

博士学位論文

時系列情報を用いた工場設備の
異常検知及び故障予測

Anomaly detection and failure prediction
of factory equipment using time series information

2021年 2月

朝日 翔太

目次

第 1 章	序論	1
1.1	研究背景	1
1.2	研究目的	2
1.3	本論文の構成	3
第 2 章	異常検知と深層学習	4
2.1	異常検知とは	4
2.1.1	異常および異常検知の分類	4
2.1.2	異常検知における課題	5
2.2	深層学習ベース異常検知手法	6
2.2.1	深層学習ベース異常検知手法の分類	6
2.2.2	深層学習ベース異常検知における課題	9
2.3	時系列データにおける異常検知	10
2.4	時系列データを用いた異常検知における課題	10
2.4.1	時系列データを用いた異常検知分野における研究動向	11
第 3 章	時系列データにおける特徴抽出技術	13
3.1	特徴抽出手法	13
3.1.1	フレーム化処理	14
3.1.2	窓関数処理	14
3.1.3	離散フーリエ変換	15
3.1.4	振幅スペクトル	15
3.2	Principal Component Analysis; PCA	15
第 4 章	異常検知・故障予測における統計手法・機械学習および深層学習	17
4.1	従来手法による異常検知	17
4.1.1	Hotelling's T-square 法	17
4.1.2	Gaussian Mixture Model; GMM	19
4.1.3	K-Nearest Neighborhood; K-NN	19

4.1.4	One Class SVM; OC-SVM	20
4.2	Neural Network; NN	21
4.2.1	多層パーセプトロン	21
4.2.2	Recurrent Neural Network; RNN	22
4.2.3	Temporal Convolutional Network; TCN	26
4.3	深層学習による異常検知	27
4.3.1	Autoencoder; AE	27
4.3.2	LSTM Autoencoder; LSTM-AE	28
4.3.3	TCN-Autoencoder(TCN-AE)	29
第 5 章	Autoencoder 構造を用いた異常検知・故障予測手法	30
5.1	手法 1 : Autoencoder と GMM による異常検知・故障予測手法	30
5.2	手法 2 : Autoencoder による end-to-end 異常検知・故障予測手法	32
第 6 章	穴あけパンチ加工におけるパンチ刃の摩耗状態の検知	34
6.1	使用するデータ	34
6.1.1	データセット 1	34
6.1.2	データセット 2	35
6.2	実験条件	36
6.2.1	AE モデル	37
6.2.2	評価指標: Area Under Curve	39
6.3	データセット 1 における実験結果	40
6.4	データセット 2 における実験結果	46
6.5	考察	52
第 7 章	実機械振動データによる故障予測	53
7.1	使用するデータおよび理想の異常度推移	53
7.1.1	データセット A	53
7.1.2	データセット B	56
7.2	実験 1: 手法 1 による故障予測の可能性および頑健性の検証	56
7.2.1	実験条件	56
7.2.2	比較手法	58
7.2.3	実験結果	58
7.2.4	考察	61
7.3	実験 2: TCN-Autoencoder に用いるパラメータの調査	64
7.3.1	実験条件	64
7.3.2	実験結果・考察	66
7.4	実験 3: 異常度算出手法間における特性の調査と精度比較	69

7.4.1	実験条件	69
7.4.2	実験結果	70
7.4.3	考察	72
7.5	実験 4: 長期間データによる故障予測実験	74
7.5.1	実験条件	74
7.5.2	実験結果	75
7.5.3	考察	77
第 8 章	結論	78
8.1	まとめ	78
8.2	今後の課題	79
謝辞		80
参考文献		82
発表文献		86

表目次

6.1	穴あけパンチ加工機に設置したセンサデータの取得条件	35
6.2	実験条件	36
6.3	AE の設定条件	37
6.4	TCN-AE の構造 (データセット 1)	37
6.5	FFNN-AE の構造 (データセット 1)	38
6.6	TCN-AE の構造 (データセット 2)	38
6.7	FFNN-AE の構造 (データセット 2)	38
6.8	6 種類の異常度算出手法による 3 種類の AUC と異常度算出時間 (データセット 1)	45
6.9	6 種類の異常度算出手法による 3 種類の AUC と異常度算出時間 (データセット 2)	51
7.1	データセット A 取得条件	55
7.2	データセット B 取得条件	56
7.3	実験 1 における, 提案手法に対する実験条件	57
7.4	実験に用いた機材およびフレームワーク	57
7.5	実験 1 における LSTM-AE の構造	57
7.6	Hotelling's T-square 法および PCA における設定条件	58
7.7	実験 2 における AE の設定条件	65
7.8	$F=(8,16)$ における TCN-AE の構造	65
7.9	積層数・フィルタ数毎のパラメータ数	67
7.10	AE の設定条件	68
7.11	入力データ長毎の異常度算出速度	68
7.12	各区間における異常度の平均, および故障前後の異常度の比. ただし, 故障前後の異常度の比 = (区間 1-区間 3)/(区間 2-区間 3)	71
7.13	モデル毎, および積層数・ノード数・ブロック数・フィルタ数毎のパラメータ数および 1 データあたりの異常度算出速度	71
7.14	3 種類の AE ベース異常度算出手法の比較	73
7.15	実験 4 実験条件	74

目次

2.1	サーベイ論文「Deep Learning for Anomaly Detection: A Survey[7]」における DAD 手法の分類. 使用するネットワーク構造および学習手法による 4 種類の大分類と, モデル構造による 6 種類の小分類により構成されている	8
2.2	サーベイ論文「Deep Learning for Anomaly Detection: A Review[12]」における DAD 手法の分類. 深層学習手法の使用用途による 3 種類の大分類と, モデリング手法による 11 種類の小分類により構成されている	9
2.3	DCASE2020 機械音異常検知チャレンジ	11
3.1	振幅スペクトル抽出の流れ	14
3.2	ハミング窓	14
3.3	ハン窓 (ハニング窓)	15
3.4	PCA	16
4.1	自由度 $k = 1, 2, 4, 8$ におけるカイ二乗分布の確率密度関数	18
4.2	k 近傍法の例. この場合, 未知のサンプル (図中緑色の点) は, $k = 3$ のとき Class 1 に, $k = 5$ のときに Class 2 に分類される.	20
4.3	One-Class SVM の適用例. データに対しカーネル ϕ を適用し, 入力データの空間 (図左) から特徴空間 (図右) に写像することで, 正常・異常の識別境界を設定する.	20
4.4	多層パーセプトロンの構造例	21
4.5	RNN の構造例	23
4.6	Gers ら [37] の LSTM ブロックの構造	24
4.7	Encoder-Decoder TCN (ED-TCN)	26
4.8	Autoencoder (AE) の構造.	28
4.9	LSTM-AE の構造. 図 4.8 における Encoder 層および Decoder 層を LSTM ブロックにより構成した構造を持つ.	29
5.1	提案手法 1 の構造.LSTM-AE と GMM の 2 種類により構成されている. LSTM-AE を用いて時系列データの振幅スペクトルより特徴を抽出し, 抽出した特徴から GMM により異常度を算出する.	30

5.2	提案手法の構造.Autoencoder のみを用いた end-to-end 構造である.	32
6.1	ROC 曲線	39
6.2	Hotelling's T-square 法による異常度算出結果 (データセット 1)	40
6.3	GMM による異常度算出結果 (データセット 1)	41
6.4	K-NN による異常度算出結果 (データセット 1)	41
6.5	OC-SVM による異常度算出結果 (データセット 1)	42
6.6	FFNN-AE による異常度算出結果 (データセット 1)	42
6.7	TCN-AE による異常度算出結果 (データセット 1)	43
6.8	Hotelling's T-square 法による異常度の ROC 曲線 (データセット 1)	43
6.9	GMM による異常度の ROC 曲線 (データセット 1)	43
6.10	K-NN による異常度の ROC 曲線 (データセット 1)	44
6.11	OC-SVM による異常度の ROC 曲線 (データセット 1)	44
6.12	FFNN-AE による異常度の ROC 曲線 (データセット 1)	44
6.13	TCN-AE による異常度の ROC 曲線 (データセット 1)	45
6.14	Hotelling's T-square 法による異常度算出結果 (データセット 2)	46
6.15	GMM による異常度算出結果 (データセット 2)	47
6.16	K-NN による異常度算出結果 (データセット 2)	47
6.17	OC-SVM による異常度算出結果 (データセット 2)	48
6.18	FFNN-AE による異常度算出結果 (データセット 2)	48
6.19	TCN-AE による異常度算出結果 (データセット 2)	49
6.20	Hotelling's T-square 法による異常度の ROC 曲線 (データセット 2)	49
6.21	GMM による異常度の ROC 曲線 (データセット 2)	49
6.22	K-NN による異常度の ROC 曲線 (データセット 2)	50
6.23	OC-SVM による異常度の ROC 曲線 (データセット 2)	50
6.24	FFNN-AE による異常度の ROC 曲線 (データセット 2)	50
6.25	TCN-AE による異常度の ROC 曲線 (データセット 2)	51
7.1	チェーンコンベアのモータ	54
7.2	モータ部に設置した加速度センサ. センサはモータの左下, ギアの右側に位置している	54
7.3	データセット A における, データ取得期間中の発生イベントと理想的な異常度の推移	55
7.4	提案手法 1 による異常度の散布図と移動平均グラフ. 異常度の数値の 90% 点は 0.7225, 95% 点は 0.9351 となり, 異常度の最大値は 4.0766 であった. 本グラフ, および比較手法のグラフより, 提案手法 1, および 3 種類の比較手法すべてにおいて, 故障を前もって予測することが出来ていることが示された.	59

7.5	比較手法 1 (PCA+Hotelling's T-square 法) による異常度の散布図と移動平均グラフ. 異常度の数値の 90% 点は 0.0526, 95% 点は 0.0954 となり, 異常度の最大値は 0.4388 であった.	60
7.6	比較手法 2 (PCA+GMM) による異常度の散布図と移動平均グラフ. 異常度の数値の 90% 点は 0.0001, 95% 点は 0.0002 となり, 異常度の最大値は 0.1731 であった.	60
7.7	比較手法 3 (LSTM-AE+Hotelling's T-square 法) による異常度の散布図と移動平均グラフ. 異常度の数値の 90% 点は 0.7858, 95% 点は 0.9836 となり, 異常度の最大値は 4.3055 であった.	61
7.8	故障 4 週間前-故障 2 週間前 (Period 1) と, 故障 2 週間前-故障直前 (Period 2) の 2 区間における異常度の中央値, および 2 区間の中央値の比 (Period 2/Period 1). 表より, PCA による特徴抽出を行った場合, LSTM-AE の場合と比較して故障直前の異常度の上昇幅が大きいことが分かった.	61
7.9	入力データ長 100 ms, 積層数 1, フィルタ数 (8) における異常度の推移	66
7.10	入力データ長 800 ms, 積層数 1, フィルタ数 (8) における異常度の推移	66
7.11	入力データ長 800 ms, 積層数 1, フィルタ数 (16) における異常度の推移	67
7.12	入力データ長 800 ms, 積層数 4, フィルタ数 (8,8,8,8) における異常度の推移	67
7.13	LSTM-AE, 積層数 2, ブロック数 (8,8) における異常度の推移	70
7.14	FFNN-AE, 積層数 2, ノード数 (256,128) における異常度の推移	70
7.15	データセット B における異常度算出結果	75
7.16	データセット B における異常度算出結果の移動平均	75
7.17	データセット B における異常度算出結果の移動分散	76
7.18	データセット B における異常度算出結果 (一部区間のみ)	76
7.19	データセット B における異常度算出結果の移動平均 (一部区間のみ)	76
7.20	データセット B における異常度算出結果の移動分散 (一部区間のみ)	77
7.21	データセット B における異常度算出結果の ROC 曲線および AUC. 左から, 異常度, 移動平均, 移動分散の ROC 曲線および AUC をそれぞれ示す.	77

第 1 章

序論

1.1 研究背景

ここ数年でコンピュータの計算能力が格段に上昇し、AI・IoT 分野の研究開発は急速に発展した。一般社会においても、AI・IoT 技術を取り入れたシステム・機器が広く普及し始めている。特に、「第 3 次 AI ブーム」とも呼ばれる昨今の人工知能技術の研究開発ムーブメントは、2012 年に行われた物体認識の精度を競うコンペティション「ILSVRC2012」 [1] において、Krizhevsky らが使用した深層学習モデル「AlexNet」 [2] が従来の機械学習手法の精度を大きく上回って優勝したことに端を発している。現在に至るまで、画像処理をはじめとしたさまざまなタスクにおいて深層学習技術を利用した手法が数多く提案され、古典的機械学習を上回る性能を発揮し、様々な実世界タスクにも応用されている。このブームは製造業をはじめとする産業分野においても同様に発生している。コンピュータを用いて課題解決を行うデジタルトランスフォーメーション (DX) という言葉およびその概念が産業分野において浸透し、工場設備や機械にセンサなどのデータ収集機器を取り付け、機械学習を応用した技術によりデータの解析を行うことで、課題解決を行うための手法およびシステムの研究開発は盛んに行われ始めている。

製造業における最重要課題の一つとして、「異常検知」というタスクが挙げられる。工場内で稀に発生する不良品や異物混入や、経年劣化などに起因する機械設備の故障など、様々な「異常」が不定期に発生する。これらの異常の発生は企業の生産活動に対する信頼の低下や事故の発生にも繋がる重大な要素であることから、異常検知に関連する研究には高い関心が集まっている。この関心の高さは、2020 年に開催された機械の駆動音から異常検知を行う国際コンペティション [3] に、国内外から 40 チームが参加し、100 件を超える異常検知モデルが提案されたことからもうかがえる。

従来より、実世界における工場設備や機械に対する異常検知の研究は数多く行われている。従来提案され、実用化に至っている手法およびシステムは、ルールベース、統計的手法ベース、または単純な機械学習ベースに基づくモデルがほとんどを占めている。例として、加工異常の検出 [4] には閾値処理に基づくルールベース異常検知手法が、機器の異常診断 [5] には主成分分析と近傍法に基づく機械学習ベース異常検知が提案手法として採用されている。一方で、産業分野においても深層学習を用いた異常検知を行う研究が増加傾向にある [6]。深層学習ベースによる異常検知は、モデルの構造によっては

専門家の知識や経験を基に行う特徴抽出のプロセスを自動化でき、ラベルの付与を行う手間を省略できる [7]。これにより、データ入力から分類・予測結果の出力まで一連のプロセスを深層学習により行う end-to-end なシステムの構築が可能である。このほか、より多くの種類のデータの収集が可能となり、ビッグデータによる異常検知に対するニーズの高まりや、データのもつ性質により、機械学習ベース異常検知では限界のある課題を、深層学習ベース異常検知が解決することへの期待などから、深層学習ベース異常検知の研究が増加していることがうかがえる。

1.2 研究目的

本研究では、製造業の抱える課題の一つである、工場内における機械および工場設備の機械設備の劣化により発生する「故障」の事前予測を、定期的を取得した振動データやセンサデータ、マイクなどによる音声などの波形信号の情報を持つ時系列データに対して深層学習技術を用いて行うことを目的として取り組んできた [8][9]。本研究における異常予測、もしくは故障予測モデルは、定期的を取得する時系列データから、正常時からどれだけ離れているかを示す指標（以下、異常度 (Anomaly score) と記す）を算出する。異常度は、その数値が高ければ高いほど異常に近い状態であるとする指標である [10]。異常度が一定期間高い数値を示した場合に、まもなく故障することを示すアラートを出す、という故障予測アルゴリズムの提案が最終的な目的となる。

本論文では、機械や工場設備にセンサやマイクなどを設置し、得られたセンサや音声などの時系列データの時間ごとの推移から、深層学習技術を用いて構築した異常度算出モデルにより、機械や工具、設備の異常および故障を予測する手法を 2 種類構築する。1 種類目は、特徴抽出を深層学習モデルで行い、異常度算出を機械学習モデルで行う 2 段階構成の異常度算出モデルである。Autoencoder による特徴抽出を行うことで、従来用いられていた次元圧縮手法よりも効果的に異常を発見できる特徴を抽出することが期待できる。2 種類目は、深層学習モデルを用いて、センサや音声などの時系列データを生データのまま入力し、異常度を算出する異常度算出モデルである。これは、データの前処理や特徴抽出などのプロセスを省略した end-to-end 方式による異常度算出モデルである。

本論文では、構築した手法による異常検知および故障予測の性能を検証するため、2 種類のデータセットにより異常検知および故障予測を実施した実験について、その実験結果および考察を記述する。

なお、本研究において扱う故障予測タスクでは、異常度算出精度に加え、データに対するモデルの頑健性も重視している。「より頑健な故障予測モデル」について、本論文では故障発生前の異常度が正常時と比較してより明確に差が見られるかどうかに加え、設備の劣化の程度に伴い徐々に異常度が推移しているかどうか、異常度の突発的な変化の発生をどれだけ抑えられるか、また故障が発生する一定期間前には故障がまもなく発生すると判定できるような異常度の推移が見られるかどうか、そして正常である期間内についても、異常時よりも十分低いものの徐々に異常度が上昇するような異常度を算出できているかどうか、という条件により該当するモデルであること、と定義する。

1.3 本論文の構成

本論文は、次のように構成されている。

第 2 章では、はじめに異常検知タスクにおける定義や課題、および深層学習ベースによる異常検知手法について、その定義や分類方法、各分野における課題などについて紹介する。その後、本論文で取り扱う音声やセンサデータなどの波形信号情報を持つ時系列データを用いた異常検知について、その特徴や課題について説明する。最後に、音声データによる異常検知を行うコンペティションの結果をもとに、近年の波形信号データを用いた異常検知タスクに対する研究動向を紹介する。

第 3 章では、加速度センサやマイクなどより得られる波形信号データにより構成される時系列データから抽出する、2 種類のデータの抽出の流れについて解説する。その後、従来より入力データに対する次元削減手法として利用され、本研究でも利用する主成分分析について説明する。

第 4 章でははじめに、本研究において使用する、従来より異常検知タスクにおいて用いられてきた機械学習ベース異常検知手法を説明する。その後、本論文において提案する異常検知・故障予測モデルにおいて使用する、深層学習ベース異常検知・特徴抽出手法を説明する。

第 5 章では、構築する 2 種類の異常検知・故障予測手法において、そのモデル構造と異常度の算出手法について詳説する。1 種類目は、深層学習手法である Autoencoder[11] による特徴抽出と、機械学習手法である Gaussian Mixture Model[10] による異常度算出の 2 段階による手法である。Autoencoder による特徴抽出を行うことで、従来用いられていた次元圧縮手法よりも効果的に異常を発見できる特徴を抽出することが期待できる。2 種類目は、Autoencoder のみを異常検知および故障予測に用いる手法である。本手法は、end-to-end 方式によりデータを学習し、異常度算出には予測と入力との再構成誤差を用いる。本手法に用いる深層学習モデルには、Feed-Forward Neural Network Autoencoder (FFNN-AE)、Long Short-Term Memory Autoencoder (LSTM-AE) および Temporal Convolutional Network Autoencoder (TCN-AE) の 3 種類の AE を採用した。

第 6 章では、実際の加工機に設置したセンサデータより収集したデータセットを用いて、穴あけパンチ加工におけるパンチ工具の摩耗状態の検知を行う実験を実施した。従来の機械学習ベースによる異常度算出結果と比較することで、異常検知および故障予測の性能を評価し、深層学習ベース異常検知モデルの有効性を検証した結果を詳説する。

第 7 章では、実際のチェーンコンベアより得られた振動データによるデータセットを用いて故障予測を行う実験を実施した。本データセットに対する最も安定性を保証できるパラメータを調査するため、TCN-AE を用いて層数やパラメータなどを変化させ、その精度および学習速度について考察した。続いて、3 種類の AE を用いて異常度算出を行い、構造による異常度の推移に対する特徴や、算出速度とパラメータ数などの特性の違いを、各モデルのもつ一般的な特徴と照らし合わせて比較し、3 種類の AE ベースのモデルによる異常度算出結果においてどのような特徴が現れるかを調査し、振動データの生波形信号を入力とした end-to-end 異常度算出手法による故障予測の有効性を考察した。

最後に第 8 章で、本論文の結論を述べ、本研究による異常検知および故障予測に対する今後の展望について述べる。

第 2 章

異常検知と深層学習

本章では、異常検知のうち、特に深層学習ベースの手法について、その定義や分類方法、各分野における課題などについて紹介する。まず、検知および予測を行う「異常」の定義と分類、および異常検知における課題について説明する。次に、異常検知タスクに使用される深層学習ベース異常検知 (Deep Anomaly Detection; DAD) 手法について、その定義の説明と手法の分類を行う。その後、同様に本研究に関連する時系列データを用いた異常検知の特徴と課題について述べる。また、本章の最後では、2020 年に行われた、機械の駆動音から異常検知を行う国際的なチャレンジの結果をもとに、最新の異常検知手法のトレンドを紹介する。

2.1 異常検知とは

異常検知の研究は、幅広い分野で異常を検知するタスクに対する高い需要があることから、数十年にわたって国内外で活発に行われている。現在、機械学習や統計学などをベースとした数多くの手法が研究・提案されている。また、その一部は既にシステムとして応用され、実用化されている例も存在する [7]。

2.1.1 異常および異常検知の分類

これらの研究において対象となる、実世界において発生する「異常」は、既に観測されているデータ群内の異常、すなわち通常のデータ群から離れた場所・地点にあるデータに起因する異常 (異常値, Anomaly) と、既知のデータ中に出現していない、新規または未観測のパターンに起因する異常 (新規値, Novelty) の 2 種類が考えられる。一方で、異常値か新規値かに関わらず、異常検知分野における研究は、次の 3 種類に大分できる [7]。

- 点 (Point) 異常
- 文脈的 (Contextual) 異常
- グループ (Group) 異常

点異常とは、他の大多数のデータと異なる異常を示す突発的な異常のことを指す。外れ値検知など、大多数の異常検知の研究は、この点異常を検知することを目的としている。文脈的異常とは、特定の条件が揃っている場合に発生が確認できる異常のことを指す。例えば映像データについて、個々の画像として観測した場合には正常であっても、動画データとして観測した場合に異常が発見できるといったような異常が文脈的異常に分類される。グループ異常とは、個々のデータで見た場合は正常であっても、グループとして見た場合に発見される異常を指す。例えば時系列データについて、1日単位で計測した場合には全てのサンプルが正常の範囲内であると判定されるのに対し、1か月継続して測定したところ先の1日のデータが異常サンプルと判定されるといったような異常がグループ異常に分類される。文脈的異常やグループ異常が発生しているデータに対し、点異常に適応した検知手法を用いると、その条件やグループをモデルが把握できず、結果として異常を発見できないことが考えられる [12]。このため、文脈的異常やグループ異常は、点異常と比較して難度の高い異常であるといえる。

2.1.2 異常検知における課題

異常検知タスクは、普段発生しない稀なイベント (=異常) を検出するタスクであり、分類・回帰問題と異なる特有の課題を抱えている [12]。この課題として、以下の4点があげられる。

- データの不明性
- 異常の不規則性
- 異常の希少性・不均衡性
- 難度の高い異常の種類の判定

「異常」は、実際に発生するまで分からない未知の現象である場合がほとんどである、という「不明性」が存在する。特に新規値を検出しようとした場合、この不明性に対してどう対処するかが大きな課題であるといえる。また、異常サンプルを発見できた場合でも、このサンプルと別の異常サンプルが同じ特性を持つとは限らないという「不規則性」を持っている可能性が考えられる。正常と異常を分類することを考えた場合、分類するクラスの設定基準によっては、新規異常サンプルを正確に分類できないという課題も存在する。また、異常の発生は稀である場合が多く、この場合は大量に異常データを収集することは困難であるという「希少性」をもつ。この希少性により、正常クラスと異常クラスのデータ数に大きな差が生じるという「不均衡性」が生じる。希少性および不均衡性が発生するデータセットについて、大規模なラベルデータやそれに基づいて事前学習を行ったクラス分類手法を適用することが困難である。そして、先にも述べた通り、点異常の検知が可能であっても、文脈的異常やグループ異常が発生している場合は検知が困難である。このように、難度の高い異常を発見することも重要な課題である。

