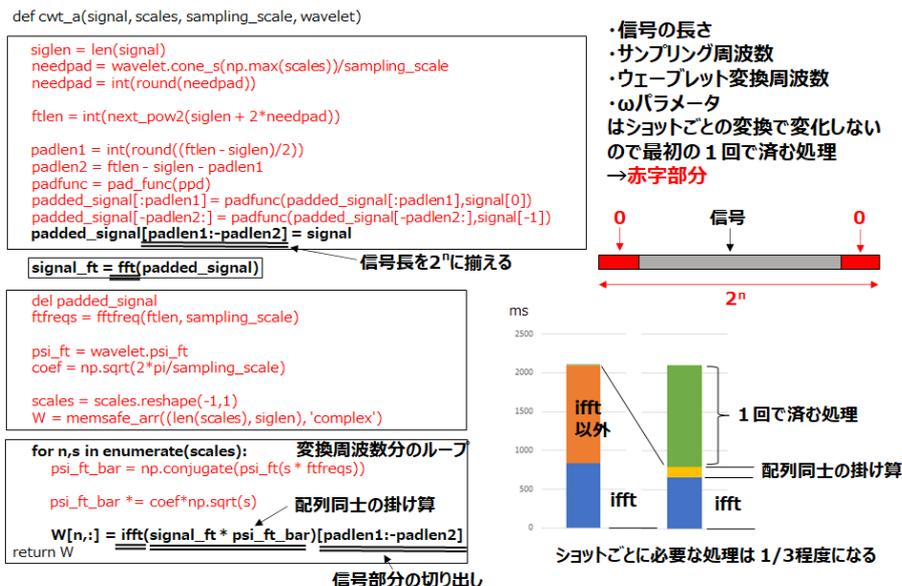


図 6.3-3 SWAN ライブラリ 初回のみで済む処理の外出し

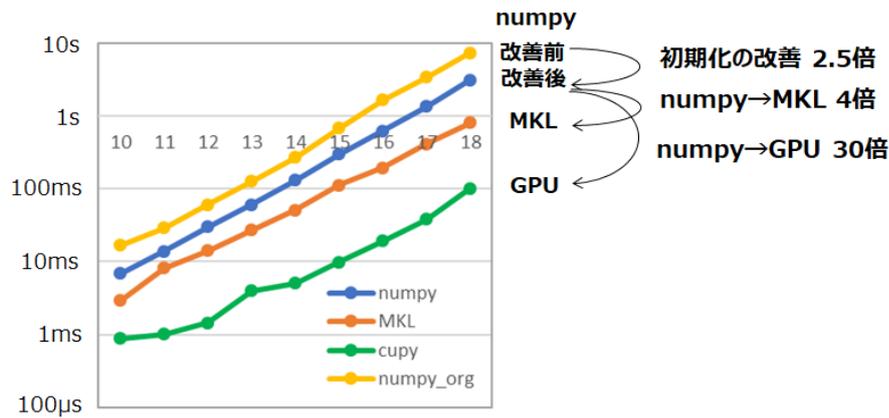


図 6.3-4 ウェーブレット変換処理時間の比較(初期化の改善効果、MKL、GPU の使用による効果)

ウェーブレット変換については、GPU を使用せず、CPU のマルチコアを使用した高速化の評価を行い、GPU の処理速度にどこまで近づくかを評価した(図 6.3-5)。なお、MKL では、マルチコアによる高速化が一部盛り込まれているが、効果が低いため(25%程度の高速化)、100 データ系列のウェーブレット変換を設定コア数に振り分ける並列化を python の multiprocessing モジュールを使用して行った。