さらに、実世界データに対して異常検知を行う手法およびシステムを提案し、実用化を考える場合、上記の課題に加えて以下の3点のような課題も発生する。

- 再現率問題

- データノイズ
- 異常判定理由の説明

正常データと異常データの不均衡性が生じているデータセットを用いた異常検知を行った場合、正常データを異常データと誤判定する偽陽性率 (False Positive Rate) が高くなり、再現率が低下する可能性がある。一方で、実運用を考えた場合、異常データを正常データと誤判定する偽陰性率 (False Positive Rate) が少しでも存在するモデルは、すなわち異常データを見逃してしまう致命的な側面を抱えてしまうため、再現率が多少高くなるとしても、偽陰性率を 0 に抑える必要がある。このため、システムとして応用する際には、判定後に人手で正常・異常の判定を行うなど、工夫が必要である。また、実世界において計測および収集されたデータには、ノイズが重畳している場合がある。もしくは、正常・異常を人手で付与したラベルにミスが発生することも考えられる。こういったデータやラベルのノイズが存在する場合、異常検知性能に大きく影響する。こういったノイズは、深層学習ベースによる異常検知手法であっても性能低下に寄与する場合があるため、重要な課題であるといえる。さらに、特に産業分野においては、異常を異常と判定した理由を説明できるモデルは、異常検知精度と同等に重視される。このため、モデルの有効性と解釈可能性を両立させたモデルの構築が求められる場合がある。

このように、異常検知タスクを解決するための課題は、様々な側面から多様に存在している。これらの課題は、次節で説明する深層学習ベース異常検知によって解決されることが期待されている。

2.2 深層学習ベース異常検知手法

近年、深層学習技術を異常検知に用いる、深層学習ベース異常検知 (DAD) の研究が多くなされ、従来手法と比較して大幅に性能を向上した結果を出している。

2.1 節でも述べた通り、数十年にわたって様々な分野において機械学習ベースおよび統計学ベースの異常検知手法が提案されてきたが、古典的な機械学習・統計的アルゴリズムによる異常検知手法では解決が困難な課題が数多く存在する。このほかにも、大量・多変量なビッグデータによる異常検知が時間的・リソース的に困難であること、専門家の知識ベースによる入力データからの特徴抽出が要求されることなど、古典的な機械学習アルゴリズムによる異常検知における課題は数多く存在している。深層学習の場合、ビッグデータから階層的に判別可能な特徴を自動的に学習することが可能であり、同時に手動での特徴抽出のプロセスを除外できるため、生データを入力し、結果を出力する end-to-end 分類や回帰などが可能になる。これをはじめ、先に提示した課題に対する解決を目的として提案・構築された手法やモデルも存在している。

2.2.1 深層学習ベース異常検知手法の分類

DAD 分野における研究において提案されている、異常検知手法の分類方法は、2 種類のサーベイ論文 ([7], [12]) によって異なる方式が提案されている。以下、本章では「Deep Learning for Anomaly Detection: A Survey[7]」を「Survey」、 「Deep Learning for Anomaly Detection: A Review[12]」を「Review」と記載する。

「Survey」における DAD 研究の分類を図 2.1 に示す。本サーベイ論文においては、はじめに対象となるネットワーク構造、および適用される学習手法によって次の 4 種類に大分される。

- 教師なし学習 (Unsupervised)
- 半教師あり学習 (Semi-supervised)
- ハイブリッド構造 (Hybrid)
- One-class Neural Networks

ただし、本分類における「ハイブリッド構造」とは、深層学習を入力データからの特徴抽出のみに用い、抽出した特徴を用いて古典的な機械学習・統計的アルゴリズムによる異常検知を行うという、2 段階の構造をもつモデルを指す [13][14]。また、One-class Neural Networks (OC-NN) とは、正常データと異常データとの距離を分離する超平面や超球に基づく識別境界面を算出し、この境界をもとに異常検知を行うモデルである [15]。なお、本分類に「教師あり学習」が含まれていない。この理由として、教師あり学習によるモデルでは、先に述べた異常検知における課題である「異常の不規則性」および「異常の希少性・不均衡性」の影響を大きく受けることに加え、「ラベルの不完全性」が存在する場合、教師あり学習モデルが利用できないことが挙げられる。教師あり学習は全ての学習データにラベルが付与されている必要があるが、異常検知タスクにおいて対象となる実世界データでは全てのデータにラベルが付与することは難しく、仮にデータラベルの入手できたとしても、そのラベルが全て正確に付与されているかという、ラベルに対する信頼性に欠ける場合がある。この場合、教師あり学習によるモデルでは性能の低下、もしくは性能に対する信頼性の低下などが懸念されるため、教師なし学習や半教師あり学習などのモデルと比べて利用が難しいという課題がある。

「教師なし学習」モデルは、モデル構造から更に「Autoencoder」「生成的・敵対的モデル (Generative/Adversarial)」「Variational」「行列分解 (Matrix factorization)」の 4 種類に分類される。教師なし学習による手法は、Autoencoder[11] や変分 Autoencoder (Variational Autoencoder; VAE)[16], Generative Adversarial Networks (GAN)[17] のような encoder-decoder 型構造をもつモデルを用いた研究が主流である [18][19]。学習時のラベルデータが不要であり、ラベル付与に要するコストを抑えられるため、異常を発見するための費用対効果が高い。このため、教師なし学習による DAD は基礎研究と産業への応用の双方において重要な研究分野である。また、行列分解による異常検知は、深層学習ベースの非負値行列因子分解 (Non-negative Matrix Factorization; NMF)[20] などの行列分解を組み込んだネットワーク構造を採用した手法である [21]。

また「半教師あり学習」に分類される研究としては、深層強化学習 (Deep Reinforcement Learning; DRL) をベースに用いたものが多い [22][23]。DRL ベース異常検知では、異常の概念に関する過程を考慮せず、蓄積された報酬のみを用いて知識を高めることで、新たな異常を検知することが期待される。

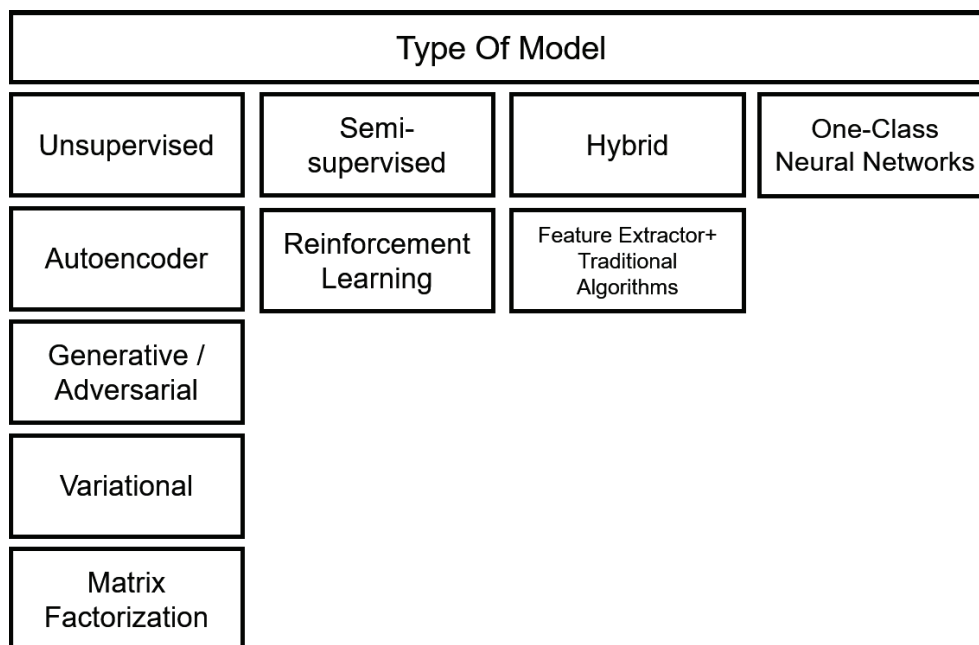


図 2.1 サーベイ論文「Deep Learning for Anomaly Detection: A Survey[7]」における DAD 手法の分類。使用するネットワーク構造および学習手法による 4 種類の大分類と、モデル構造による 6 種類の小分類により構成されている

つづいて、「Review」における DAD 手法の分類を図 2.2 に示す。本サーベイ論文においては、はじめに異常検知タスクに対する深層学習手法の使用用途によって次の 3 種類に大分される。

- 深層学習による特徴抽出 (Deep Learning for feature extraction)
- 正常データの特徴表現学習 (Learning feature representation of normality)
- End-to-end 異常度算出 (End-to-end anomaly score learning)

「深層学習による特徴抽出」に分類される研究では、深層学習は異常検知に直接関与させず、高次元・非線形に分離可能なデータから、低次元の特徴表現を抽出することに用いられる。特徴抽出に深層学習モデルを用いることで、主成分分析 (Principal Component Analysis; PCA)[24] や線形判別分析 (Linear Discriminant Analysis; LDA)[25] などの従来手法よりも意味情報や非線形な特徴関係を抽出する上で優れた性能を発揮することが期待できる。

「正常データの特徴表現学習」はさらに「一般的な正常データの特徴学習 (Generic normality feature learning)」と「異常度に依存した特徴学習 (Anomaly feature-dependent feature learning)」の 2 種類に分類できる。前者は、正常サンプルのみ、もしくはほぼ正常サンプルで構成された学習データを用いて Autoencoder や GAN などなどの深層学習モデルを学習し、再構成誤差などを用いて異常度を算出する手法である。後者は K 近傍法 (K-Nearest Neighborhood; KNN)[26] や One Class SVM (OC-SVM)[27] などの既存の異常検知手法に入力することを想定して入力データの特徴表現を学習する手法である。「Survey」における「ハイブリッド構造」は、「Review」における「深層学習による特

「特徴抽出」と「異常度に依存した特徴学習」の 2 つが該当する。

「End-to-end 異常度算出」は、既存の異常度算出手法とは異なり、学習データから異常度を直接学習する深層学習モデルを構築する研究である。ランク学習ベースによる手法 [28] や One-class 分類ベースによる手法 [29] など、特に画像データによる異常検知において注目されている研究分野である。

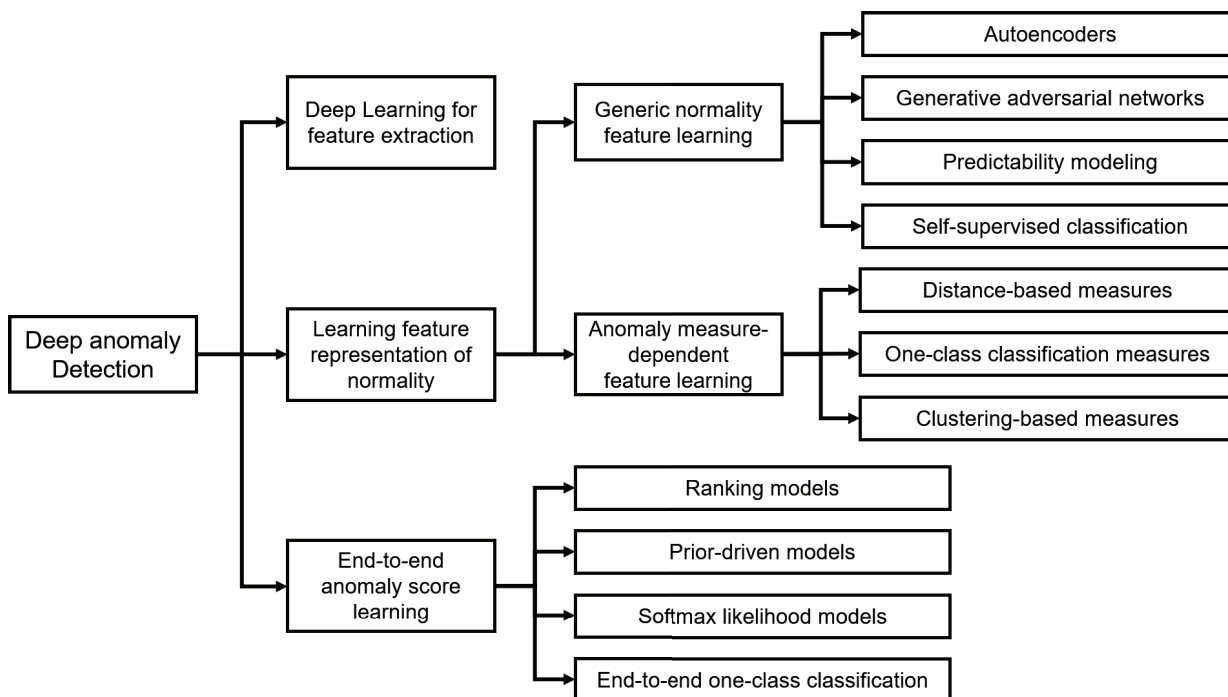


図 2.2 サーベイ論文「Deep Learning for Anomaly Detection: A Review[12]」における DAD 手法の分類。深層学習手法の使用用途による 3 種類の大分類と、モデリング手法による 11 種類の小分類により構成されている

2.2.2 深層学習ベース異常検知における課題

先にも述べた通り、DAD は従来の異常検知手法では解決が困難であった課題を解決することが期待されている。一方で、DAD の分野自体が現在研究途上であるため、すべての異常検知問題に応用できるとは限らない。ただし、DAD 技術を応用した場合に従来の機械学習ベース異常検知よりも性能が向上しているという研究は存在していることから、DAD を利用することによる異常検知タスクへの貢献度は高く、引き続き DAD に対するニーズは多く存在することが示唆される。DAD 技術を実世界データに対して用いる際に、どの構造や手法を用いるかの見極めを人手で行う必要があることは、DAD を利用する上での課題と言える。例えば、正常と異常のデータ数がほぼ同数であれば、教師あり学習による分類タスクとして扱うことで最も性能を発揮することが知られている。また、実世界データにおける異常検知を行う場合、使用するモデルの計算複雑性は重要な側面であるが、教師あり学習が最も計算がシンプルとなることが分かっている。一方で、DAD 技術を用いる場合においても、課題は残って

いる。例として、適用するデータセットによっては正常・異常挙動は正確に定義できない、というものがある。この定義不足は、用いるアルゴリズムが深層学習か否かにかかわらず、異常検知タスクとしての解決すべき課題の一つといえる。さらに、先に述べた「再現率問題」や「データノイズ」といった課題については、DAD を用いる上で大きな課題であるといえる。データノイズについては、半教師あり学習および弱教師あり学習 (Weakly supervised) による深層学習モデルの場合、様々な要因によって発生するラベルの付与ミス [30] がデータ中があると精度が低下することが分かっている。また、深層学習モデルによって出力された異常検知結果は、異常を異常と判断した根拠を説明できないブラックボックス状態に陥り、モデルの信頼性に欠いてしまうという課題がある。この課題を解決するため、説明可能な AI (Explainable AI; XAI) に関する研究が進んでおり、将来的には異常検知分野においてもモデルの有効性と解釈可能性を両立させたモデルの構築が期待できる。

2.3 時系列データにおける異常検知

異常検知モデルへの入力データとして使用される実世界データは、画像やログデータ、テキストデータなど様々な種類が存在する。本節ではこのうち、本論文にて使用するデータである、音声やセンサデータなど、時間軸に沿ってデータが変化する、時系列データにおける異常検知について紹介する。時系列データは、温度センサなど、一変量のみで推移する場合（一変量時系列）と、3 軸加速度センサなど、一度に多変量のデータが相互に関係を持ちながら推移する場合（多変量時系列）に分類できる。一方で、時系列データにおいて発生する異常は、突発的に発生するような点異常、連続的に異常が発生するような文脈的異常、そして点異常が複数回発生し、その突発以上が何らかの法則に従って発生していると推測できるようなグループ異常の 3 つに分類できる。

2.4 時系列データを用いた異常検知における課題

時系列データを用いた異常検知における課題としては、次の 4 点が挙げられる。

- 異常パターンが定義することが難しい場合がある。
- 入力データ内のノイズが性能に大幅に影響することがある。
- 時系列データ長が増大するにつれ、計算の複雑性も増す。
- 時系列データは非定常・非線形・動的に変化するため、リアルタイムに異常検知する必要がある。

以上の 4 点とも、異常検知を行う際に大きな障害となり得る課題である。観測される異常の時間的な法則性が無く、全て異なる挙動をとる場合など、データに基づいて異常パターンが定義することが難しい場合が考えられる。このとき、どこを異常とし、どこを正常とするかは経験的知見や収集対象の状態の遷移などに基づいて考察する必要がある。入力データ内にノイズが発生している場合、画像データと同様にノイズの削減および除去が難しく、かつ実際の異常に起因しないノイズが異常度算出に大きく寄与した場合、意図したモデルが作成できなくなる。時系列データ長が増大すると、計算にかかる時間も増大し、かつデータの複雑性も増すため、計算の複雑性や難易度も増大する。最期に、デー

タが時刻とともに非定常・非線形・動的に変化するため、画像のように静的な側面だけでなく、動的に正常・異常を判断するシステムが要求される。以上のような課題が発生することもある。時系列データにおける異常検知タスクは難易度が高く、多くの研究者が現在検知精度、もしくは予測精度の向上のために調査および研究を進めている途上の分野である。

2.4.1 時系列データを用いた異常検知分野における研究動向

本項では、2020 年に行われた、機械の駆動音から異常検知を行う国際コンペティション「DCASE2020 Task 2: Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring」[3] の結果を基に、本論文を記述する時点での異常検知関連の研究傾向を紹介する。本コンペティションに参加したチームは 40 チーム。合計 117 件の異常検知モデルがエントリーした。

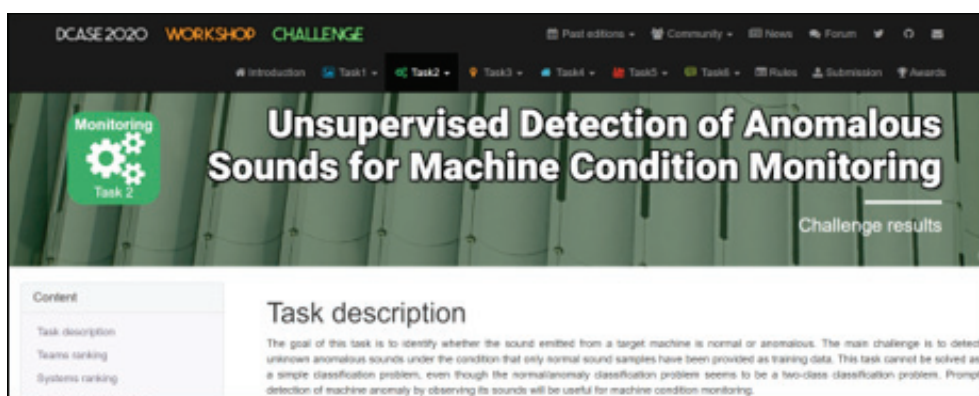


図 2.3 DCASE2020 機械音異常検知チャレンジ

はじめに、異常検知モデルに関する傾向について紹介する。この 117 件のうち、深層学習構造であるニューラルネットワークを使用していないものは 16 件であった。すなわち、9 割近い異常検知モデルが深層学習ベースであることが分かる。また、上位 10% の異常検知モデルは全て深層学習ベースの異常検知モデルであることから、検知精度の観点からも深層学習が従来の機械学習ベースによる異常検知モデルを上回る結果となる傾向にあることが伺える。検知精度上位のモデルは、ほぼすべてが畳み込みニューラルネットワーク (CNN) 系の構造により構成されていた。音声による異常検知タスクであるため、時系列データの分類を得意とする再帰型ニューラルネットワーク (RNN) や Transformer[31] 系のネットワーク構造を用いる参加チームもあったものの、CNN ベース異常検知と比較すると精度が低い結果となった。また、検知精度 12 件のうち、複数の AE 構造によるアンサンブル学習を行ったモデルを採用したものは 7 件存在し、また検知精度上位 4 件はすべてアンサンブル学習を採用していた。このことから、時系列データによる異常検知タスクは、単独モデルによる検知性能の改善が難しい分野であることが考えられる。一方で、実用上はアンサンブル学習によるモデルが有用であることも示唆される。

続いて、異常検知モデルへの入力データに関する傾向について紹介する。本タスクにおける入力データは音声データであるが、この音声を生のまま使用している参加チームは、117 件中わずかに 5 件で

あった。ほとんどのチームは log-mel 特徴や MFCC など、音声から特徴を抽出してモデルに入力していた。さらに言えば、先述した 5 件についても、log-mel 特徴も併用している。このことから、時系列データ、特に音声のような波形信号情報を持つ時系列データに関しては、依然として特徴抽出を行った上でモデルに入力する傾向にあることが伺える。

第3章

時系列データにおける特徴抽出技術

本研究で扱う、センサやマイクなどから収集される音信号やセンサデータなど、波形信号情報を持つ時系列データは、単位時刻毎に複数次元の特徴を持つ周波数方向の情報だけでなく、その周波数方向の情報が時間とともに変化するため、時間方向の情報にも注目する必要がある。周波数方向・時間方向の双方を加味した時系列データの特徴抽出を行う手法は複数提案されてきており、現在でも深層学習モデルへの入力データに対する前処理として用いられることがある。

本章では、本研究において深層学習モデルへの入力として利用する生データの切り出し方法、および振幅スペクトルの抽出手法について解説する。その後、入力データに対する次元削減手法として利用する主成分分析 (Principal Component Analysis; PCA) について説明する。

3.1 特徴抽出手法

本研究では、時系列データを用いた異常検知および故障予測モデルへの入力データとして、振幅スペクトルと生データの2種類のデータを使用する。入力データの抽出の流れを図3.1に示す。本節では、はじめに振幅スペクトルの抽出手法を説明し、最後に本研究における生データの抽出手法、および取り扱いについて述べる。

入力された音信号やセンサデータを、フレームと呼ばれる短いデータに分割し、高速フーリエ変換 (Fast Fourier Transform; FFT) [32] を行うことで算出した、振幅スペクトルを異常検知および故障予測モデルへの入力データに使用する。時系列データから抽出した振幅スペクトルをモデルに連続的に入力することで、データの持つ時間次元と周波数次元の双方の情報をモデルに入力することが可能である。

本研究において使用する「振幅スペクトル」は、次のように抽出する。まず、データを短い時間毎に分割する、フレーム化を行う。続いて、フレーム化したデータに対し、窓関数処理を行う。その後、高速フーリエ変換 (Fast Fourier Transform; FFT) を行い、周波数スペクトルを抽出する。最後に、抽出した周波数スペクトルに対し絶対値をとることで、振幅スペクトルが算出できる。

また、本研究にて使用する「生データ」は、図3.1のとおり、時系列データに対してフレーム化処理 (3.1.1節) のみを行ったデータを指す。

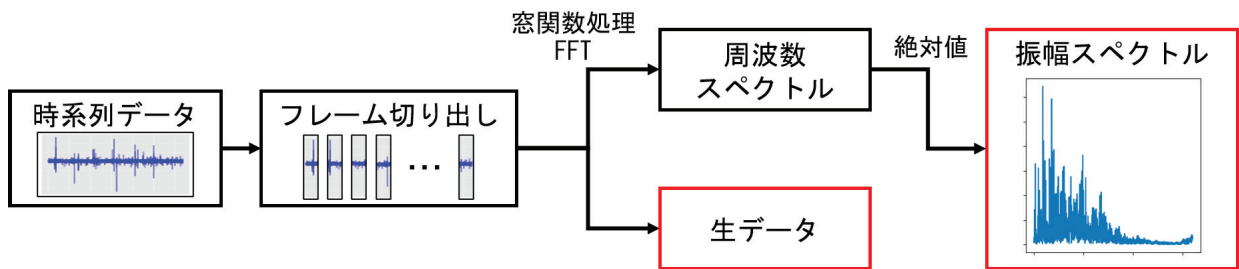


図 3.1 振幅スペクトル抽出の流れ

3.1.1 フレーム化処理

はじめに、時系列データをフレームと呼ばれる短時間のデータに分割する。一般的に、波形信号データにおけるフレーム化処理を行う場合、時間的に変化するスペクトルを詳細に把握するために、フレームを一定時間オーバーラップさせながら行う。このフレームの時間長をフレーム長、オーバーラップする時間幅の長さをフレームシフトと呼ぶ。フレーム長とフレームシフトはフレーム化を行う上で必要不可欠なハイパーパラメータであり、対象となるデータの種類やタスクによって適切に設定する必要がある。

3.1.2 窓関数処理

続いて、分割したフレームに対して窓関数を乗算する、窓関数処理を行う。分割したフレーム中の始端付近と終端付近でデータの値が連続でない場合、後述する離散フーリエ変換の結果が不連続性に影響され、振幅スペクトルの抽出結果に大きく影響する。窓関数処理は、こうした不連続性を吸収するために行われる処理である。一般的な信号処理の分野では、ハミング窓 (図 3.2) やハン窓 (ハンニング窓, 図 3.2) などの窓関数を用いられる。本研究ではこのうちハン窓を使用している。なお、ハミング窓は式 (3.1) で、ハン窓は式 (3.2) で表される。 x はフレーム内インデックスを標準化したものである。

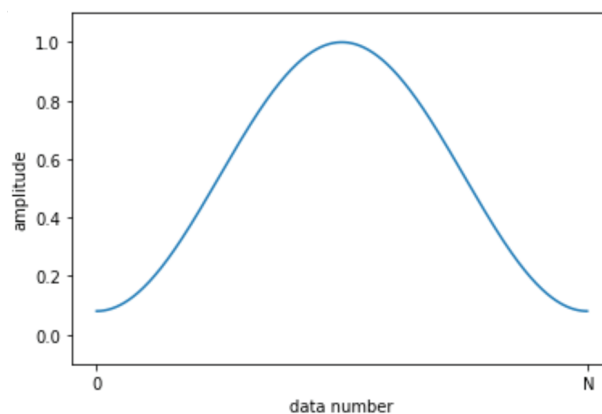


図 3.2 ハミング窓

$$w_{ham}(x) = 0.54 - 0.46 \cos 2\pi x, \quad (0 \leq x \leq 1) \quad (3.1)$$

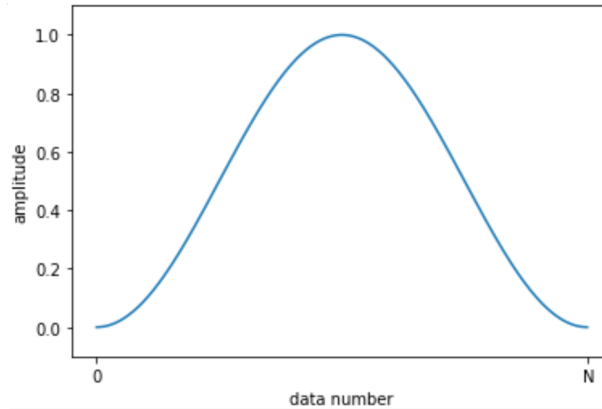


図 3.3 ハン窓 (ハニング窓)

$$w_{han}(x) = 0.5 - 0.5 \cos 2\pi x, \quad (0 \leq x \leq 1) \quad (3.2)$$

3.1.3 離散フーリエ変換

時系列データに対してフレーム化処理および窓関数処理を実施した後、振幅スペクトルを取得するために離散フーリエ変換 (Discrete Fourier Transform; DFT) を行う。離散フーリエ変換を行うことで、時系列データの持つ時間情報が周波数領域でのスペクトル情報に変換される。離散フーリエ変換は、式 (3.3) で表される。

$$X(\omega) = \sum_{t=0}^{N-1} x(t) e^{-j \frac{2\pi\omega t}{N}} \quad (3.3)$$

なお、計算機上で実際に計算を行う際には、離散フーリエ変換をより高速に行うアルゴリズムである高速フーリエ変換 (Fast Fourier transform; FFT, [33]) により算出する 경우가ほとんどである。

3.1.4 振幅スペクトル

3.1.3 節にて述べた離散フーリエ変換を時系列データに対して行うと、周波数スペクトルとよばれるスペクトル情報が出力される。周波数スペクトルは、時系列データの振幅情報と位相情報を保持するスペクトル情報である。振幅スペクトルは、周波数スペクトルの絶対値を取ることで求められる。

3.2 Principal Component Analysis; PCA

主成分分析 (Principal Component Analysis; PCA) とは、多次元データのもつ情報を、低次元データ空間に情報を圧縮する手法の一つである [24]。PCA の概略を図 3.4 に示す。この手法は、解析す

るデータの持つ特徴量次元数が大きい場合に、その次元数を削減するために用いられることが多い。PCA は、データの分散に着目し、新しい座標軸を求める手法である。はじめに、入力データの特徴空間での分布において平均値をとる点を通り、分散の最も大きい方向の直線を第 1 主成分の座標軸（図 3.4 右側における PC1）と定義する。続いて第 1 成分に直交し、かつ第 1 主成分に次いで分散が大きくなる方向の直線を第 2 主成分の座標軸（図 3.4 右側における PC2）と定義する。同様に第 3 主成分、第 4 主成分、…と算出していく。全情報量のうち、第 m 主成分が保持する元の情報量の割合を第 m 主成分の寄与率とよぶ。また、第 1 主成分から第 m 主成分までの寄与率の和を、第 m 主成分までの累積寄与率とよぶ。この寄与率および累積寄与率を用いることで、全情報量をどの主成分まででどの割合で再現できているかを算出できる。

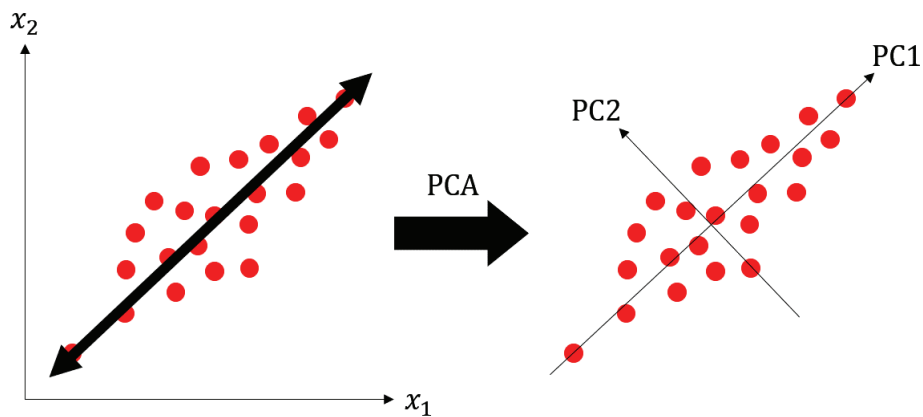


図 3.4 PCA

本研究においては、「生データ」に対して、その周波数方向の特徴次元数を削減するために利用する。本研究において実験の際に使用する次元圧縮後の特徴量の次元数 m は、累積寄与率が 95% もしくは 99% など、一定の数値以上となる第 m 主成分を基準として選択する。

第 4 章

異常検知・故障予測における統計手法・機械学習および深層学習

時系列データを入力とする分類・回帰などのタスクにおいて、さまざまな統計的手法や機械学習技術、および深層学習技術が適用されてきている。本章では、本研究において使用する機械学習技術である Hotelling's T-square 法, Gaussian Mixture Model, K-Nearest Neighborhood, One Class SVM, および深層学習技術として基本となる Neural Network (NN) と, NN の応用構造である Recurrent Neural Network, Temporal Convolutional Network, Autoencoder とその複合手法について詳説する。

4.1 従来手法による異常検知

本節では、従来より用いられている統計的手法および機械学習に基づく異常検知手法のうち、本研究で用いている 4 種類の手法について、その特徴および異常度の導出手法を紹介する。

4.1.1 Hotelling's T-square 法

Hotelling's T-square 法 (ホテリングの T^2 法) とは、統計的概念に基づく異常度算出手法の一つであり、対象となるデータセット中の異常データ群が、正常データ群と比較して極少数であるか、もしくは正常データ群のみであると仮定できる場合に利用される手法である [10]。データセット D は、 $D = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$ の N 個のデータで構成され、各サンプル \mathbf{x}^i は M 次元のベクトル $\{x_1^i, x_2^i, \dots, x_M^i\}$ であるとする。このとき、データセット D 中のデータ群は、次の式 (4.1) のように平均 μ 、共分散行列 Σ の多次元正規分布の確率密度関数 $N(\mathbf{x}|\mu, \Sigma)$ に従うと仮定する。

$$N(\mathbf{x}|\mu, \Sigma) = \frac{|\Sigma|^{-\frac{1}{2}}}{(2\pi)^{\frac{M}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right\} \quad (4.1)$$

学習用のデータセット D 中には異常データ群はほぼ含まれていないという仮定より、確率密度関数

$N(\mathbf{x}|\mu, \Sigma)$ は正常データ群の状態モデルといえる。このとき、未知のデータ \mathbf{x}' に対する異常度 $a(\mathbf{x}')$ は、次の式 (4.2) で算出できる。ここで、式 (4.2) を展開し、異常度の変化に大きな影響を及ぼさない定数項を削除することで、式 (4.3) のように近似することが可能である。一般的に Hotelling's T-square 法を用いる場合には、式 (4.3) により異常度を算出するが多い。

$$a(\mathbf{x}') = -\ln N(\mathbf{x}'|\mu, \Sigma) \quad (4.2)$$

$$a(\mathbf{x}') = (\mathbf{x}' - \mu)^T \Sigma^{-1} (\mathbf{x}' - \mu) \quad (4.3)$$

異常度 $a(\mathbf{x}')$ は、 D 中に出現する確率の高い正常データであれば小さな値を、出現確率の低い異常データであれば大きな値をとる。また、各データの特徴量次元数 M に対して D 中のデータ数 N が十分に大きい場合には、異常度 $a(\mathbf{x}')$ の分布は自由度 M 、スケール因子 1 のカイ二乗分布に従う。自由度 $k = 1, 2, 4, 8$ におけるスケール因子 1 のカイ二乗分布図を図 4.1 に示す。横軸が異常度、縦軸が確率密度を示しており、異常度が大きくなるにつれて確率密度が小さくなるということがわかる。このことを利用し、一般的な異常判定に使用する際は、閾値 a_{th} に異常データの出現率 α (ただし、 $\alpha < 1$ である) を設定し、カイ二乗分布により算出する。そして、未知データより算出した異常度が、閾値 a_{th} を超えた場合に警報を出すことにより、正常・異常の判定を行う。

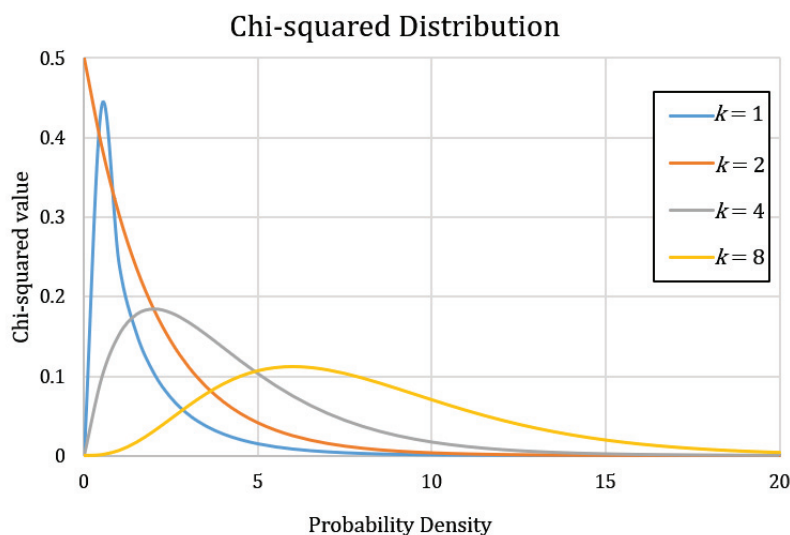


図 4.1 自由度 $k = 1, 2, 4, 8$ におけるカイ二乗分布の確率密度関数

Hotelling's T-square 法と同様にマハラノビス距離を利用した、異常検知などに用いられる機械学習手法として、マハラノビス・タグチ法 (MT 法) が挙げられる。MT 法も Hotelling's T-square 法と同様に、正常データ群の分布空間を基準とし、正常データ群からのマハラノビス距離を利用した異常検知を行う手法である [34]。

4.1.2 Gaussian Mixture Model; GMM

混合ガウス分布モデル (Gaussian Mixture Model; GMM) は、与えられたデータセットを K 個の正規分布 (ガウス分布) $N(\mathbf{x}|\mu_k, \Sigma_k)$ (ただし, $k = 1, 2, \dots, K$) の重畳により表現する混合モデルである [10]. データセットに対して GMM を適用し, それぞれの正規分布クラスに所属する確率をサンプルごとに算出することで, クラスタリングやパターン認識などに応用されることが多い.

異常検知に用いる場合, 前節と同様, 学習データセット D に異常データ群がほぼ含まれていないという仮定に基づいて混合ガウス分布モデルを学習し, 推論時にはモデルにより算出したデータごとの重み付き対数事後確率を異常度として出力する. はじめに, 混合ガウス分布の式を, データセットから最尤推定法により式 (4.4) のように算出する.

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k N(\mathbf{x}|\mu_k, \Sigma_k) \quad (4.4)$$

式 (4.4) において, K を正規分布の数, π_k は各正規分布の重みを示す. ただし, $\pi_k > 0$, $\sum_{k=1}^K \pi_k = 1$ である. その後, 予測するサンプルに対して混合ガウス分布モデルにより重み付き対数確率 $p_{ln}(\mathbf{x})$ を, 式 (4.5) により算出する.

$$p_{ln}(\mathbf{x}) = \ln\left(\sum_{k=1}^K \pi_k N(\mathbf{x}|\mu_k, \Sigma_k)\right) \quad (4.5)$$

ここで, 算出された重み付き対数事後確率は, そのデータが異常であればあるほど数値が小さくなる. 本論文における異常度は, その数値が高いほど異常を示すものとするため, 本論文における GMM による異常度 $a(\mathbf{x}')$ は式 (4.6) により算出する.

$$a(\mathbf{x}') = -p_{ln}(\mathbf{x}') \quad (4.6)$$

4.1.3 K-Nearest Neighborhood; K-NN

k 近傍法 (K-Nearest Neighborhood; K-NN) は, もっとも単純な機械学習手法であり, かつノンパラメトリックな分類手法である [26]. lazy learning の一種で, 教師あり学習のアルゴリズムである. 学習時には, 多次元の特徴空間における学習データの各点の位置とクラスラベルのみをモデルが学習する. 分類時には, 未知のサンプルの特徴空間上の位置と, 学習データの各点の位置とのユークリッド距離を算出し, k 個の最近傍のデータが属するクラスラベルに分類する. 図 4.2 において, 緑色の点を未知のサンプルとして k 近傍法を適用する場合, $k = 3$ のとき Class 1 に, $k = 5$ のときに Class 2 に分類される.

異常検知に用いる際には、前節と同様、学習データセット D に異常データ群がほぼ含まれていないという仮定に基づいて K-NN を学習し、推論時には k 個の最近傍のデータとの距離の合計値、もしくは平均値を異常度として出力することで実現できる。

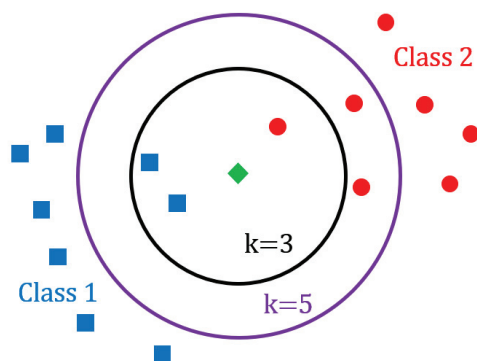


図 4.2 k 近傍法の例. この場合、未知のサンプル（図中緑色の点）は、 $k = 3$ のとき Class 1 に、 $k = 5$ のときに Class 2 に分類される。

4.1.4 One Class SVM; OC-SVM

One Class Support Vector Machine (OC-SVM) は、Support Vector Machine (SVM)[27] と呼ばれるパターン認識アルゴリズムを利用した異常検知手法である [35]。学習の際のデータセットに含まれるサンプルの全て、もしくはほぼ全てが正常であるという仮定のもと、学習データに対するラベルは正常クラスの 1 クラスのみとして学習し、異常検知時には識別境界を決定することでその境界をもとに異常検知を行う。この性質を用いることで、主に外れ値の検知において性能を発揮する手法である。図 4.3 に One-Class SVM による異常検知の例を示す。学習時には、学習データ D が全て正常クラス

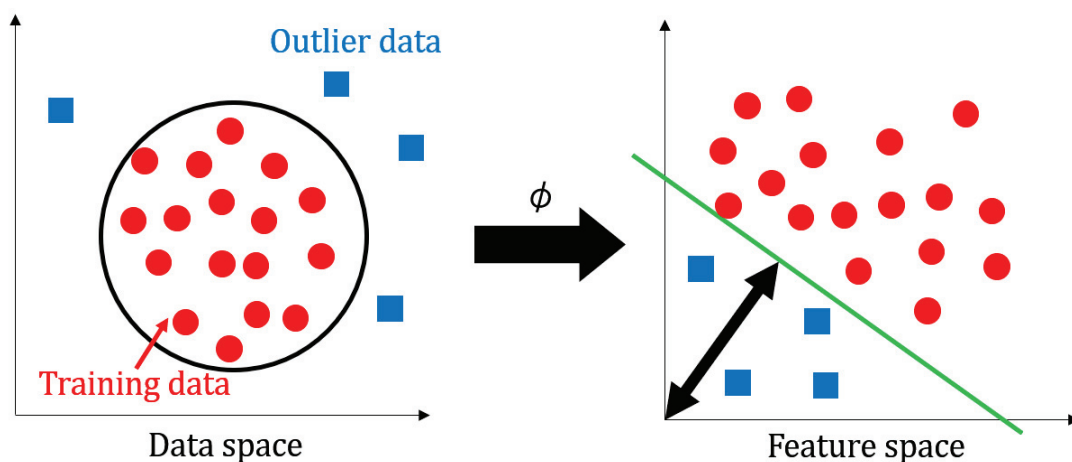


図 4.3 One-Class SVM の適用例. データに対しカーネル ϕ を適用し、入力データの空間（図左）から特徴空間（図右）に写像することで、正常・異常の識別境界を設定する。

に属し，原点のみが異常クラスに属するとして学習を行う．この際，図 4.3 のようにカーネルトリックを用いて特徴空間にデータを写像し，学習データは原点から遠い位置に写像する．写像が行われた特徴空間において，原点とのマージンを最大化するように，識別境界の設定を行うよう学習を行う．推論時には，学習データと類似したデータを入力した場合は特徴空間において原点から遠い位置に，学習データと類似しないデータを入力した場合は原点に近い位置に集まる．この性質を利用し，識別境界を用いて正常・異常の分類を行う．

4.2 Neural Network; NN

Neural Network (ニューラルネットワーク; NN) とは，人間の脳で行われる神経伝達の構造を簡易的に模した数学モデルで，現在パターン認識やクラスタリングなど様々な分野に応用されている．本節以降では，単純な NN の構造である多層パーセプトロンと，多層パーセプトロンの構造を応用し，連続情報を扱う課題に特化した NN モデルについて紹介する．

4.2.1 多層パーセプトロン

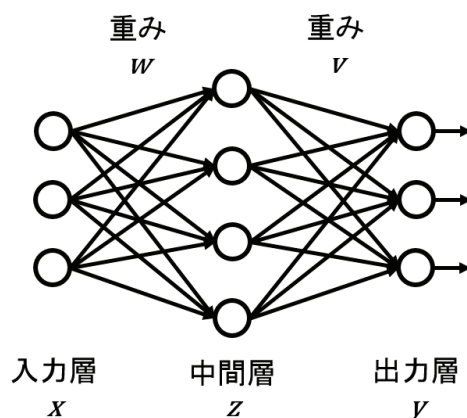


図 4.4 多層パーセプトロンの構造例

多層パーセプトロン (Multi-Layer Perceptron; MLP) とは，順伝播型 NN (Feed Forward Neural Network; FFNN) の一種であり，NN の基本的な構造として挙げられる．多層パーセプトロンは，入力パターンを入力とする素子（以下，ノードと記す）の集合（以下，入力層と記す）と入力層のノードからの信号の加重和を入力とするノードの集合（以下，中間層と記す），そして中間層からの加重和を入力とし，出力を行うノードの集合（以下，出力層と記す）の 3 種類のノード層を持つ．多層パーセプトロンには，1つの入力層および出力層と，1つ以上の中間層，少なくとも合計 3つのノード層から構成される．多層パーセプトロンでは，とある層のノードの値は，前の層のノードの値の線形重み付き和に，活性化関数 f を乗算して出力したものとなる．これを順伝播という．図 4.4 に，ノード層が 3 層からなる多層パーセプトロンの構造例を示す．このような多層構造と，非線形な活性化関数を用いる

ことで、線形分離可能でないデータの識別を行うことができる。図 4.4 における中間層 z の i 番目のノード z_i の値について、入力層 x の j 番目のノード x_j から z_i への重みを w_{ij} とすると、

$$z_i = f\left(\sum_j w_{ij}x_j + w_{i0}\right) \quad (4.7)$$

という式で表せる。ただし、 w_{i0} はバイアス項である。この計算は、出力層のノードを含む全ての場
合において同じように行われる。中間層への活性化関数 f に用いられるものとして、シグモイド関数
(式 4.8) や \tanh 関数 (式 4.9), Rectified Linear Unit (ReLU, 式 4.10) などが挙げられる。また、出
力層への活性化関数 f に用いられるものとして、ソフトマックス関数 (式 4.11) などが挙げられる。

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4.8)$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.9)$$

$$\text{relu}(x) = \begin{cases} 0, & x \leq 0 \\ x, & x \geq 0 \end{cases} \quad (4.10)$$

$$\text{softmax}(x) = \frac{e^{x_i}}{\sum_i e^{x_i}} \quad (4.11)$$

順伝播により出力した結果と、期待される結果との間には誤差が生じる。この誤差に基づいて、出力層
から中間層へ、中間層から入力層へと逆方向に誤差を伝播し、誤差が小さくなるように重みとバイア
スを更新する。これを誤差逆伝播法 (Back Propagation; BP) という。重み w_{ij} の更新式は、出力結
果と期待される結果の二乗誤差を E_i 、学習率を η とすると、次の式 4.12 で表せる。

$$\hat{w}_{ij} = w_{ij} - \eta \frac{\partial E_i}{\partial w_{ij}} \quad (4.12)$$

順伝播と誤差逆伝播法を繰り返すことで、誤差を減少させることができ、識別や認識の精度を向上さ
せることができる。

4.2.2 Recurrent Neural Network; RNN

多層パーセプトロンのような FFNN では、テキストや音声などの連続データにおいて重要な情報
である、隣り合う時刻のデータの相関を用いることができないという問題がある。この問題を解決
するため、再帰型ニューラルネットワーク (Recurrent Neural Network; RNN) が提案されている。
本節では RNN の基本的な構造について述べ、RNN の一種であり本研究で用いる Long Short Term
Memory について説明する。