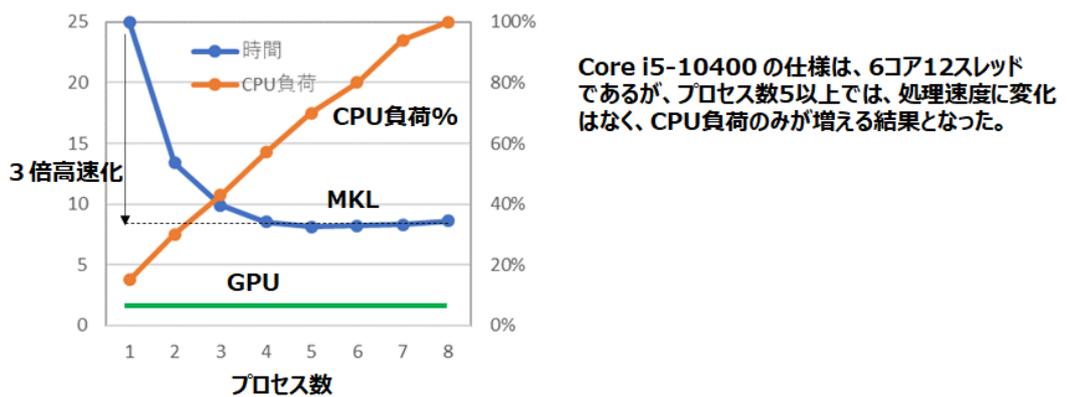


図 6.3-5 マルチコア化による高速化の効果

周波数変換のような、大規模なデータに対して同一処理を行う場合に、MKL、GPU とともに効果があった。とくに、複数系列データの処理をまとめて行う場合やヒルベルト変換、ウェーブレット変換のように、複数ステップからなる処理を GPU において連続で行なう場合においては、GPU は数 10 倍近い高速化が図れた。

各手法には、その方式やハードウェア構造からなるオーバーヘッドがそれぞれ違うところにあるため、効果を最大化するためには、手法の中身をよく理解することが必要であることが分かった。

## 7. 機械学習

### 7.1. CNN

CNN(Convolutional Neural Network) [6] [7]は「畳み込みニューラルネットワーク」といい、主に画像認識に利用されている機械学習手法の1つである。CNNのネットワーク構造は、図7.1-1に示すように「畳み込み層」「プーリング層」「全結合層」の組み合わせで構成される。

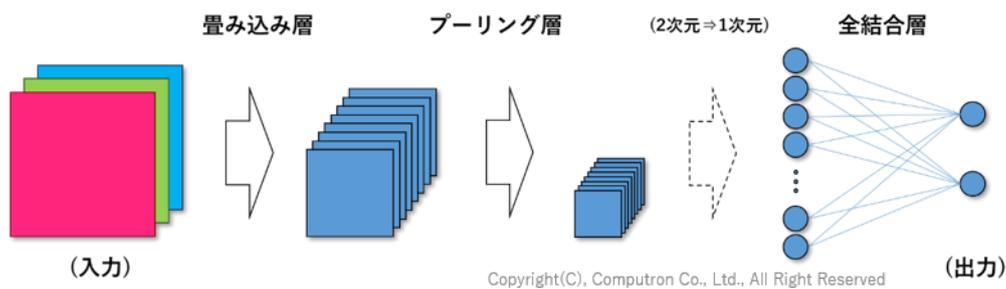


図 7.1-1 CNN のネットワーク構成

#### (a) 畳み込み層

畳み込みは、図7.1-2に示すように画像全体にフィルターをかけて、画像の特徴を抽出した小さな画像を取得し、複数のフィルターを用いることで、複数の特徴を得る。

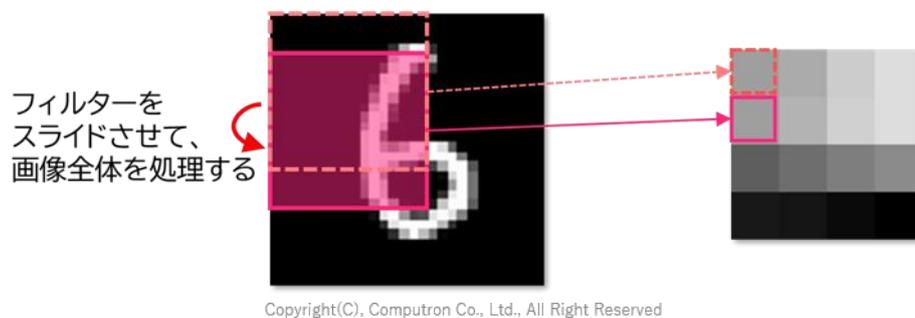


図 7.1-2 畳み込み層の処理

フィルターは図 7.1-3 に示すように数値として重み(W)とバイアス(b)を持ち、これらの値を使って計算することで畳み込みを行う。CNN は学習の過程で重みとバイアスを最適値(出力と正解値の誤差最小)に近づけるように調整する。