RNN の一般的な構造を図 4.5(a) に示す。また、RNN の構造を時間方向に展開したものを図 4.5(b)
に示す。

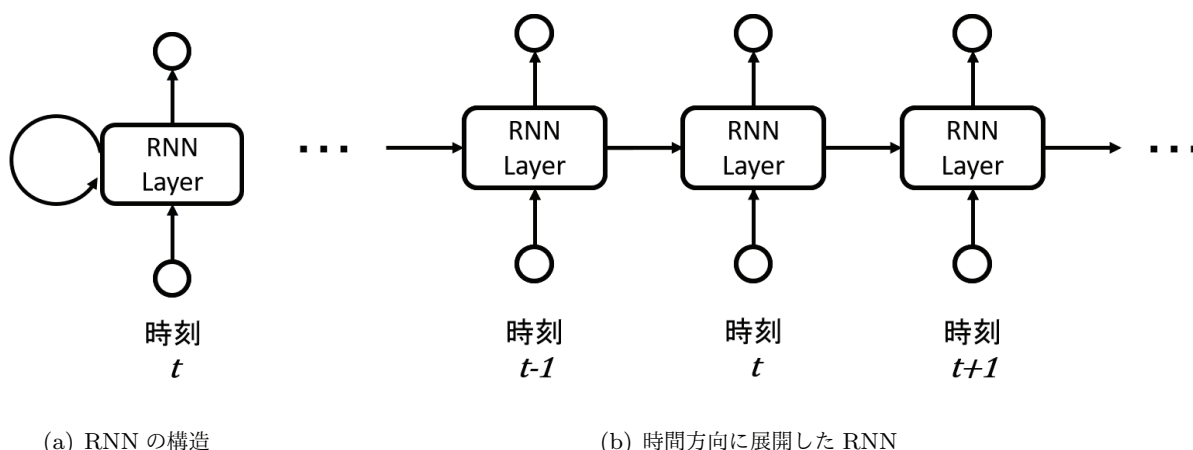


図 4.5 RNN の構造例

連続データの持つ時刻間の相関情報を学習するため、時刻 t における RNN のノードは、前の層の出力に加え 1 時刻手前、すなわち時刻 $t-1$ における RNN のノードの情報をフィードバックする構造を持つ。これにより、データの学習や推論を行う際にデータの持つ連続的な情報を利用することができる。また、学習時に全ての時刻のデータがそろっている場合、過去からだけでなく未来からの情報も伝播する双方向型構造 (Bidirectional RNN, BRNN) を持つこともでき、通常の RNN よりもより高い予測性能が期待できる。

時刻 t における RNN の中間層の i 番目のノードの出力 $z_i^{(t)}$ は、入力層のノード $x_j^{(t)}$ とその重み $w_{ij}^{(t)}$ 、時刻 $t-1$ におけるノードの出力ベクトル $\mathbf{z}^{(t-1)}$ 、そしてフィードバックの重みベクトル $\mathbf{v}^{(t-1)}$ を用いて、以下の式 (4.13) で表せる。

$$z_i^{(t)} = f\left(\sum_j w_{ij}^{(t)} x_j^{(t)} + \mathbf{v}^{(t-1)} \mathbf{z}^{(t-1)}\right) \quad (4.13)$$

RNN の学習は、多くの場合 Back Propagation Through Time (BPTT) [36] を用いる。これは、誤差逆伝播法を RNN へ応用したものであり、多層パーセプトロンにおける誤差逆伝播法の計算に加えてフィードバックした情報も伝播する。これにより RNN でも学習を行うことができるが、BPTT には問題点がある。通常の誤差逆伝播法に加え時間方向の学習も加わり、伝播する経路が大幅に長くなるため、最終的な誤差の値が非常に小さい値、もしくは非常に大きな値となる問題が同層数の多層パーセプトロンと比べて生じやすくなってしまおうというものである。このため、5-10 単位時間以上の特徴を学習するのは難しいとされている [37]。

Long Short Term Memory; LSTM

先に説明した誤差が消失する問題を解決するため、RNN の拡張構造である Long Short Term Memory (LSTM) が提案されている。LSTM は、図 4.5 で紹介した RNN の中間層のユニットを、LSTM ブロックと呼ばれるユニットに置き換えることで実現する。LSTM は時代の変遷とともにより良い連続データの学習のため構造の工夫がなされてきているが、本節においては Gers ら [37] が提案した LSTM ブロックについて扱う。LSTM ブロックの内部構造を図 4.6 に示す。

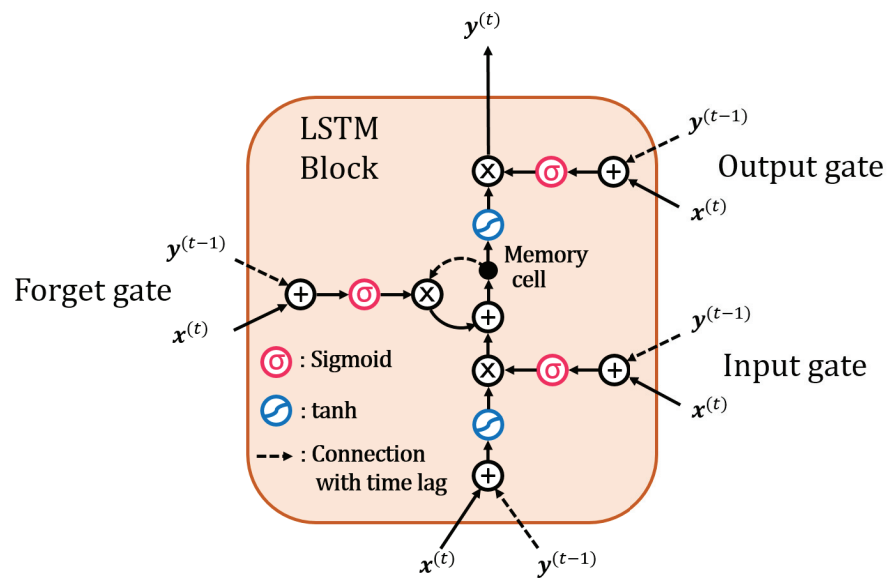


図 4.6 Gers ら [37] の LSTM ブロックの構造

LSTM ブロックは, input, output, forget と呼ばれる 3 種類のゲートとメモリセルと呼ばれるセルから構成されている. 図 4.6 中に示した, 時刻 t における LSTM ブロックへの入力 $\mathbf{x}^{(t)}$, input ゲートの出力 $\mathbf{i}^{(t)}$, forget ゲートの出力 $\mathbf{f}^{(t)}$, メモリセルの出力 $\mathbf{c}^{(t)}$, output ゲートの出力 $\mathbf{o}^{(t)}$, そして LSTM ブロックの出力 $\mathbf{y}^{(t)}$ のベクトルは, それぞれ式 (4.14), 式 (4.15), 式 (4.16), 式 (4.17), 式 (4.18), 式 (4.19) のように計算される. なお, 演算子 \odot は, 対応するベクトル要素同士の乗算を意味する. それぞれのゲートは時刻 $t-1$ からのフィードバックを用いる構造となっている. 入力信号と出力信号に対してそれぞれゲートを設置することで, 必要な情報のみを保持して学習することが可能となる [38]. また忘却ゲートの設置により, 出力への影響が小さい情報を忘却することで, RNN 構造で発生する課題の一つである, 誤差が消失してしまう現象を抑えることができ, 長期間の学習を行うことが可能となっている.

$$\mathbf{x}'^{(t)} = \tanh(W_{\mathbf{x}}\mathbf{x}^{(t)} + R_{\mathbf{x}}\mathbf{y}^{(t-1)} + \mathbf{b}_{\mathbf{x}}) \quad (4.14)$$

$$\mathbf{i}^{(t)} = \text{sigmoid}(W_{in}\mathbf{x}^{(t)} + R_{in}\mathbf{y}^{(t-1)} + \mathbf{b}_{in}) \quad (4.15)$$

$$\mathbf{f}^{(t)} = \text{sigmoid}(W_{for}\mathbf{x}^{(t)} + R_{for}\mathbf{y}^{(t-1)} + \mathbf{b}_{for}) \quad (4.16)$$

$$\mathbf{c}^{(t)} = \mathbf{i}^{(t)} \odot \mathbf{x}'^{(t)} + \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} \quad (4.17)$$

$$\mathbf{o}^{(t)} = \text{sigmoid}(W_{out}\mathbf{x}^{(t)} + R_{out}\mathbf{y}^{(t-1)} + \mathbf{b}_{out}) \quad (4.18)$$

$$\mathbf{y}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}) \quad (4.19)$$

4.2.3 Temporal Convolutional Network; TCN

Temporal Convolutional Network は、音声やセンサデータなどの系列データに対し、時間方向に 1 次元畳み込みを繰り返すことで学習する NN 構造である。1 次元の畳み込みをモデルに組み込む方法には、回帰タスクなどに適合した入力と出力が異なる構造とする方法もあるが、本研究では Encoder-Decoder TCN (ED-TCN) [39] の構造をベースとしたモデルを構築しているため、本構造について説明する。図 4.7 に ED-TCN の構造を示す。ED-TCN は、Encoder 部と Decoder 部の層数・

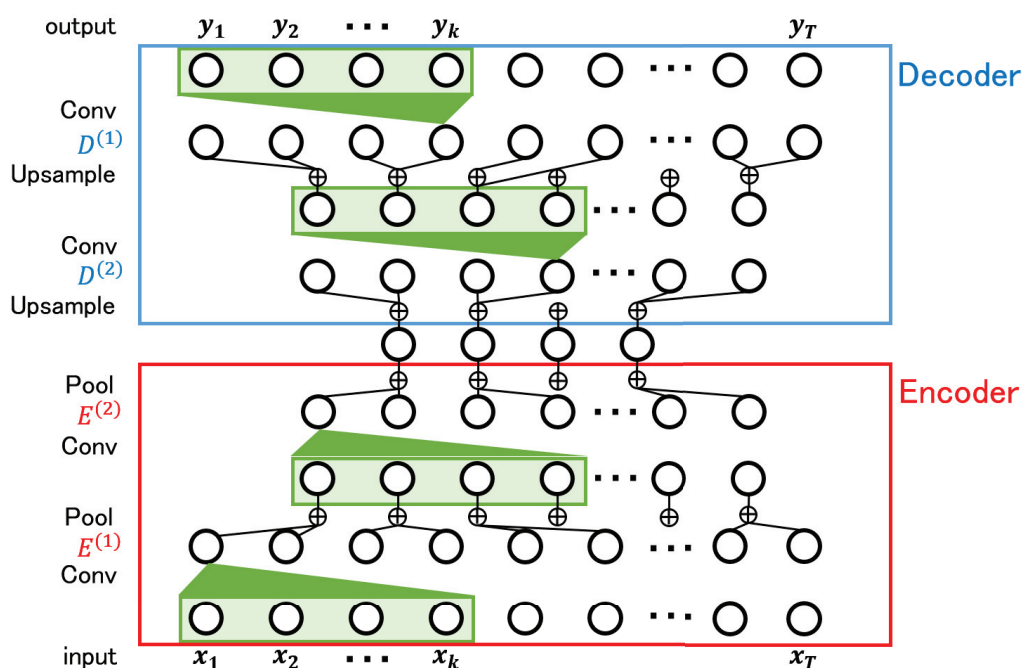


図 4.7 Encoder-Decoder TCN (ED-TCN)

次元数が線対称となるように設計された、畳み込み層、プーリング層およびアップサンプリング層により構成される Encoder-Decoder モデルである。ただし、図 4.7 中の x_k において、 k は畳み込みの長さ (カーネルサイズ) を示し、後述する実験を含め、本研究においてはサンプリング周波数との関係から $k = 128$ としている。Encoder の各層では、1 次元畳み込みとプーリングを行う。 l 番目の Encoder 層 $E^{(l)}, l \in \{1, 2, \dots, L\}$ は、1 つ前の層 $E^{(l-1)}$ 、畳み込みフィルタ $\mathbf{W} = \{W^{(i)}\}_{i=1}^{F_l}$ (F_l は l 層目における畳み込みフィルタ数)、バイアス項 \mathbf{b} および活性化関数 f を用いて、式 (4.20) のように表せる。

$$\mathbf{E}^{(l)} = f(\mathbf{W} * \mathbf{E}^{(l-1)} + \mathbf{b}) \quad (4.20)$$

ここで、本構造における畳み込み層の活性化関数には、式 (4.21) で示すような正規化 ReLU (normalized ReLU) を用いている。ただし、式中の ϵ は定数である。

$$f(x) = \frac{\text{relu}(x)}{\max(\text{relu}(x)) + \epsilon} \quad (4.21)$$

畳み込み処理後、最大プーリングを行うことで、データ長 $T^{(l)}$ は 1 つ前の Encoder 層の $\frac{1}{2}$ となる ($T^{(l)} = \frac{1}{2}T^{(l-1)}$)。この処理により、時間窓の長い系列データを効果的に学習することができる。Decoder における各層の構成は Encoder 層と構造的に同様であるが、Encoder 層では各層における畳み込み後にプーリングを行うのに対し、Decoder 層では畳み込み後に、 l 番目の Decoder 層 $D^{(l)}$, $l \in \{L, \dots, 2, 1\}$ におけるデータ長が 1 つ前の層 $D^{(l+1)}$ の 2 倍 ($T^{D^{(l)}} = 2T^{D^{(l+1)}}$) となるようにアップサンプリングを行う。

4.3 深層学習による異常検知

本節では、本研究で用いる深層学習モデルである Autoencoder について、その構造および異常検知における利用方法について説明する。

4.3.1 Autoencoder; AE

Autoencoder (AE) とは、NN 構造を用いた教師なし学習アルゴリズムの一つであり、次元圧縮や再構成誤差に基づくパターン認識などに用いられる。図 4.8 には、FFNN のみで構成された AE (FFNN-AE) の構成例を示す。AE は中心となる中間層を基準に対称な層数および次元数によるノードで構成される Encoder-Decoder 構造である [11]。AE は、入力層と出力層に同じデータを用いて学習を行うことで、推論時には入力されたデータを再構成して出力する。

学習を行ったモデルを用いて次元圧縮を行う際には、中心となる中間層のノード (図 4.8) における「中間表現」部) をモデルの出力層として扱うことで、入力層よりも少ないデータ次元の圧縮表現を獲得することが可能である。また、NN 構造を用いるため、主成分分析などでは難しい非線形な次元圧縮を行うことが可能である。

異常検知分野において AE 構造を活用する場合には、先に述べた次元圧縮した特徴を異なる異常検知モデルに入力する手法の他にも、AE 構造のもつ性質を活用して異常度を算出する手法がある。正常サンプルのみにより構成されたデータセットを用いて AE を学習することで、正常状態とは挙動が異なる、故障に近いデータを入力した場合に AE がうまく再構成できず、再構成誤差が高くなる性質を利用する [7]。この再構成誤差を異常度として扱うことで、入力されたデータに基づき異常検知を行う。

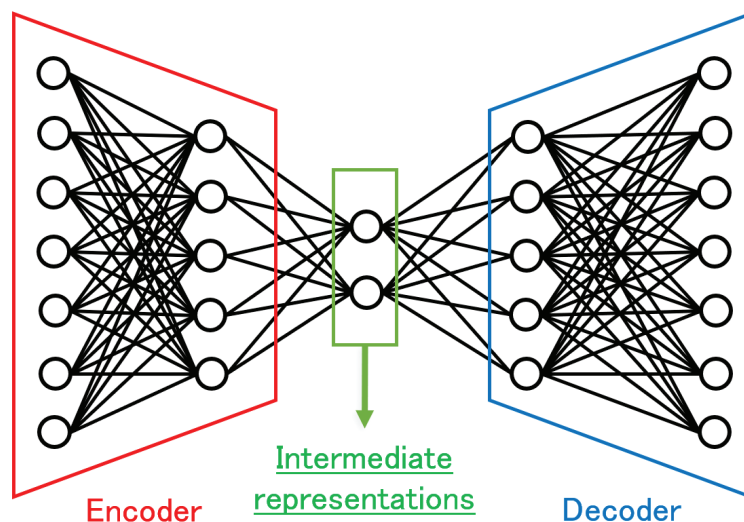


図 4.8 Autoencoder (AE) の構造.

4.3.2 LSTM Autoencoder; LSTM-AE

LSTM-Autoencoder(LSTM-AE, 図 4.9) とは, 中間表現部を含む中間層のノードを LSTM ブロックにより構成した, 再帰型構造を持つ AE である. これにより, AE による非線形な次元圧縮に加え, 時系列情報も同時に次元圧縮の際の学習に用いるように設計する [40]. このため, データ中の文脈的異常に対応した次元圧縮を行うことが期待できる. ここで, 本論文における LSTM-AE の中間表現は, Encoder 部のうち, 最後の入力が行われる時間ステップの LSTM ブロックの出力のことを指す. また図 4.9 に示している通り, Decoder 部における LSTM ブロックへの入力, は, 前の時刻の出力と LSTM-AE の中間表現である. さらに, Encoder 部と Decoder 部における LSTM ブロックは独立した異なる層であり, 重みの共有などは行っていない.

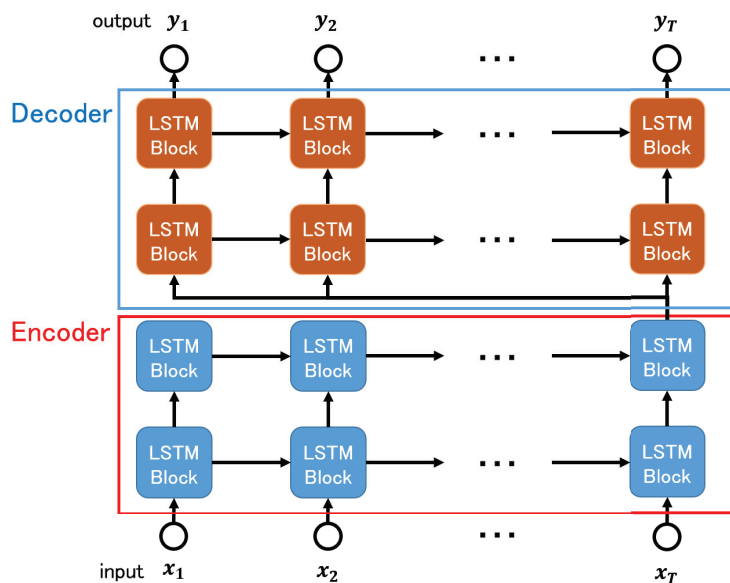


図 4.9 LSTM-AE の構造. 図 4.8 における Encoder 層および Decoder 層を LSTM ブロックにより構成した構造を持つ.

4.3.3 TCN-Autoencoder(TCN-AE)

本論文において異常検知に利用する構造である TCN-AE は、構造自体は先述した ED-TCN と同様である。しかし、[39]において定義される ED-TCN は、用いられているタスクが識別タスクであるため、時刻 t における出力 \hat{Y}_t は、Decoder の最終層 $D^{(1)}$ に対し softmax 関数を掛け合わせている。一方本研究では、ED-TCN の構造を Autoencoder として使用する。すなわち入力データと出力データに同一の系列データを使用するため、出力層 Y において用いる関数を線形関数に変更している。出力層は以下の式 (4.22) により算出される。ただし、 U は $D^{(1)}$ における重みベクトル、 c はバイアス項である。

$$\hat{Y}_t = U * D_t^{(1)} + c \quad (4.22)$$

第 5 章

Autoencoder 構造を用いた異常検知・故障予測手法

本研究において提案する異常検知および故障予測手法は 2 種類である。以下より、各手法の構造および異常度算出の流れについて説明する。

5.1 手法 1 : Autoencoder と GMM による異常検知・故障予測手法

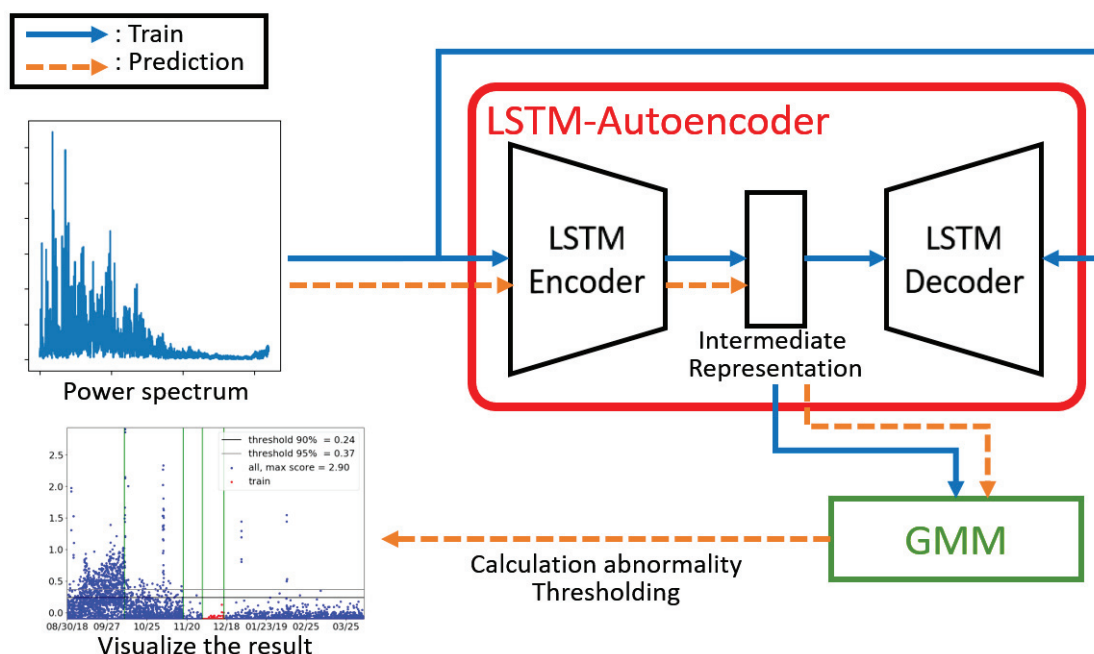


図 5.1 提案手法 1 の構造. LSTM-AE と GMM の 2 種類により構成されている。LSTM-AE を用いて時系列データの振幅スペクトルより特徴を抽出し、抽出した特徴から GMM により異常度を算出する。

提案手法の 1 つ目は、LSTM-AE を特徴抽出手法に、GMM を異常度算出手法として採用した手法である。図 5.1 に提案手法の流れを示す。本手法は、振幅スペクトルを入力として LSTM-AE による特徴抽出を行い、その後 GMM による異常度算出を行う。算出された異常度に対して閾値を設定し、一定期間閾値を超えた場合に故障が予測されるアラートを発生する、という流れである。実線の矢印で示した流れが異常度算出を行うためのモデルの学習を行う段階 (学習フェーズ)、破線の矢印で示した流れが実際に故障予測を行う段階 (故障予測フェーズ) である。

入力データについて、LSTM-AE には振動データを変換することで得られる振幅スペクトルを使用し、GMM への入力には LSTM-AE の中間層より得られる中間表現を用いる。

学習フェーズ

学習フェーズにおいては、正常状態におけるデータ群のみからデータを選出して学習用データとして用いる。このことにより、モデルが正常状態におけるデータの分布を学習し、正常状態のデータが入力されれば低い異常度を、故障に近い状態のデータが入力されれば高い異常度を出力するように設計している。

学習フェーズでは、はじめに LSTM-AE を学習し、その後 GMM の学習を行うという 2 段階での学習を行う。まず、LSTM-AE について、入力層と出力ラベルの双方に振動データより得られた振幅スペクトルを用いて学習する。続いて、学習した LSTM-AE モデルに対して同様に学習データを入力し、中間層より得られる中間表現を取得する。学習フェーズに用いるデータは正常状態のデータのみであるため、LSTM-AE の中間層より取得する中間表現も正常状態のデータのみである。この中間表現データを用いて、GMM を学習し、正常状態におけるデータの分布を学習する。最後に、学習した 2 つのモデルに対し、後述する故障予測フェーズにおいて LSTM-AE の中間表現が GMM の入力として伝播されるように設定する。

故障予測フェーズ

故障予測フェーズでは、学習を行った LSTM-AE により特徴量抽出を行い、GMM により異常度を算出した後、あらかじめ設定した閾値と比較して故障が近いかどうかを判定する。学習済みモデルに対し振幅スペクトルを入力したとき、正常なデータであれば異常度は低い値となり、故障状態に近ければ近いほど異常度が高くなる。しかし、センサ位置の変更や設備の一時停止など、本研究で扱う劣化に伴う故障以外の突発的な変化が起こった場合にも異常度が高くなる可能性があり、必ずしも異常度が閾値を超えた場合に故障が近いとはいえない。このことを踏まえ、実装の際には、突発的な変化に対してアラートを発生させないよう、一定期間に対して異常度の移動平均を算出し、この移動平均の値が閾値を超えた場合にアラートを発生させるように設計する。

5.2 手法 2 : Autoencoder による end-to-end 異常検知・故障予測手法

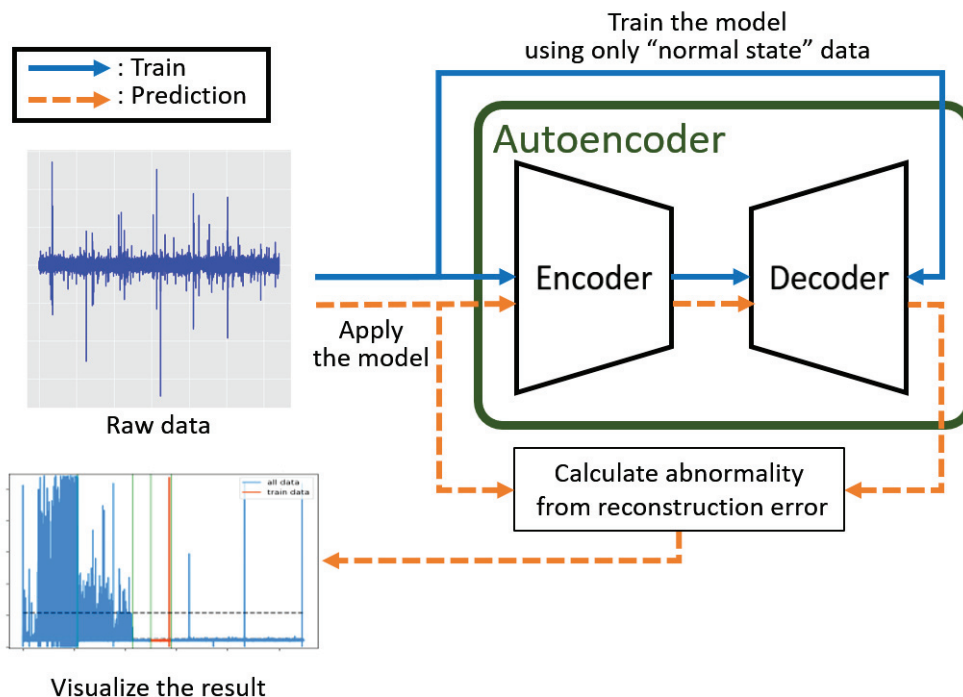


図 5.2 提案手法の構造. Autoencoder のみを用いた end-to-end 構造である.

提案手法の 2 つ目は, AE に時系列データの生波形信号データを入力し, 出力が異常度となるように設計した end-to-end な異常検知・故障予測手法である. 図 5.2 に提案手法の流れを示す. 4 章で紹介した 3 種類の Autoencoder 構造 (FFNN-AE, LSTM-AE, TCN-AE) を図中の「Autoencoder」部に適用し, モデルへの入力と, モデルによる出力との再構成誤差を異常度として算出する. 本手法では, 算出された異常度に対して閾値を設定し, 一定期間閾値を超えた場合に故障が予測されるアラートを発生する, という流れである. 実線の矢印で示した流れが異常度算出を行うためのモデルの学習を行う段階 (学習フェーズ), 破線の矢印で示した流れが実際に故障予測を行う段階 (故障予測フェーズ) である.

入力データについては, 先にも述べた通り, 時系列データより短い時間で切り出した生波形信号データを利用する.

学習フェーズ

学習フェーズにおいては、正常状態におけるデータ群のみからデータを選出して学習用データとして用いる。このことにより、モデルが正常状態におけるデータの分布を学習し、正常状態のデータが入力されれば低い異常度を、故障に近い状態のデータが入力されれば高い異常度を出力するように設計している。

故障予測フェーズ

異常度算出フェーズでは、はじめに学習済みモデルにデータを入力し、再構成誤差を算出する。データ長が T である入力データを $\mathbf{X} = \{x_1, x_2, \dots, x_T\}$ 、出力結果を $\mathbf{Y} = \{y_1, y_2, \dots, y_T\}$ としたとき、再構成誤差 $a(\mathbf{Y}; \mathbf{X})$ は次の式 (5.1) で計算される。

$$a(\mathbf{Y}; \mathbf{X}) = \sum_{t=1}^T (y_t - x_t)^2 \quad (5.1)$$

算出した再構成誤差 $a(\mathbf{Y}; \mathbf{X})$ は、そのまま対象データに対する異常度として扱う。

算出した異常度に基づいて故障予測を行う場合、以下のような手順によって行う。はじめに、一定期間の幅における異常度の移動平均を算出し、データ間での異常度の推移を平滑化する。つづいて、集計した時刻順に異常度の推移に基づき、正常・異常判定を行うための閾値を設定する。最後に、異常度の移動平均の推移が閾値を超えた場合に、故障が予測されることをアラートするようにシステムを設計する。

第 6 章

穴あけパンチ加工におけるパンチ刃の摩耗状態の検知

本研究では、5章で構築した手法による異常検知および故障予測の性能を評価し、有効性を検証するため、2種類のデータセットを用いた異常検知・故障予測実験を行った。6章および7章において、各データセットを用いた実験の内容および結果を示し、考察を行う。6章では、構築した手法による異常検知精度を検証するため、実際の加工機に設置したセンサデータを用いて穴あけパンチ加工におけるパンチ工具の摩耗状態の検知を行う実験について、そのデータおよび条件、提案手法における実験手法、条件、および結果を述べる。

6.1 使用するデータ

本章において使用するデータセットはデータセット1とデータセット2の2種類である。双方とも、穴あけパンチ加工において用いられる、パンチ工具の摩耗状態を計測するためにプレス機に設置されたセンサより取得したデータをもとに構築されている。

本実験におけるデータセットによるタスクは、センサデータを入力とした異常度算出モデルにより異常度を算出し、その推移をもとに3種類の刃先の摩耗状態を分類することである。すなわち、New状態とHalf-worn状態、Half-worn状態とWorn状態、そしてNew状態とWorn状態を分類することが目的となる。

6.1.1 データセット1

データセット1は、加工機のパンチ工具の付近に3軸加速度センサ、マイク、変位センサ2種類、圧力センサの計5種類の計測装置を設置し、計7次元の計測データを取得し、入力のためのデータとして構成している。データセット1のデータに対するラベルは、実際の刃先の摩耗状態によってNew（刃先が新品である状態、摩耗率0%付近）、Half-worn（刃先がやや摩耗した状態、摩耗率50%付近）、Worn（刃先が摩耗し、交換する直前の状態、摩耗率100%付近）という3種類が付与されている。そ

それぞれの状態において、2250 個分の穴を開けた際のデータを取得している。

6.1.2 データセット 2

データセット 2 は、加工機のパンチ工具の付近に、マイク、変位センサ 2 種類、圧力センサ、ロードセル（加重センサ）の計 5 種類の計測装置を設置し、計 5 次元の計測データを取得し、入力のためのデータとして構成している。データセット 2 は、1 個のパンチ刃を交換直後から完全に摩耗するまで一定のスピードで稼働させた際のセンサデータを取得したのち、5 種類の摩耗率 (0%, 25%, 50%, 75%, 100%) 付近のデータを抽出し、データセットとして構築したものである。データセット 2 のデータに対するラベルは、実際の刃先の摩耗状態によって New, Half-worn, Worn の 3 種類が付与され、さらに Half-worn 状態を 3 種類のより細かい 3 種類の状態（摩耗率 25% 付近, 50% 付近, 75% 付近）に細分化している。ただし今回の実験では、細分化されたラベルは使用せず、データセット 1 と同様に 3 種類のラベルとして使用する。

本データの取得条件を表 6.1 に示す。ただし、サンプリング周波数は、加速度センサ、マイクおよびロードセルは 20 kHz、この 3 種類以外のセンサは 2kHz である。本実験では、全てのセンサデータを同じ条件下で用いるため、全てのセンサデータのサンプリング周波数が 2kHz となるように、サンプリング周波数 20kHz である系列データに対しダウンサンプリングを行っている。また、ラベルごとのデータ数はほぼ同数である。

表 6.1 穴あけパンチ加工機に設置したセンサデータの取得条件

条件	設定値 (データセット 1)	設定値 (データセット 2)
取得したセンサ数 / データの次元数	5 個 / 7 次元	5 個 / 5 次元
統一後のサンプリング周波数	2kHz	
合計データ長 (point)	6,533,729 pt	4,500,000 pt
合計データ長 (秒換算)	約 326.68 秒	約 225.00 秒
付与されたラベル	3 種類	5 種類
実験に使用するラベル	3 種類	

6.2 実験条件

本実験における条件を表 6.2 に示す。本手法において異常度算出に使用する手法は、構築したモデルのうち、FFNN-AE と TCN-AE の 2 種類である。また、構築したモデルの性能を比較するため、4 章にて紹介した K-NN (KNN), OC-SVM (SVM), Hotelling's T-square 法 (T2) および GMM の 4 種類の機械学習アルゴリズムによる異常度算出モデルを構築し、異常度を算出した。ただし、K-NN における近傍数は $k = 3$ を使用している。

本実験では、特に AE 構造のモデル学習のため、はじめにデータセットごとに、各次元に対して平均 0・分散 1 となるように標準化を行う。その後、データセットを長さ 256point の短時間のデータに切り出し、入力データとして用いている。このとき、TCN-AE の入力データは (256 point, データの次元数) の 2 次元配列である。ここで、FFNN-AE と OC-SVM の入力データは、TCN-AE の入力データである 2 次元配列に対し時間方向に、256point から 128point にダウンサンプリングした後、1 次元配列に平坦化したものを使用する。すなわち、入力データの次元数は (128 point × データの次元数) となる。そして、K-NN, Hotelling's T-square 法および GMM 法の入力データは、TCN-AE の入力データである 2 次元配列を 1 次元配列に平坦化した後 PCA を行い、累積寄与率が 95% を超える最小次元数まで次元圧縮したデータを用いる。各手法に対する入力データの次元数は、表 6.2 中に示している。本実験の評価は、後述する Area Under Curve (AUC) と異常度算出時間の 2 種類の指標により行う。ただし、算出時間を求める際に条件を同様にするため、FFNN-AE と TCN-AE について、モデルの学習時には GPU を使用し、異常度算出時には機械学習アルゴリズム算出と同一の CPU を使用する。また、異常度間の分散に伴う異常検知への影響を吸収するため、200 データごとの単純移動平均と単純移動最小値を算出する。また、AUC は、異常度の単純移動最小値から計算する。

表 6.2 実験条件

条件	設定値 (データセット 1)	設定値 (データセット 2)
データ切り出し後の 1 データの長さ	256 point	
データ切り出しのオーバーラップ	128 point	64 point
データ切り出し後のデータ数	51,041	70,295
学習に使用するデータ数	8,520	7,029
TCN-AE への入力次元数	(256,7)	(256,5)
FFNN-AE, OC-SVM への入力次元数	896	640
上記以外のモデルへの入力次元数 (PCA による次元圧縮後の次元数)	419	347
PCA による次元圧縮基準	累積寄与率 95% 点	
移動平均および移動最小値の算出単位	200 データ	
モデル評価指標	AUC, 異常度算出時間	

6.2.1 AE モデル

AE の学習における設定条件を表 6.3 に、各データセットに対する TCN-AE と FFNN-AE の構造を表 6.4, 表 6.5, 表 6.6, 表 6.7 に示す. 各モデルに対する入力データの次元数の選定方法は、3 種類の入力データにより実験した結果のうち、後述する 3 種類の AUC が最も高いものを選定している.

各モデルの学習には、New 状態のラベルが付与されたデータのみを用いたが、New 状態全てのデータを学習に使用せず、奇数番目のデータを学習用、偶数番目のデータをテスト用とさらに分割して使用した.

表 6.3 AE の設定条件

条件	設定値
使用 OS	Windows 10
使用フレームワーク	Keras[42] (TensorFlow[41] backend)
TCN 部における活性化関数	Normalized ReLU
AE の出力における活性化関数	Linear
誤差関数	Mean Squared Error
最適化アルゴリズム	Stochastic Gradient Decent
学習率	0.01
AE 学習エポック数	2000 (with Early Stopping)

表 6.4 TCN-AE の構造 (データセット 1)

層名	フィルタサイズ	出力サイズ
input		(256,7)
Convolution 1D	16	(256,16)
Max Pooling 1D	-	(128,16)
Convolution 1D	- 32	(128,32)
Max Pooling 1D	-	(64,32)
Up Sampling 1D	-	(128,32)
Convolution 1D	16	(128,16)
Up Sampling 1D	-	(256,16)
Convolution 1D	7	(256,7)

表 6.5 FFNN-AE の構造 (データセット 1)

層名	出力サイズ
input	896
Full Connection	448
Full Connection	224
Full Connection	448
Full Connection	896

表 6.6 TCN-AE の構造 (データセット 2)

層名	フィルタサイズ	出力サイズ
input		(256,5)
Convolution 1D	16	(256,16)
Max Pooling 1D	-	(128,16)
Convolution 1D	32	(128,32)
Max Pooling 1D	-	(64,32)
Up Sampling 1D	-	(128,32)
Convolution 1D	16	(128,16)
Up Sampling 1D	-	(256,16)
Convolution 1D	5	(256,5)

表 6.7 FFNN-AE の構造 (データセット 2)

層名	出力サイズ
input	640
Full Connection	320
Full Connection	160
Full Connection	320
Full Connection	640

6.2.2 評価指標: Area Under Curve

本実験において使用する評価指標の一つである Area Under Curve (AUC) は、二値分類の性能を評価する指標であり、ROC 曲線 (Receiver Operating Characteristic curve、受信者動作特性曲線、図 6.1) と呼ばれるグラフより算出される [10]。ROC 曲線の算出方法は以下のとおりである。はじめに、複数個の異常判定の閾値 (カットオフポイント) における分類精度を計算する。ここで計算される精度は、異常サンプルを異常と判定できた比率を示す True Positive Rate (TP) と、正常サンプルを異常サンプルと誤判定した比率を示す False Positive Rate (FP) の 2 種類である。続いて、設定したカットオフポイント毎に縦軸が TP、横軸が FP となるグラフ上にプロットし、そのポイントを線で繋ぐ。このようにして導出された曲線が ROC 曲線である。AUC は、この ROC 曲線の下部面積にあたる。指標として使用する際は、AUC が大きければ大きいほど分類精度の高い異常検知モデルであることを示し、完全に分類できる場合には $AUC=1$ となる。

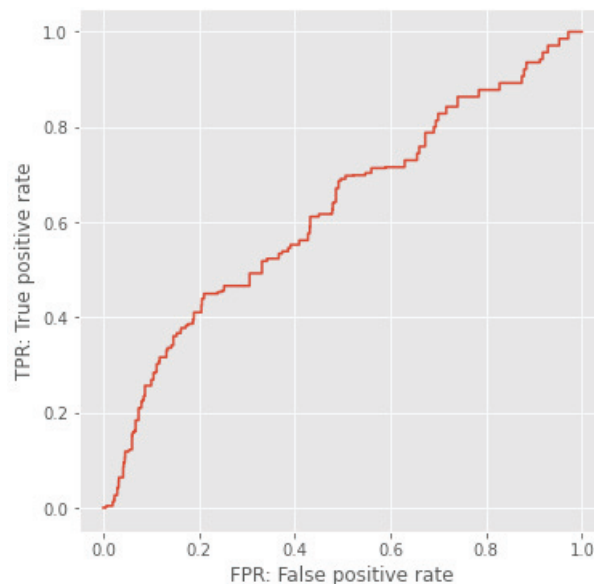


図 6.1 ROC 曲線

本実験では、ラベルは 3 クラス存在しているため、以下の 3 種類の ROC 曲線および AUC を算出し、評価に用いた。

- ROC_1, AUC_1: New 状態を 1 クラス目, Half-Worn 状態を 2 クラス目にして 2 値分類を行った場合の、移動最小値の ROC 曲線と AUC
- ROC_2, AUC_2: Half-Worn 状態を 1 クラス目, Worn 状態を 2 クラス目にして 2 値分類を行った場合の、移動最小値の ROC 曲線と AUC
- ROC_3, AUC_3: New 状態を 1 クラス目, Worn 状態を 2 クラス目にして 2 値分類を行った場合の、移動最小値の ROC 曲線と AUC

6.3 データセット 1 における実験結果

本節では、6 種類の手法によって算出されたデータセット 1 に対する異常度の推移のグラフと、移動最小値に基づく ROC 曲線、そして AUC と異常度算出時間の結果を示す。はじめに、6 種類の手法による異常度推移のグラフを、図 6.2 から図 6.7 までに示す。ただし、各図のうち上部の赤いグラフがデータごとの異常度、黄色のグラフが異常度の移動平均、下部のグラフが異常度の移動最小値の推移を示している。また、グラフ中に 2 本の縦線を描画し、データを 3 個の区間に分割している。このうち、緑色の縦線より左側の区間が New 状態、緑色の縦線と青色の縦線の間の区間が Half-Worn 状態、青色の縦線より右側の区間が Worn 状態のデータである。続いて、6 種類の手法によって算出された異常度の移動最小値に対する 3 種類の ROC 曲線のグラフを、図 6.8 から図 6.13 までに示す。ただし、各図のうち、左から ROC_1, ROC_2, ROC_3 のグラフをそれぞれ示している。最後に、6 種類の手法によって算出された異常度に対する 3 種類の AUC と、異常度算出時間を表 6.8 に示す。

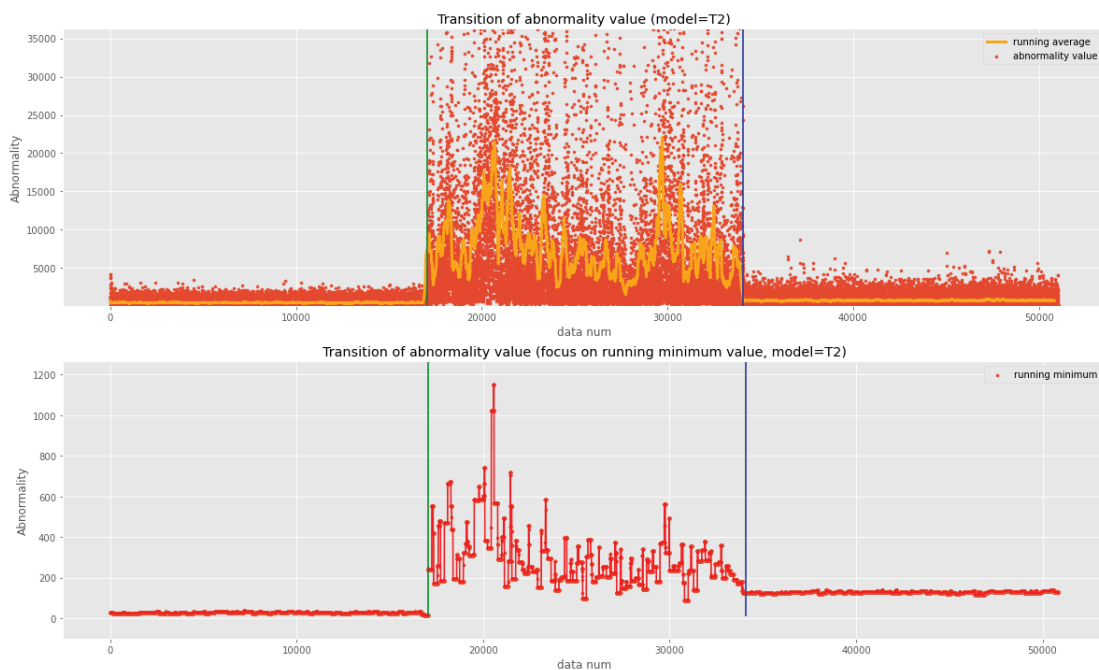


図 6.2 Hotelling's T-square 法による異常度算出結果 (データセット 1)

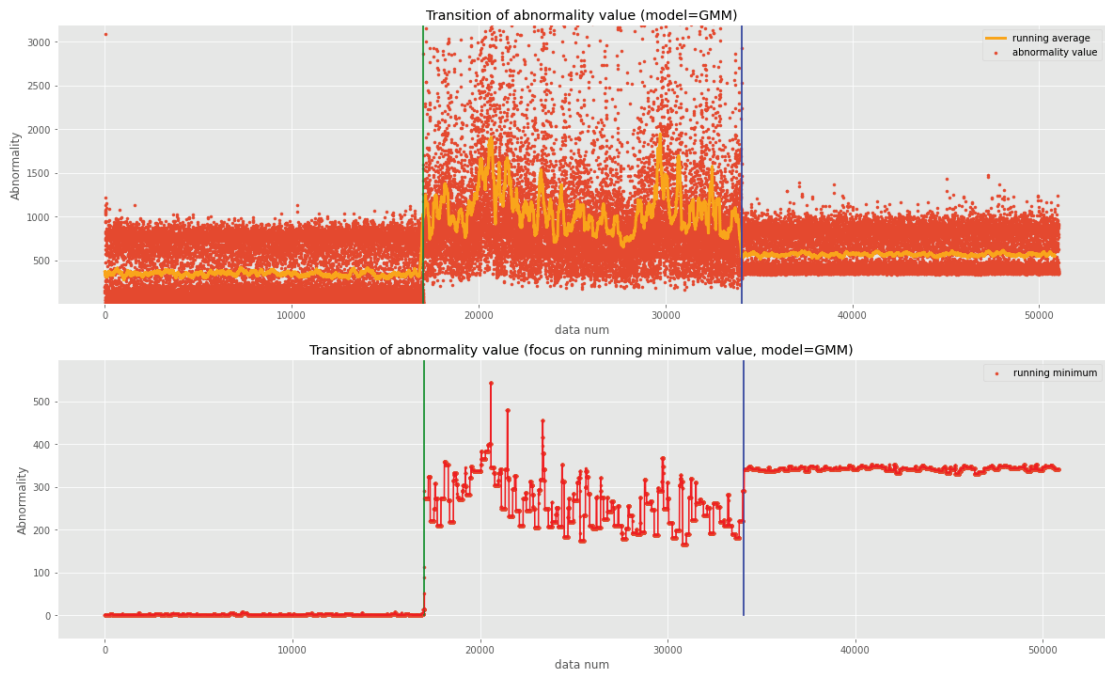


図 6.3 GMM による異常度算出結果 (データセット 1)



図 6.4 K-NN による異常度算出結果 (データセット 1)