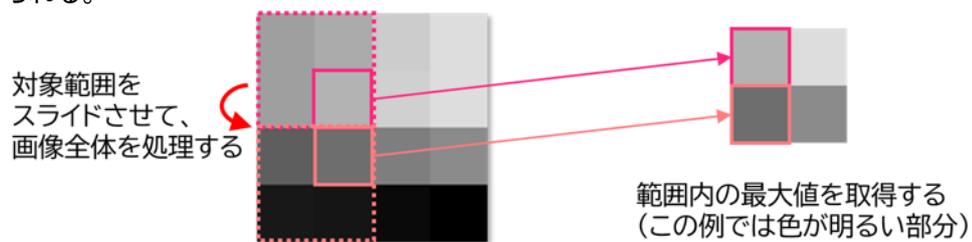


Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.1-3 畳み込み演算

(b) プーリング層

プーリングは画像内の位置ずれを吸収するための計算である。図 7.1-4 に示すように画像全体から一定の規則で画素を取り出すことで小さな画像を取得し、等間隔に最大値を取る最大値プーリングがよく用いられる。

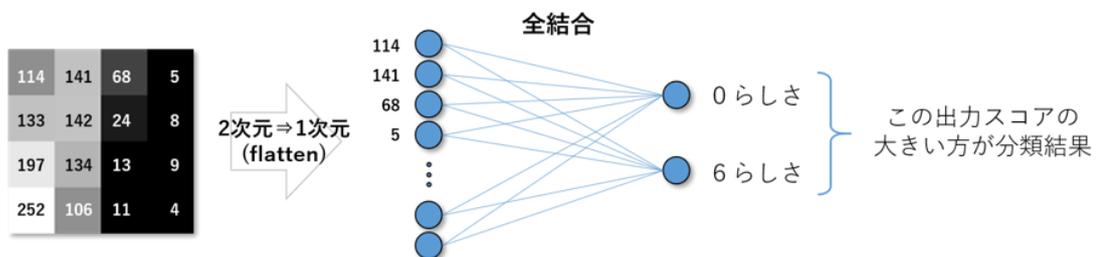


Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.1-4 プーリング層の処理

(c) 全結合層

全結合は1次元の数値の並びを扱う計算方法で、畳み込みによって得られた特徴をもとに分類を行う。図 7.1-5 に示すように分類したい数と同じ数のそれぞれの分類らしさを表す数値(確率)を計算して出力する。

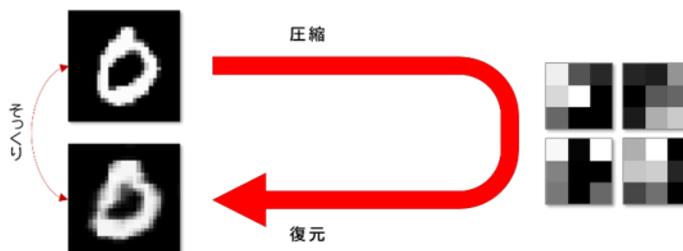


Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.1-5 全結合層の処理

## 7.2. AE/CAE

AE(Autoencoder) [8]は「自己符号化器」といい、ニューラルネットワークの一種。オートエンコーダには多くの派生があり、その総称としてもこの名前はよく使われる。まず、全結合という計算を使う基本的なオートエンコーダについて説明する。オートエンコーダは図 7.2-1 に示すように、なにか特定の分野について圧縮・復元ができるように学習する。例えば、0を学習したオートエンコーダでは1の圧縮・復元がうまく出来ない、など。



Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.2-1AE の処理

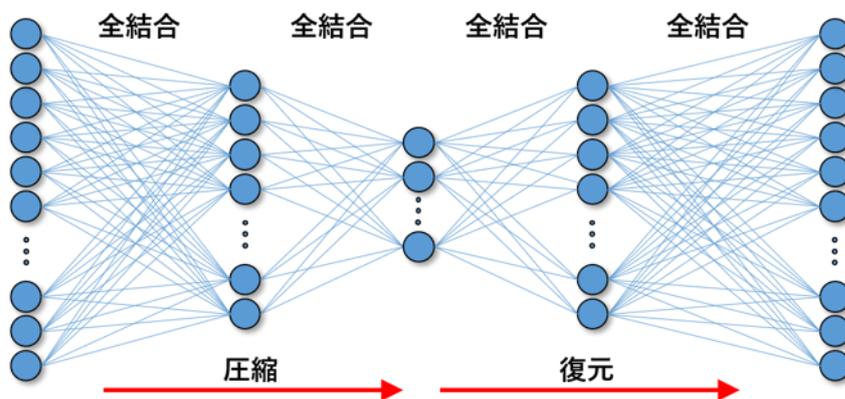
図 7.2-2 のように圧縮と復元という機能を活かし、オートエンコーダは様々な応用がなされている。



Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.2-2 AE の応用

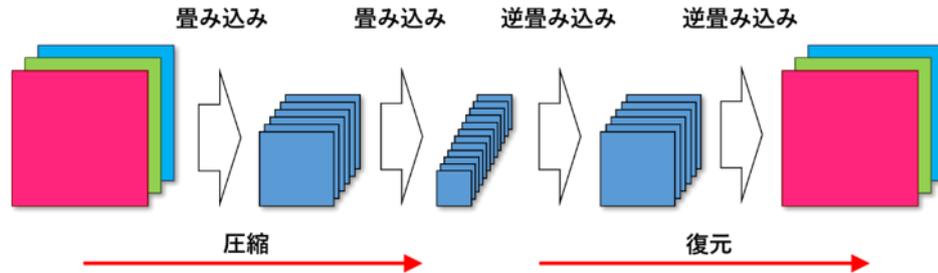
AE は、図 7.2-3 のように「全結合」を組み合わせて圧縮と復元を行う。



Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.2-3 AE のネットワーク構成

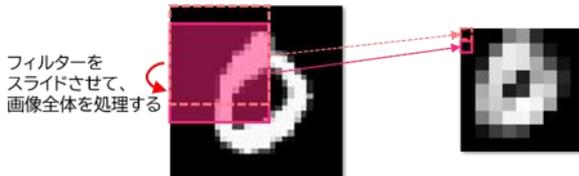
CAE(Convolutional AutoEncoder) [9]は「畳み込みオートエンコーダ」といい、畳み込みという計算を使うAEである。CAEもAEと同様、データの特徴を圧縮して復元するが、CAEのほうがより高い精度で圧縮・復元を行える。図 7.2-4 に示すように CAE では、圧縮、復元に畳み込みおよび逆畳み込みを行う。



Copyright(C), Computron Co., Ltd., All Right Reserved

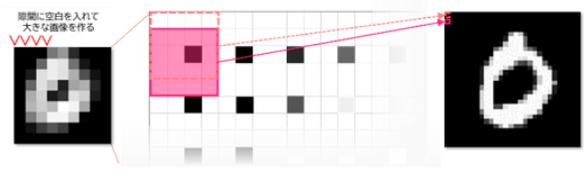
図 7.2-4 CAE の処理

データの圧縮で行う畳み込み処理は、画像全体にフィルターをかけて、画像の特徴を抽出した小さな画像を取得する。複数のフィルターを用いることで、複数の特徴を得る。一方、データの復元には逆畳み込みという計算を行い、画像に空白を入れてサイズを大きくした後、圧縮と同じように畳み込みを行う。(図 7.2-5 および図 7.2-6 参照)



Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.2-6 逆畳み込み処理



Copyright(C), Computron Co., Ltd., All Right Reserved

図 7.2-5 畳み込み処理

## 8. 課題と今後の展望

### (1) 取り組みの成果

- ① OK/NG 判定(OK200/NG200): 前処理/アルゴリズムの工夫で 100%判定を達成した。
- ② 予知保全(長期データの傾向分析):
  - ✓ 数値解析によりデータ傾向の概略は把握できた。
  - ✓ 機械学習によるデータ傾向変化の自動予測の見通しを得た。
- ③ データ採取時の課題の抽出:
  - ✓ ショット毎の頭出し、作業記録の充実、装置メンテナンス時のセンサーの取り外し、等。