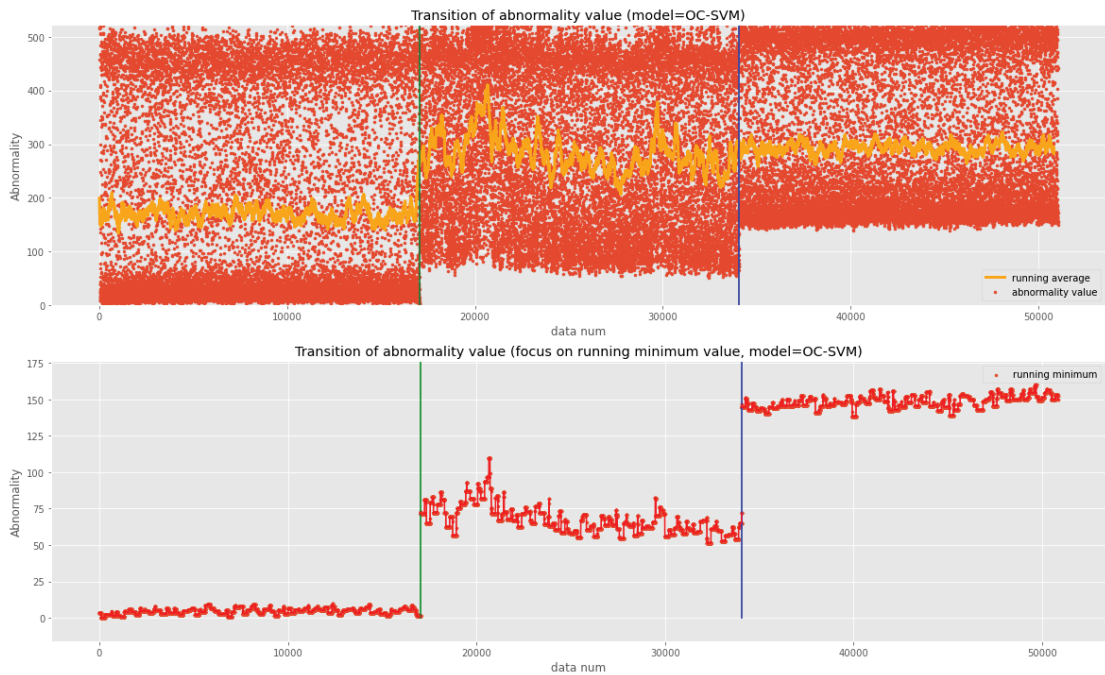


図 6.5 OC-SVM による異常度算出結果 (データセット 1)

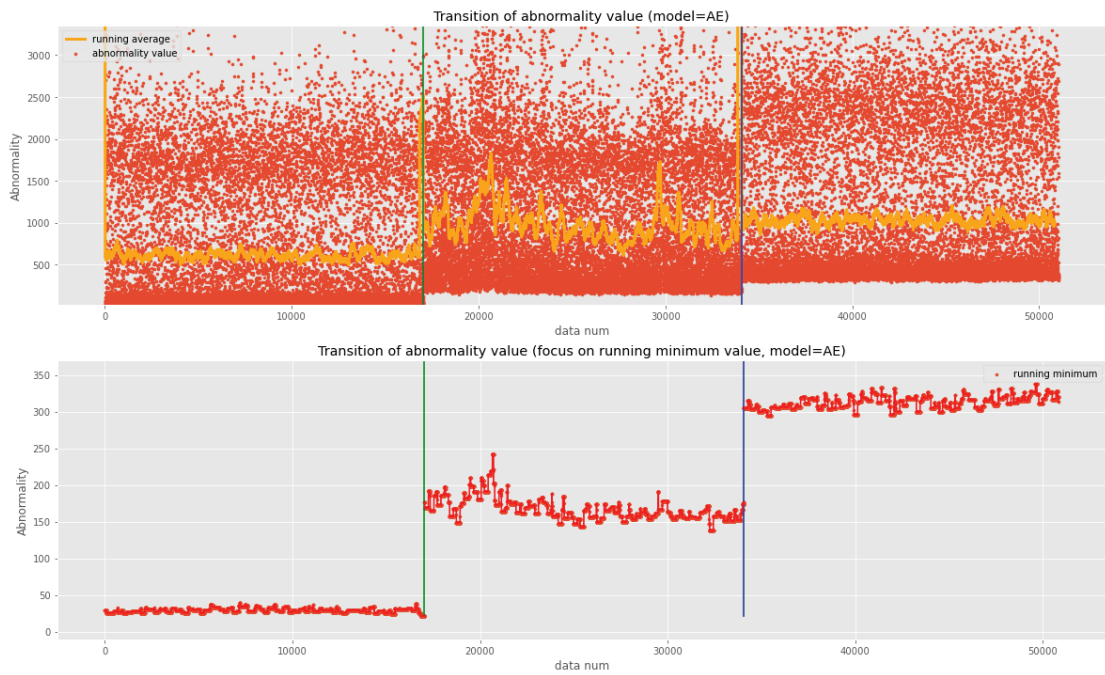


図 6.6 FFNN-AE による異常度算出結果 (データセット 1)

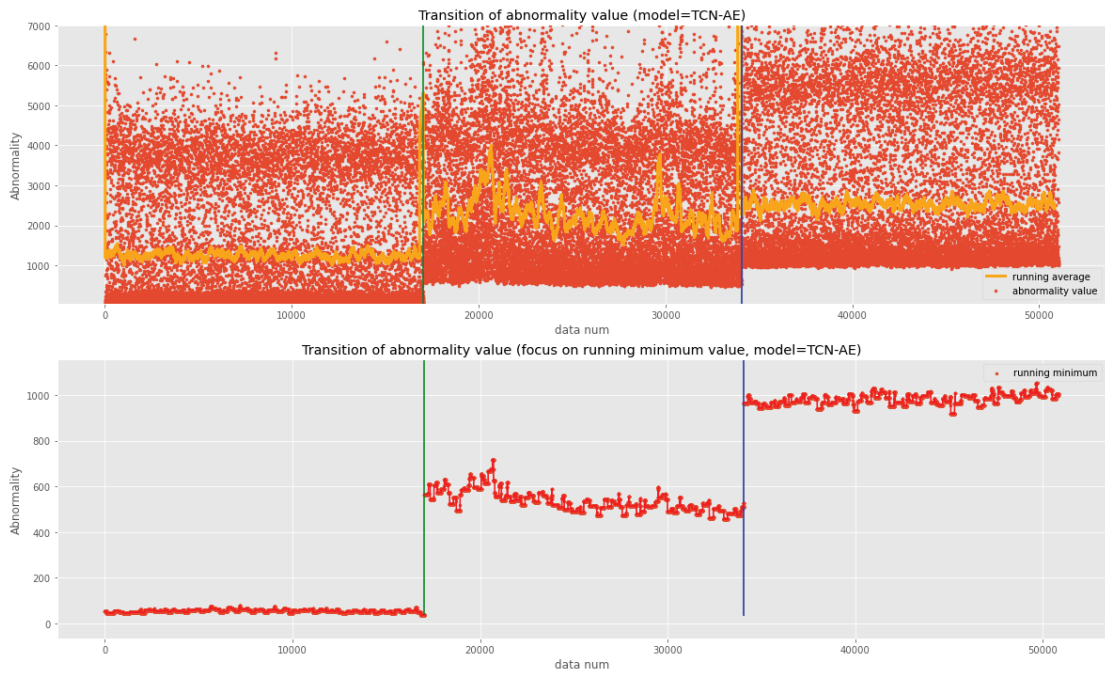


図 6.7 TCN-AE による異常度算出結果 (データセット 1)

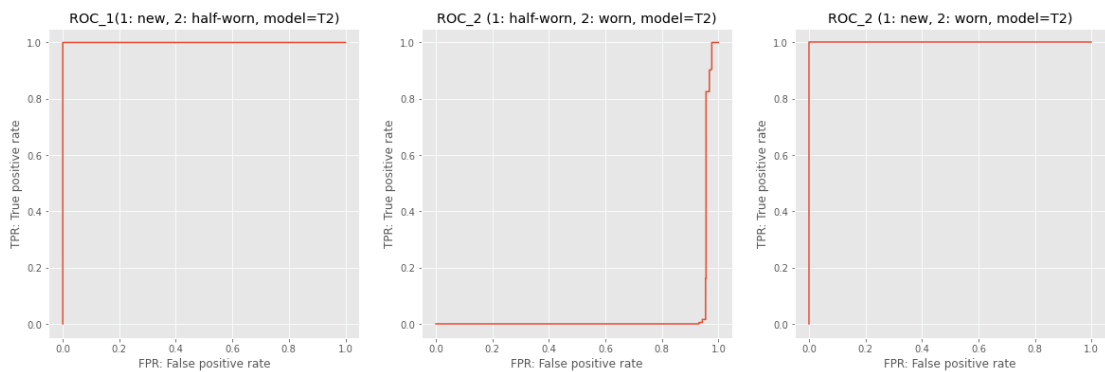


図 6.8 Hotelling's T-square 法による異常度の ROC 曲線 (データセット 1)

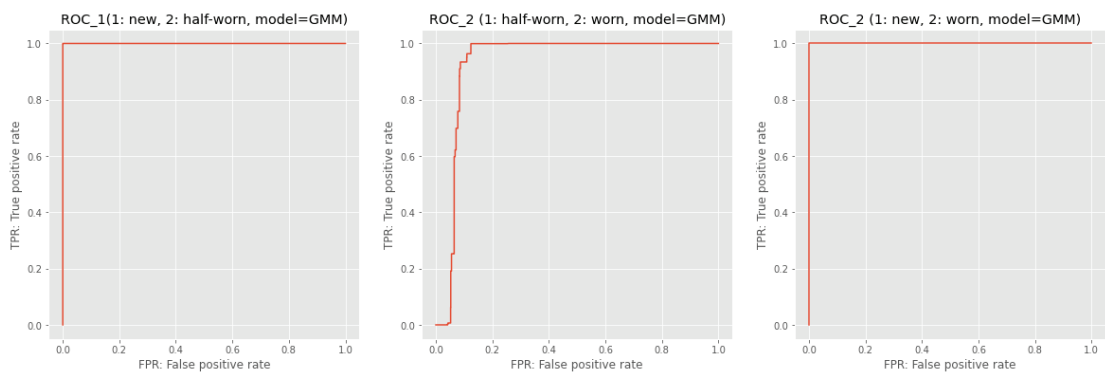


図 6.9 GMM による異常度の ROC 曲線 (データセット 1)

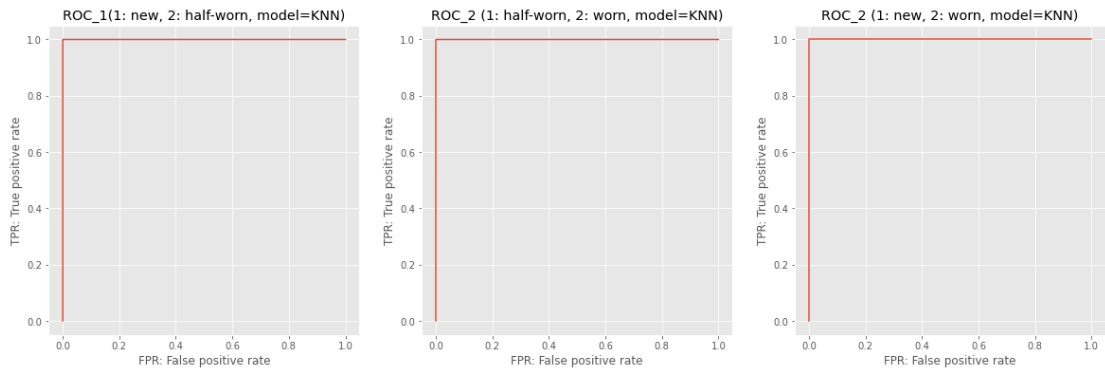


図 6.10 K-NN による異常度の ROC 曲線 (データセット 1)

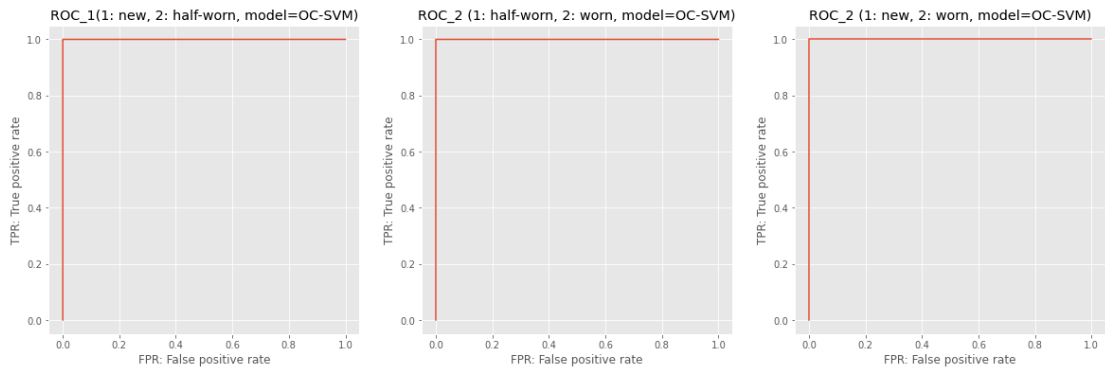


図 6.11 OC-SVM による異常度の ROC 曲線 (データセット 1)

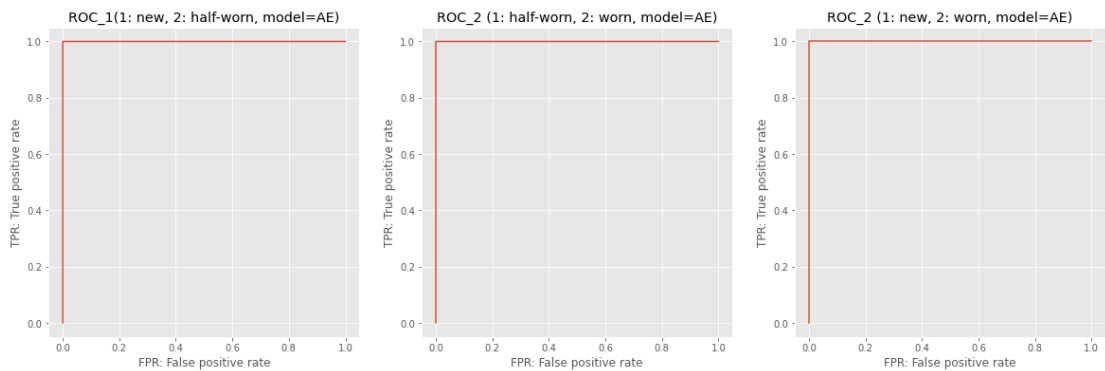


図 6.12 FFNN-AE による異常度の ROC 曲線 (データセット 1)

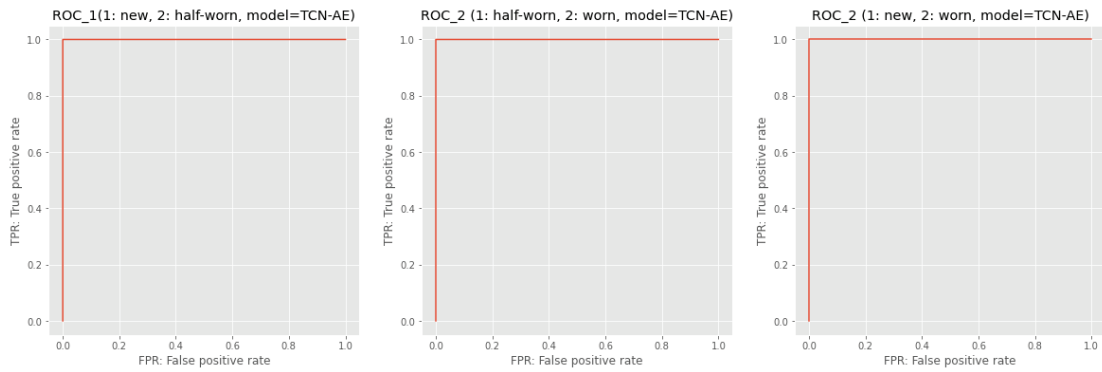


図 6.13 TCN-AE による異常度の ROC 曲線 (データセット 1)

表 6.8 6 種類の異常度算出手法による 3 種類の AUC と異常度算出時間 (データセット 1)

モデル	AUC_1	AUC_2	AUC_3	異常度算出時間 [sec]
Hotelling's T-square 法	1.0000	0.0415	0.9999	509.9673
GMM	1.0000	0.9298	1.0000	0.4710
K-NN	1.0000	1.0000	1.0000	450.8965
OC-SVM	1.0000	1.0000	1.0000	197.7401
FFNN-AE	0.9999	0.9999	1.0000	7.1962
TCN-AE	1.0000	1.0000	1.0000	13.1796

6.4 データセット 2 における実験結果

本節では、6 種類の手法によって算出されたデータセット 2 に対する異常度の推移のグラフと、3 種類の ROC 曲線、そして AUC と異常度算出時間の結果を示す。以降の図表は、6.3 節と同様の条件により作成した。はじめに、6 種類の手法による異常度推移のグラフを、図 6.14 から図 6.19 までに示す。ただし、グラフ中に 4 本の縦線を描画し、データを 5 個の区間に分割している。このうち、最も左側の区間が New 状態、中間の 3 区間が Half-Worn 状態、最も右側の区間が Worn 状態のデータである。続いて、6 種類の手法によって算出された異常度に対する、3 種類の ROC 曲線のグラフを、図 6.20 から図 6.25 までに示す。最後に、6 種類の手法によって算出された異常度に対する 3 種類の AUC と、異常度算出時間を表 6.9 に示す。



図 6.14 Hotelling's T-square 法による異常度算出結果 (データセット 2)



図 6.15 GMM による異常度算出結果 (データセット 2)

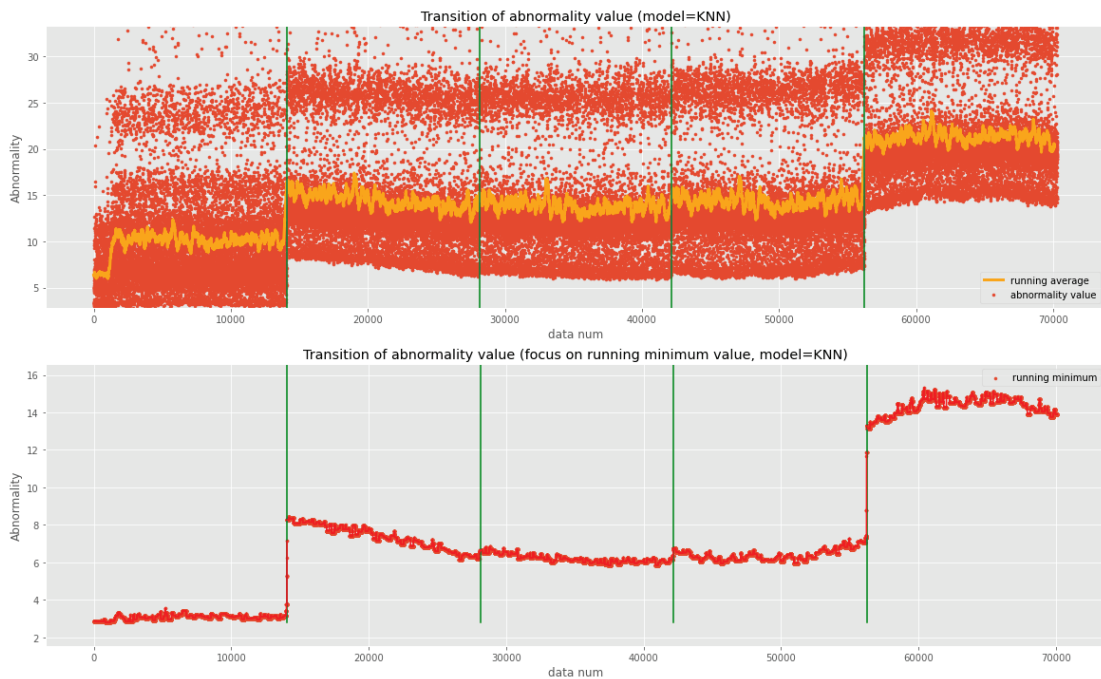


図 6.16 K-NN による異常度算出結果 (データセット 2)

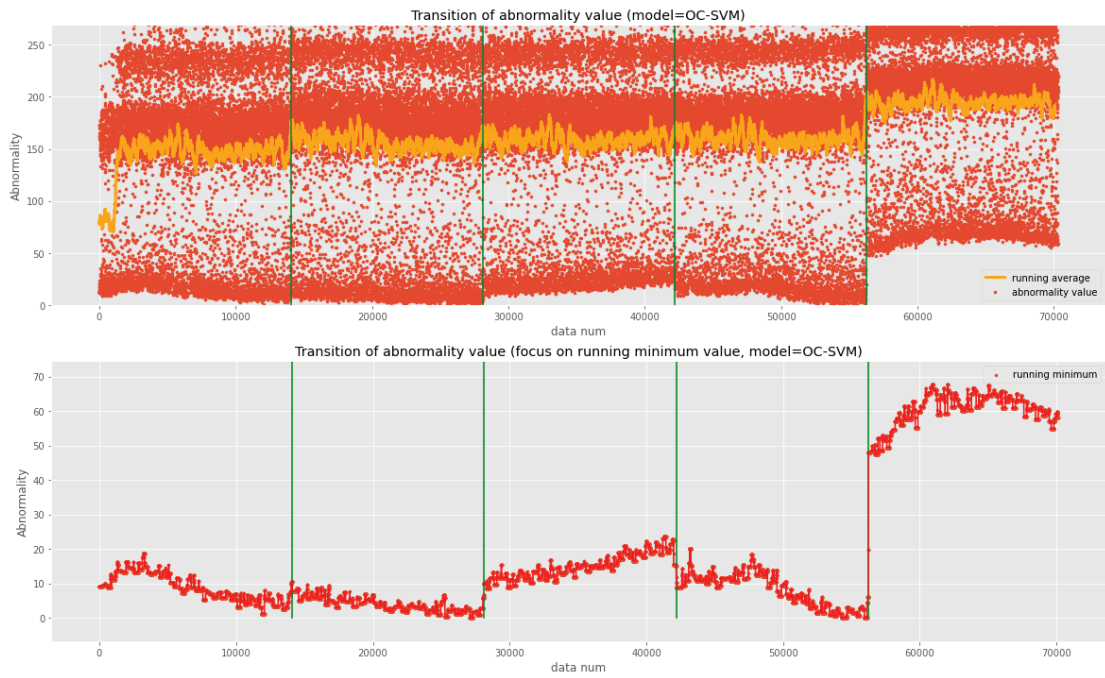


図 6.17 OC-SVM による異常度算出結果 (データセット 2)



図 6.18 FFNN-AE による異常度算出結果 (データセット 2)



図 6.19 TCN-AE による異常度算出結果 (データセット 2)

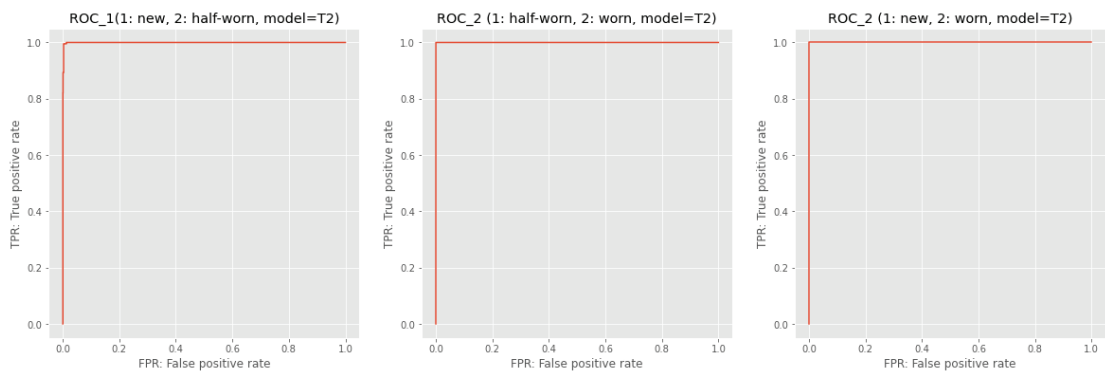


図 6.20 Hotelling's T-square 法による異常度の ROC 曲線 (データセット 2)

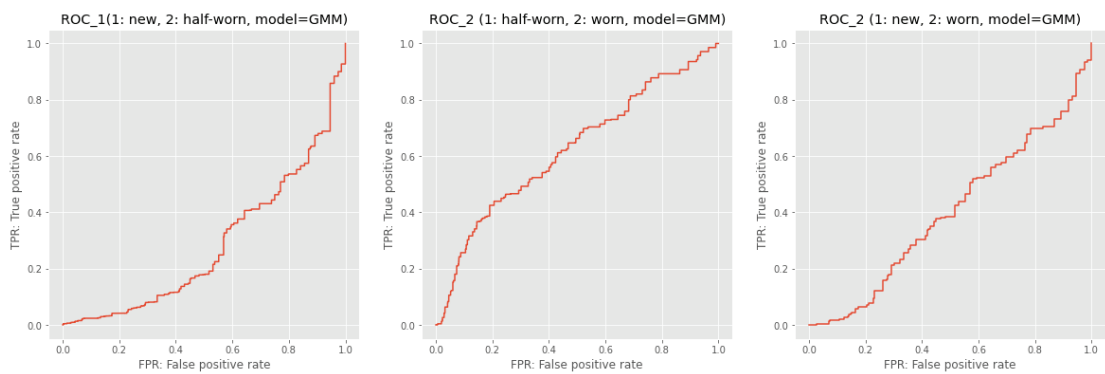


図 6.21 GMM による異常度の ROC 曲線 (データセット 2)

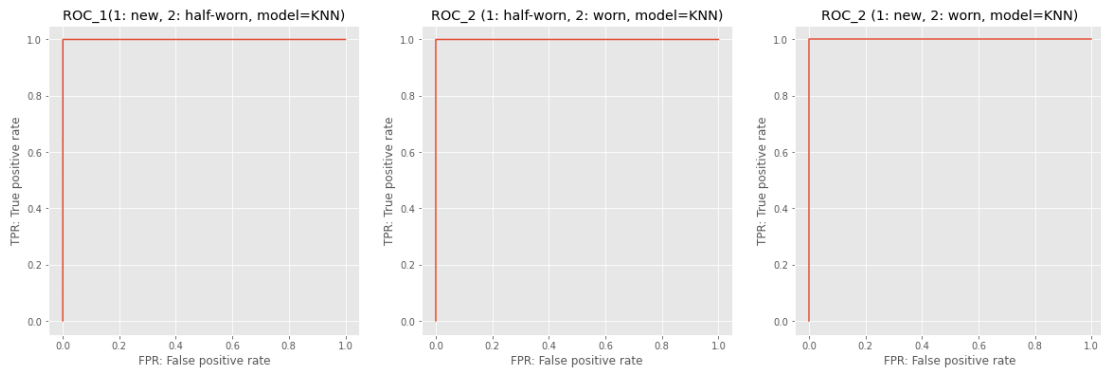


図 6.22 K-NN による異常度の ROC 曲線 (データセット 2)

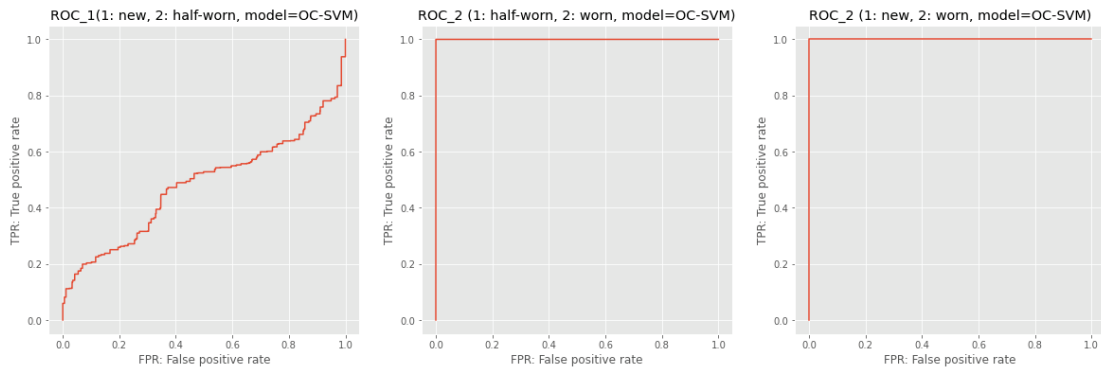


図 6.23 OC-SVM による異常度の ROC 曲線 (データセット 2)

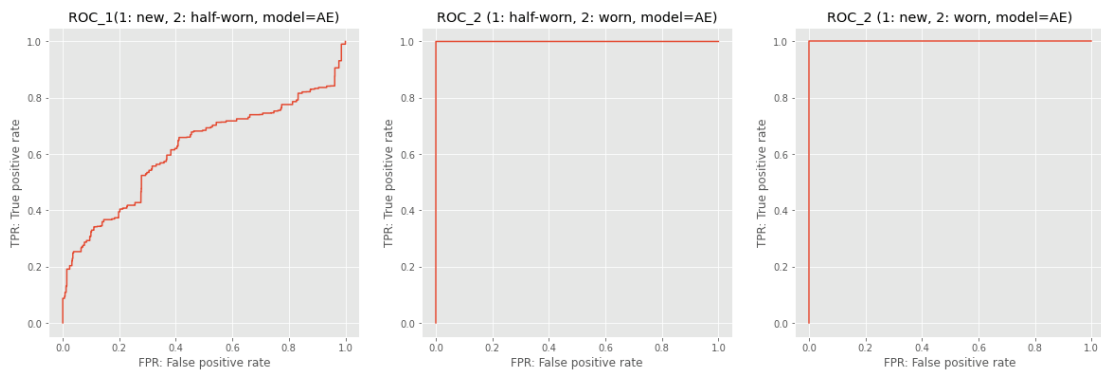


図 6.24 FFNN-AE による異常度の ROC 曲線 (データセット 2)

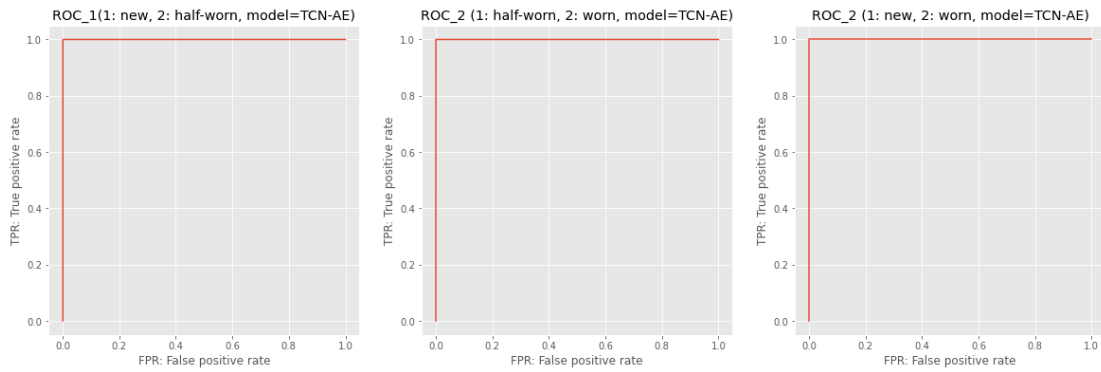


図 6.25 TCN-AE による異常度の ROC 曲線 (データセット 2)

表 6.9 6 種類の異常度算出手法による 3 種類の AUC と異常度算出時間 (データセット 2)

モデル	AUC_1	AUC_2	AUC_3	異常度算出時間 [sec]
Hotelling's T-square 法	0.9992	0.9998	1.0000	390.4237
GMM	0.2841	0.6225	0.4009	0.5556
K-NN	0.9997	1.0000	1.0000	381.3115
OC-SVM	0.4799	1.0000	1.0000	91.4443
FFNN-AE	0.6151	0.9999	1.0000	5.1946
TCN-AE	0.9996	1.0000	1.0000	80.7757

6.5 考察

はじめに、構築した AE ベースによる異常検知モデルと従来の機械学習アルゴリズムによる異常検知モデルとの性能を比較するため、表 6.8 および表 6.9 より、AUC および異常度算出時間の 2 種類の観点から考察を行う。AUC の結果について、データセット 1 において K-NN, OC-SVM および TCN-AE の 3 種類の手法の場合において、全ての AUC が 1 となり、3 状態を完全に分類することができた。一方で、データセット 2 においては Hotelling's T-square 法, K-NN および TCN-AE の 3 種類の手法の場合において、全ての AUC が 0.99 を超えており、3 状態をほぼ完全に分類することが可能であることが示された。また異常度算出時間について、AE ベースの異常検知モデルは、GMM 以外の機械学習アルゴリズムによる異常検知モデルよりも小さく抑えられていることが分かる。特に TCN-AE について、双方のデータセットにおいて AUC が同様の精度となった K-NN と比較すると、データセット 1 では異常度算出時間を 0.02922 倍、データセット 2 では 0.2705 倍に抑えられている。このことから、本データセットにおいて、TCN-AE により構築したモデルを用いた場合に、比較手法として機械学習ベースの異常度算出モデルよりも高速に、かつ精度を維持した状態で異常検知が可能であり、構築モデルが有効であることを示せた。

続いて、AE ベースによる異常検知モデル間においてその精度を比較した場合、FFNN-AE モデルよりも TCN-AE モデルの精度が高く、かつ異常度の分散が抑えられているという結果となった。これは、TCN-AE を用いた場合、周波数情報と時間情報の双方をより加味した学習が可能であるため、より実際の機械の状態に近い異常度算出が可能であることが考えられる。

最後に、本実験における評価指標として採用した異常度の単純移動最小値に関して考察を行う。異常度の推移のグラフの推移より、移動平均よりも移動最小値の方が、実際の刃先の摩耗状態である New, Half-worn そして Worn をより明確に識別できるような数値の推移を取ることが分かる。このことから、本データセットにおける指標として移動最小値を採用したことで、刃先の摩耗状態をより明確に捉えられることが示唆される。

第7章

実機械振動データによる故障予測

本章では、構築した手法による故障予測精度を検証するため、実際のチェーンコンベアより得られた振動データを用いて故障予測を行う実験について、そのデータおよび条件、提案手法における実験手法、条件、および結果を述べる。

7.1 使用するデータおよび理想の異常度推移

本章において故障予測を行う工場設備は、自動車部品工場にて製品の搬送に用いられるオーバーヘッドトロリー式のチェーンコンベアである。このコンベアの耐用年数は3年から5年とされている。このコンベアを駆動するモータ(図7.1)に3軸加速度センサを設置し、10分ごとに10秒間の振動データを自動的に取得した。図7.2に示している今回設置した加速度センサは、図7.1の左下にあるギア部の右手前部に設置した。チェーンコンベアの挙動に影響が無く、かつモータ振動以外の雑振動の影響を受けにくい場所に設置している。

7.1.1 データセット A

取得したデータのうち、実際に工場が稼働していた7か月分の振動データを、後述する故障予測手法の学習および検証実験に用いている。本章では、このデータセットを「データセット A」とよぶ。データセット A 収集期間中のイベントを図7.3のグラフのうち、4本の縦線で示している。この期間中、一度チェーンの噛合い不良に起因する異音が発生し、コンベアを停止したという事象が発生している(①)。本論文では、これ以降この事象を「故障」として扱う。故障発生後から、新しいチェーンと交換するまでの5週間後(②)までの期間は、劣化したチェーンを調整し、使用している。チェーンを新しいものに交換し、その2週間後に初期メンテナンスに伴うチェーンの初期伸びカットを行っている(③)。また、図7.3には、この期間中における理想的な異常度の推移を曲線で示している。図中の破線を閾値としたとき、異常度がこの閾値を故障が発生する前に設計することで、故障予測を行うことが期待できる。ただし、(①)から(②)の間中は、チェーンを交換せず劣化状態のチェーンを使用しているため、故障直前よりは低いものの正常状態よりも高い異常度を保持すると考えられる。

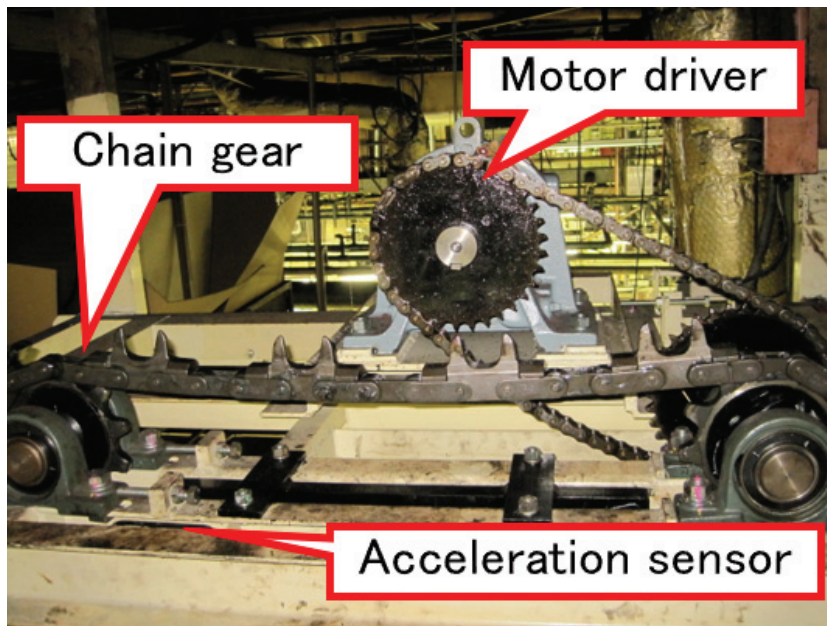


図 7.1 チェーンコンベアのモータ

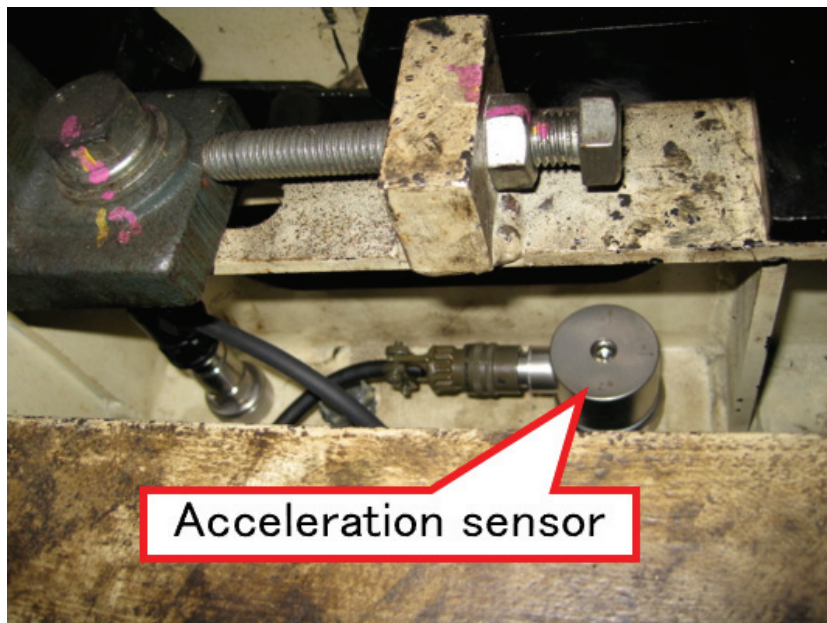


図 7.2 モータ部に設置した加速度センサ。センサはモータの左下，ギアの右側に位置している

表 7.1 にチェーンコンベアにおける振動データの取得条件を示す。本実験群では、本データのうちチェーンの初期伸びカット後から 2 週間分の区間 (図中③-④の区間, 500 ファイル) をモデル学習に用いる正常データ区間として扱う。また、本研究では、3 軸加速度センサより得られた 3 方向データのうち、もっともコンベアの挙動と振動の傾向に近い鉛直 1 方向のデータのみを用いている。

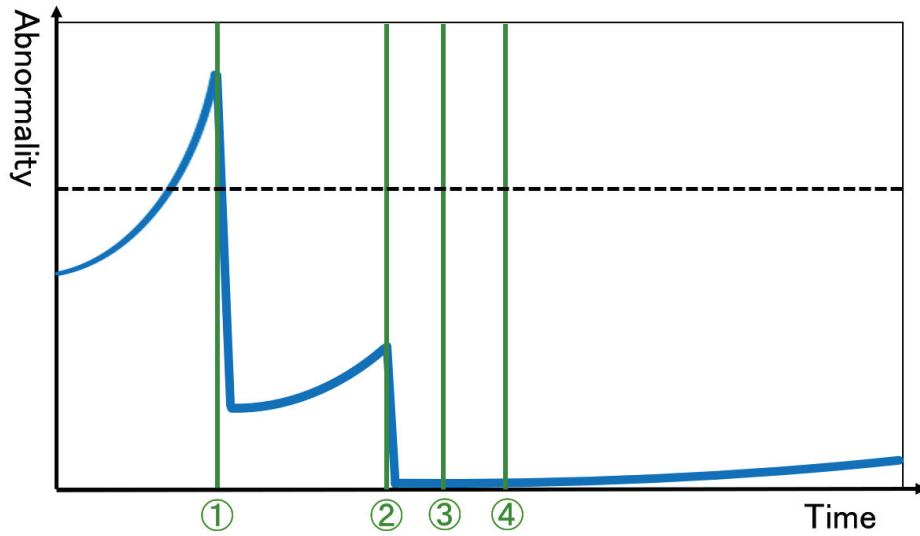


図 7.3 データセット A における，データ取得期間中の発生イベントと理想的な異常度の推移

表 7.1 データセット A 取得条件

条件	設定値
1 ファイル中のデータ長	10 秒
データの取得間隔	10 分
チェーンコンベアによる製品の搬送間隔	約 0.67 秒
ファイル数	7452 件
学習用データのファイル数	500 件
サンプリング周波数	12.8 kHz

7.1.2 データセット B

より長期的なデータにより，提案手法による劣化検知の有効性を検証するため，一部を除いて表 7.1 と同様の条件および同じチェーンコンベアにおいて，データセット A 収集開始から約 2 年間のデータを計測および取得した．本データセットを「データセット B」とする．ただし，本データセットにおいて計測したチェーンコンベアは，データセット A 中にも現れた故障以降，計測期間中に故障や不具合，劣化の兆候などの異常な挙動は報告されていない．このため，本データセットにおけるタスクは，より長期的なデータを用いた学習による劣化予測の精度の調査，および異常度の推移の調査である．データセット B の取得条件を表 7.1 に示す．データセット A と異なる点として，データの取得間隔が 10 分から 60 分となっている．

表 7.2 データセット B 取得条件

条件	設定値
1 ファイル中のデータ長	10 秒
データの取得間隔	60 分
ファイル数	3393 件
サンプリング周波数	12.8 kHz

7.2 実験 1: 手法 1 による故障予測の可能性および頑健性の検証

5.1 節において提案した手法 1 を用いて実際のモータに設置した振動データによる学習および異常度算出を行い，その結果から故障予測を事前に行うことが可能であるか，かつその故障予測に頑健性があるかを検証する実験を行った．

7.2.1 実験条件

本実験では，データセット A を用いる．本実験を行うにあたり，設定すべきパラメータは，振幅スペクトルの抽出条件，LSTM-AE における学習条件，GMM における条件，および閾値の設定方法である．表 7.3 に各条件を示す．また，本章における実験において学習および異常度算出に用いた機材およびフレームワークを表 7.4 に示す．また，本実験で用いる LSTM-AE の構造を 7.5 に示す．

表 7.3 実験 1 における, 提案手法に対する実験条件

条件	設定値
フレーム化におけるフレーム長	500 ms
フレーム化におけるフレームシフト	250 ms
総データ数 (ファイル数×ファイルあたりフレーム数)	290,628
学習用データ数	19,500
LSTM-AE の学習エポック数	500 エポック
LSTM-AE の最適化アルゴリズム	RMSprop
GMM の入力次元数	128 次元
GMM の出力次元数 (=異常度の次元数)	1 次元
異常度の算出単位	フレーム毎
異常度のグラフ表示単位	ファイル毎 (フレーム抽出した元データ毎の平均)
単純移動平均 (SMA) の算出単位	50 ファイル
グラフ中に表示する閾値	全体データの 90%・95% 点

表 7.4 実験に用いた機材およびフレームワーク

条件	設定値
使用 OS	Ubuntu 18.04.4 LTS
計算処理 GPU	GeForce GTX 1070 Ti
使用フレームワーク	Keras[42] (TensorFlow[41] backend)

表 7.5 実験 1 における LSTM-AE の構造

層名	時間ステップ数	出力サイズ
input	-	(320)
LSTM	10	(10,128)
RepeatVector	-	(10,128)
LSTM	10	(320)

本実験におけるフレーム長とフレームシフトは, 予備実験の結果 [9] をもって決定した. 本実験において, 異常度はフレーム毎に算出するが, 後述する実験結果では, フレーム抽出した元データ毎に平均を取ったデータを表示している. また, 本実験で使用する移動平均は単純移動平均 (Simple Moving Average; SMA) であり, 以下の式 (7.1) で算出される. ただし, t は移動平均を取る基準のデータの位置, $a(x^{(t)})$ は移動平均をとる異常度の基準のデータ, k は移動平均の対象となるデータ数である. 本

実験においては $k = 50$ ，すなわち基準から 50 データ先までのデータによる移動平均を算出した。これは，時刻に換算すると約 5 時間分のデータの平均を算出したことになる。

$$SMA(a(x^{(t)})) = \frac{\sum_t^{t+k} a(x^{(t)})}{k} \quad (7.1)$$

さらに本実験においては，閾値として全体の異常度中の 90% と 95% の位置の 2 種類を設けている。これは，2 種類の閾値レベルのうち，どちらであればあらかじめ故障が予測できると判断できるかを検証するためである。ここで，本実験において上記のように閾値を設定する理由として，使用する振動データの取得期間中に故障が発生しており，故障期間を含めた全異常度をあらかじめ算出できるためであることを補足しておく。

7.2.2 比較手法

本論文における提案手法のモデル構成は，LSTM-AE による特徴抽出，および GMM による異常度算出が重要な部分を占める。本手法と比較を行うための手法として，特徴抽出手法には PCA，異常度算出手法には Hotelling's T-Square 法を用いた。すなわち，特徴抽出手法には LSTM-AE と PCA を，異常度算出手法には GMM と Hotelling's T-square 法の各 2 種類を組み合わせ，合計 4 種類の手法による故障予測手法を実施した。今後，PCA と Hotelling's T-square 法を組み合わせた手法を比較手法 1，PCA と GMM を組み合わせた手法を比較手法 2，LSTM-AE と Hotelling's T-square 法を組み合わせた手法を比較手法 3 と記す。

ここで，比較手法に用いた PCA および Hotelling's T-square 法に対する条件は以下の表 7.6 の通りである。なお，LSTM-AE，GMM，移動平均および閾値設定の条件は，先述した表 7.3 の通りである。

PCA への入力データは，提案手法と条件を揃えるため，異常度を算出フレームとそれより前の 9 フレーム，合計 10 フレーム分の振幅スペクトル 320 次元を用いている。すなわち，PCA の入力次元数は 3200 次元となる。また，使用するデータにおいて PCA を実施したところ，3200 次元の入力に対し第 9 主成分において累積寄与率が 95% を超えるため，PCA の出力次元数を 9 次元としている。

表 7.6 Hotelling's T-square 法および PCA における設定条件

条件	設定値
PCA の入力次元数	3200 次元 (320 point × 10 フレーム)
PCA の出力次元数	9 次元
Hotelling's T-square 法の出力次元数	1 次元

7.2.3 実験結果

実験結果は，2 種類のグラフにより表示する。1 種類目は横軸を時間，縦軸をデータごとの異常度とした散布図，2 種類目は横軸を時間，縦軸を異常度の移動平均とした線グラフである。本章では，前者

のグラフをグラフ (a), 後者をグラフ (b) と記す. グラフ (a) について, 1 個の点が 10 秒分のデータ 1 件の異常度の平均に相当する. 赤い箇所が学習データの期間である. また, グラフ中には 4 本の縦線が引かれているが, これは左側より故障 (①), チェーン交換 (②), チェーンの初期伸びカット (③), データの変化 (④) がそれぞれ発生した日付に相当する. ここで, 実際のチェーンコンベアの挙動について, チェーンの初期伸びカット後は正常に作動していた. 一方で, 故障前については, 故障が近づくにつれて挙動が正常時と異なるものになっていたほか, 故障後から初期伸びカット前までの期間も, 正常な状態とは挙動が若干異なっていた. また, 2 本の横線は, 点線が 90% の閾値, 実線が 95% の閾値をそれぞれ示している.