### (2) 今後の課題

- ① データ傾向分析の詳細化(長期連続データ等ロットデータの追加)。
- ② データ傾向の変化に影響する要素の洗い出しと金型摩耗の特徴明確化。

これらの成果は、CIOF プラットフォーム用の AI エンジンとして API 実装を進め、Edge での異常判定を実現する予定です。

本書で解析に用いたセンサーデータは、株式会社ミスズ工業様よりご提供いただきました。  
ご協力に感謝申し上げます。



## 参考文献

- [1] 日本金型工業会, “日本金型工業会,” [オンライン]. Available: <https://www.jdmia.or.jp/>.
- [2] “アコースティック・エミッション,” フリー百科事典『ウィキペディア (Wikipedia)』, 5 日 3 月 2021 年. [オンライン]. Available: <https://ja.wikipedia.org/wiki/%E3%82%A2%E3%82%B3%E3%83%BC%E3%82%B9%E3%83%86%E3%82%A3%E3%83%83%E3%82%AF%E3%83%BB%E3%82%A8%E3%83%9F%E3%83%83%E3%82%B7%E3%83%A7%E3%83%B3>.
- [3] SONY, “SONY Neural Network Console,” [オンライン]. Available: <https://dl.sony.com/ja/>.
- [4] 福井健一, python と実例で学ぶ機械学習、識別・予測・異常検知, オーム社, 2018 年 11 月.
- [5] L. Maaten , G. Hinton, “Visualizing Data using t-SNE,” Journal of Machine Learning Research 9 (2008) 2579-2605, 2008.
- [6] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard , a. L. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Computation, Vol.1, No.4, pp.541-551, 1989..
- [7] Y. LeCun, L. Bottou, Y. Bengio , P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, Vol.86, No.11, pp2278-2324, 1998.
- [8] Salakhutdinov , G. H. a. R., “Reducing the dimensionality of data with neural networks,” Science, Vol.313, No.5786, pp.504-507, 2006.
- [9] A. Alqahtani, . Xie , M. J. Deng, “A Deep Convolutional Auto-encoder with embedded clustering,” 25th IEEE International Conference on Image Processing, 2018.
- [10] “実効値 フリー百科事典『ウィキペディア (Wikipedia)』,” 23 5 2020. [オンライン]. Available: <https://ja.wikipedia.org/wiki/%E5%AE%9F%E5%8A%B9%E5%80%A4>.
- [11] “尖度 フリー百科事典『ウィキペディア (Wikipedia)』,” 14 7 2020. [オンライン]. Available: <https://ja.wikipedia.org/wiki/%E5%B0%96%E5%BA%A6>.
- [12] “歪度 フリー百科事典『ウィキペディア (Wikipedia)』,” 20 7 2020. [オンライン]. Available: <https://ja.wikipedia.org/wiki/%E6%AD%AA%E5%BA%A6>.

## 著者およびメンバー

### 著者

---

- 秋元 一泰 (5.2 節を担当)  
秋山 智宏 (5.1 節、5.4 節を担当)  
河田 健一 (5.5 節、5.7 節、6 章を担当)  
平田 俊明 (1 章～4 章、5.3 節、5.6 節、8 章を担当)

### 研究会メンバー

---

- 平田 俊明 (コンピュータロン) サポート会員 : 主査  
後藤 宏二 (三菱電機) 正会員 : 副査  
秋元 一泰 (華為技術日本) 正会員  
秋山 智宏 (アンリツ) 正会員  
浅香 忠満 (AAC) 正会員  
石川 晴行 (華為技術日本) 正会員  
河田 健一 (ダイキン) 正会員  
佐藤 博義 (ADSTEC) サポート会員  
朱 厚道 (華為技術日本) 正会員  
藤田 泰則 (シグマトロン) サポート会委員  
松岡 康男 (東芝) 正会員  
渡辺 充昭 (新東工業)  
黄 奏連 (ダイフク) 正会員

### 協力者

---



杉浦 俊昭 (ニコン)

## センサーデータ活用の事例とその技術

---

発行者 一般社団法人インダストリアル・バリューチェーン・イニシアティブ  
理事長 西岡 靖之

〒102-0073 東京都千代田区九段北 4-3-28-302

電子メール: [office@iv-i.org](mailto:office@iv-i.org) URL: <https://iv-i.org>

発行日 2021年8月31日

定価 非売品

(発行者に無断で複製または印刷を禁止します。)



一般社団法人

インダストリアル・バリューチェーン・イニシアティブ