提案手法 1 による 2 種類のグラフを図 7.4 に示す. また, 比較手法 1,2,3 による 2 種類のグラフを, 順に図 7.5, 図 7.6, 図 7.7 に示す. さらに, 各手法における異常度算出結果において, チェーン交換の準備に要する 2 週間を基準として, 故障 2 週間前から実際の故障が起こるまでの期間とその前の期間とを比較して異常度がどの程度上昇するか, 手法間の比較を行った. 同時間幅である故障 4 週間前-故障 2 週間前 (Period 1) と, 故障 2 週間前-故障直前 (Period 2) の 2 区間における異常度の中央値を算出した結果, および 2 区間の中央値の比 (Period 2/Period 1) の比較を図 7.8 に示す. ただし, 図 7.4 から図 7.7 においては, 突発的な異常度の上昇がどの程度現れるかも含めた異常度の推移を確認することが目的であるため, 突発的な異常度の推移も計算に含まれる移動平均を使用した. 一方で, 図 7.8 では期間中の異常度がどの程度増加するかを確認することが目的であり, 平均値により算出した場合は外れ値の影響を強く受けるため, その影響を軽減できる中央値を使用した. また, 算出した異常度は平均 0, 分散 1 となるように正規化を行ったあと, 最小値が 0 になるように調整している. 学習データについては, 全データのうち 500 ファイル分を使用して学習したが, これは学習データ量としては十分確保されており, 異常度の推移に支障をきたすような過学習は起きなかった.

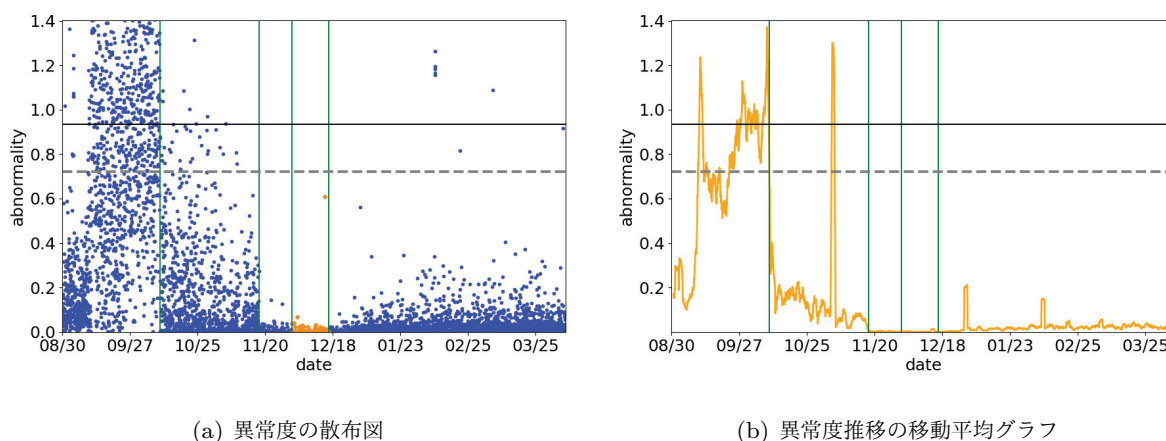


図 7.4 提案手法 1 による異常度の散布図と移動平均グラフ. 異常度の数値の 90% 点は 0.7225, 95% 点は 0.9351 となり, 異常度の最大値は 4.0766 であった. 本グラフ, および比較手法のグラフより, 提案手法 1, および 3 種類の比較手法すべてにおいて, 故障を前もって予測することが出来ていることが示された.

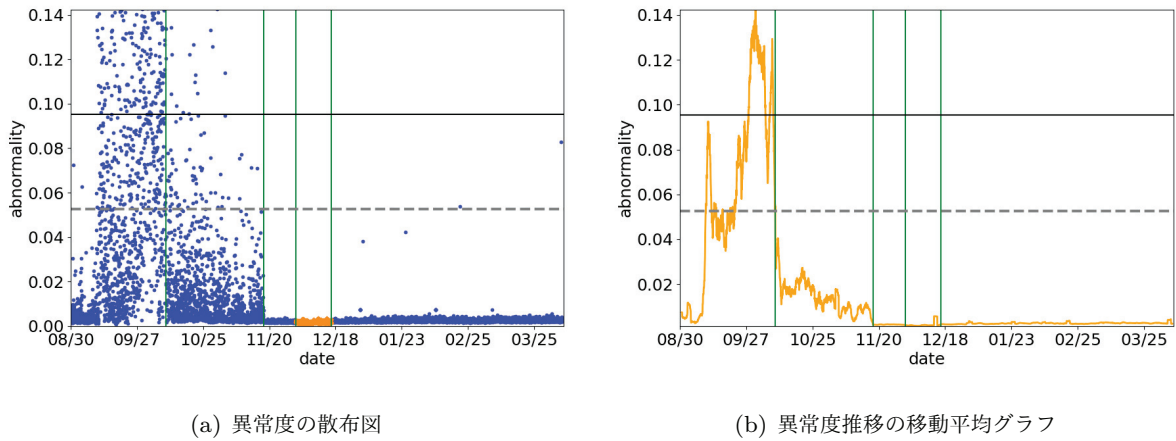


図 7.5 比較手法 1 (PCA+Hotelling's T-square 法) による異常度の散布図と移動平均グラフ. 異常度の数値の 90% 点は 0.0526, 95% 点は 0.0954 となり, 異常度の最大値は 0.4388 であった.

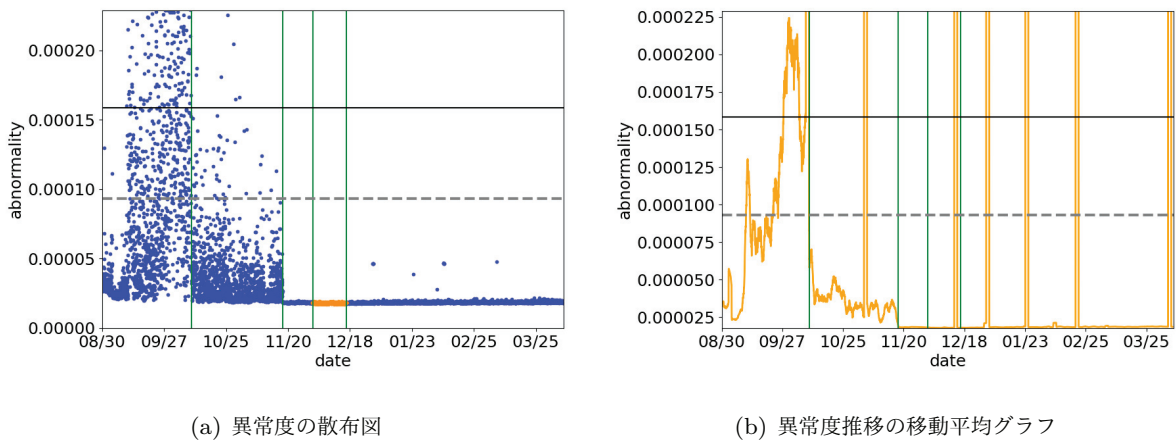


図 7.6 比較手法 2 (PCA+GMM) による異常度の散布図と移動平均グラフ. 異常度の数値の 90% 点は 0.0001, 95% 点は 0.0002 となり, 異常度の最大値は 0.1731 であった.

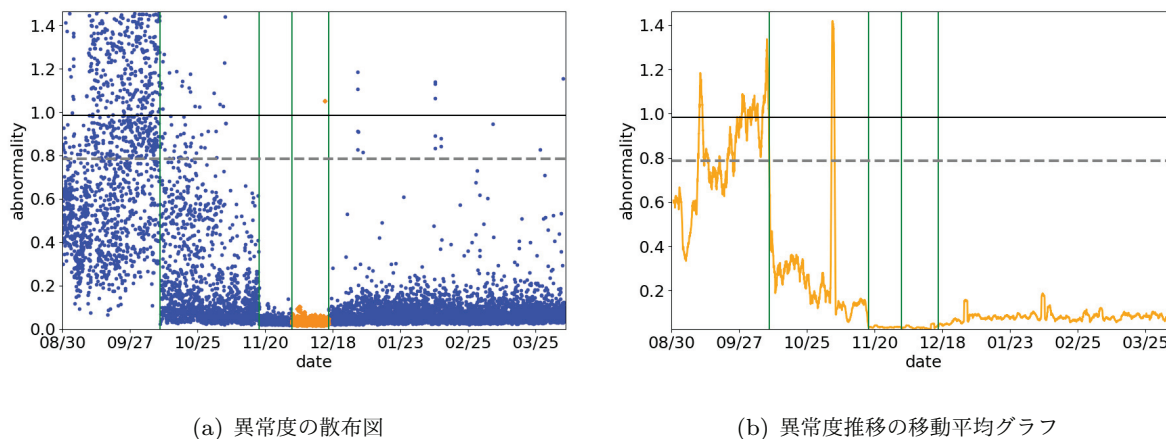


図 7.7 比較手法 3 (LSTM-AE+Hotelling’s T-square 法) による異常度の散布図と移動平均グラフ。異常度の数値の 90% 点は 0.7858, 95% 点は 0.9836 となり, 異常度の最大値は 4.3055 であった。

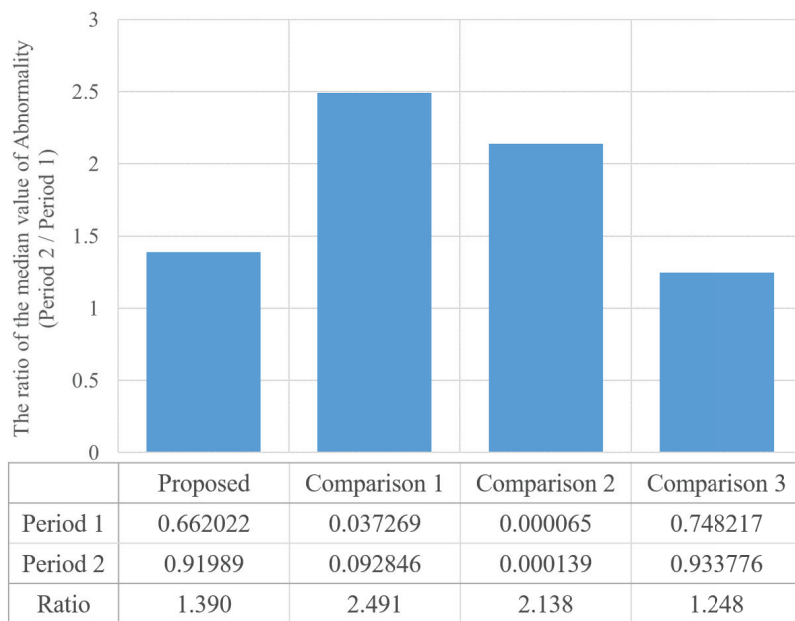


図 7.8 故障 4 週間前-故障 2 週間前 (Period 1) と, 故障 2 週間前-故障直前 (Period 2) の 2 区間における異常度の中央値, および 2 区間の中央値の比 (Period 2/Period 1). 表より, PCA による特徴抽出を行った場合, LSTM-AE の場合と比較して故障直前の異常度の上昇幅が大きいことが分かった。

7.2.4 考察

8 種類のグラフより, 提案手法および 3 種類の比較手法のすべての手法において, 計測開始直後から時間に比例して異常度が上昇し, 故障が発生する 3 週間程度前には異常度の推移が 90% の閾値を上回

る結果が得られた。チェーン交換の準備に要する期間である 2 週間より前には異常度が高い値を維持することから、どの手法を用いた場合においても、劣化に伴う故障をあらかじめ予測することができるような異常度の推移となることを示せた。一方で、比較手法 2 においては正常状態に含まれる、データの変化後の期間中に複数回突発的に閾値を大きく上回っている箇所が現れた。また、故障後からチェーン交換までの期間中に、比較手法 1 以外の 3 種類の手法において、突発的に閾値を大きく上回る箇所が確認できた。この現象は、データを収集する加速度センサ自体の変化に起因する外れ値であると考えられ、突発的な異常度上昇を抑える処理を加えることにより解決できる。しかし、突発的な異常度上昇の頻度が増加した場合に、頑健な故障予測に影響を及ぼす一要因となりうることが考えられる。

続いて、それぞれの故障予測手法の結果の比較を行い、その頑健性に関する詳細な考察を行う。

はじめに、異常度算出を行う手法である Hotelling's T-square 法と GMM に対する異常度の推移について比較する。特徴抽出手法が同じである場合、2 種類の異常度算出手法において異常度の移動平均のグラフより大きく差がみられるような箇所はなかったが、先述した通り異常度の突発的な上昇が GMM による手法では現れた一方、Hotelling's T-square 法ではこの傾向をある程度抑えられた。しかしながら特徴抽出を LSTM-AE で行った場合において、突発的な異常度のピークではなく、時間変化に伴う異常度の推移を見た場合には、Hotelling's T-square 法の場合には区間の変更直後から異常度が上昇し、全体的に 0.1 付近まで上昇しているのに対し、GMM の場合には区間の変更直後における異常度の上昇は起きず、データ集計終了時点での異常度も小さい値を維持していた。このことから、突発的な異常度の上昇を抑えながら、正常な期間におけるデータと故障前におけるデータの間の異常度の推移に差が出る、より頑健な故障予測を行うためには、GMM と Hotelling's T-square 法を統合した手法を考案する必要があると考えられる。

次に、特徴抽出手法である LSTM-AE と PCA に対する異常度の推移について比較する。先述した通り、異常度の推移については、LSTM-AE と PCA 双方とも同様であることがわかった。一方で、異常度より算出された閾値を比較すると、PCA の場合は 90% の閾値と 95% の閾値の幅が、0% から 90% までの幅とほぼ同程度であった。故障前の推移についても、LSTM-AE の場合は一部大きく下がる箇所があるものの徐々に異常度が 90% の閾値を上回り、故障 2 週間前-故障区間では 95% の閾値付近を前後するように推移しているのに対し、PCA の場合は急激に 95% の閾値を上回るように上昇している傾向が見られている。また、図 7.8 により各手法間における故障前の区間の異常度の変化の割合を比較したところ、LSTM-AE で特徴を抽出した提案手法では 1.3895、比較手法 3 では 1.2480 と、異常度が徐々に上昇していたことを示したのに対し、PCA で特徴を抽出した比較手法 1 では 2.4912、比較手法 2 では 2.1385 となり、異常度が 2 倍以上に上昇していることが分かった。この急激な異常度の上昇により、正常時や故障 2 週間前の異常度と故障直前の異常度において大きく差が生じたことが、PCA の場合において 90% の閾値から 95% の閾値と 0% から 90% の閾値の幅がほぼ同程度となった原因である。これらのことから、PCA により特徴抽出した異常度算出結果は、故障が発生する 2 週間前から故障直前までの異常度とそれ以前を比較した場合、LSTM-AE による特徴抽出よりも大きく変化していた、といえる。

本研究においては、図 7.3 で示すような異常度の推移、すなわち故障前においても異常度が徐々に

増加し、故障が発生する一定期間前から異常度が閾値を上回るような推移となる異常度算出モデルが、チェーンコンベアの劣化を予測する上では望ましい推移である。このため、LSTM-AE による異常度算出結果は、PCA よりも理想的な異常度の推移に近いものであったと考えられる。

このような異常度の推移が得られた理由として、LSTM-AE が周波数情報と時間情報を効率的に学習し、かつ非線形な次元圧縮な特徴抽出ができるため、線形的な次元圧縮手法である PCA よりも表現力の高い特徴量の抽出に成功しており、よりチェーンコンベアの状態を反映した異常度の算出が行えていると考えられる。このことから、故障を直前に発見することを想定した場合には LSTM-AE と PCA 双方有効であるが、1 章で定義したような「頑健な」故障予測を行う場合には、閾値の幅や推移の上下幅を見ると LSTM-AE による特徴抽出が有効であるといえる。

7.3 実験 2: TCN-Autoencoder に用いるパラメータの調査

提案手法において、AE の構造を決定するパラメータのうち、入力データ長はモデルがどれだけの期間のデータを学習の際に加味するかを表すモデルの受容野の幅となる。また積層数やフィルタ数は、入力データの持つ変動パターンに対して、どれだけ学習するかに相当するモデルの表現力に関わる。実験 2 では、この 2 種類のパラメータを変化させ、TCN-AE により異常度を算出し、異常度の推移やその安定性、およびパラメータ数や学習、推論速度から、最適なパラメータについて考察する。

7.3.1 実験条件

本実験では、実験 1 と同様、データセット A を用いる。変化させるパラメータの条件について、以下の表 7.7 に示す。ただし、変化させないパラメータについては、[39] と同等のパラメータを使用する。入力データのパラメータ削減のため、サンプリング周波数を 10 分の 1 にダウンサンプリングしている。このファイルから AE への入力としてデータを用いる場合には、入力データ長に対応するサンプリング点数分のデータを、1 点ずつずらして順番に抽出していく。例えばデータ長を 800ms とした場合、サンプリング周波数が 1,280Hz となるのでサンプリング点数は 1,024 点となる。この 1,024 点を、各ファイルから順に抽出していく。すなわち、1 番目のデータは 1 点目から 1,024 点目まで、2 番目のデータは 2 点目から 1,025 点目まで、3 番目のデータは 3 点目から 1,026 点目まで、といったようにデータを抽出する。この処理により、より多くのデータをファイルから取り出すことが可能であり、大きなモデルの学習にも十分対応できるデータ数を確保できる。また、encoder/decoder の積層数・フィルタ数の組み合わせは、対応する順に括弧内に表記している。例えば、表のように、encoder 層が 2 層で構成され、フィルタ数が 1 層目は 8、2 層目は 16 となる構造を持つ場合、 $F=(8,16)$ のように表記している。異常度算出結果のグラフについて、データごとに算出した異常度に対し移動平均を取ったものである。また、グラフ中に水平方向に引かれた点線は、異常度の移動平均値全体の 95% の地点を示している。

表 7.7 実験 2 における AE の設定条件

条件	設定値
ダウンサンプリング後のサンプリング周波数	1,280 Hz
入力データ長	100/200/400/800 (ms)
encoder/decoder の積層数・フィルタ数	(8)/(16)/(32)/(8,8)/(8,16)/(8,8,8)/(8,8,8,8)
TCN 部における活性化関数	Normalized ReLU
AE の出力における活性化関数	Linear
誤差関数	Mean Squared Error
最適化アルゴリズム	Stochastic Gradient Decent
学習率	0.01
移動平均を行うデータ幅	3,000

表 7.8 F=(8,16) における TCN-AE の構造

層名	フィルタサイズ	出力サイズ
input		(320,1)
Convolution 1D	8	(320,8)
Max Pooling 1D	-	(160,8)
Convolution 1D	16	(160,16)
Max Pooling 1D	-	(80,16)
Up Sampling 1D	-	(160,16)
Convolution 1D	8	(160,8)
Up Sampling 1D	-	(160,8)
Convolution 1D	1	(320,1)

7.3.2 実験結果・考察

設定パラメータ条件における異常度算出結果のうち、積層数 1, フィルタ数 (8), 入力データ長 100 ms の場合を図 7.9 に, 入力データ長 800 ms の場合を図 7.10 に, フィルタ数 (16), 入力データ長 800 ms の場合を図 7.11 に, 積層数 4, フィルタ数 (8,8,8,8), 入力データ長 800 ms の場合を図 7.12 に示す. また, 表 7.9 に各パラメータにおける 1 データあたりの異常度算出速度およびパラメータ数を示す.

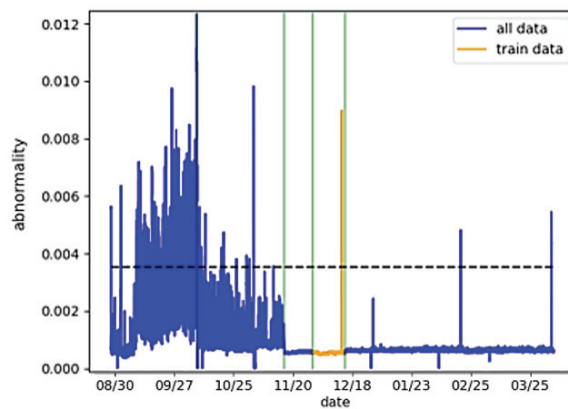


図 7.9 入力データ長 100 ms, 積層数 1, フィルタ数 (8) における異常度の推移

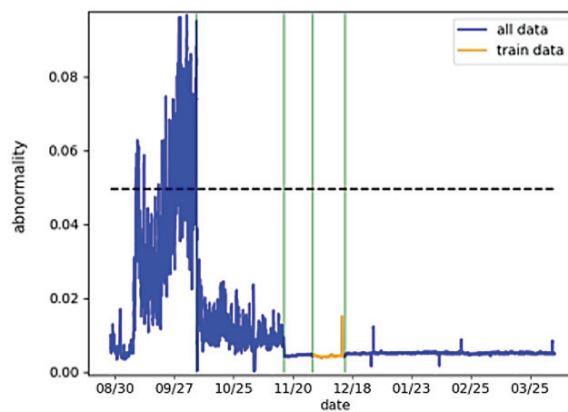


図 7.10 入力データ長 800 ms, 積層数 1, フィルタ数 (8) における異常度の推移

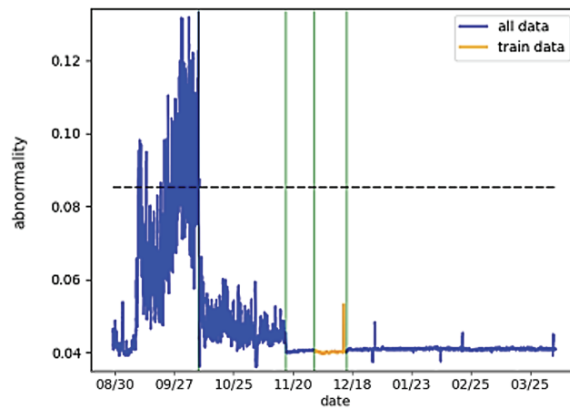


図 7.11 入力データ長 800 ms, 積層数 1, フィルタ数 (16) における異常度の推移

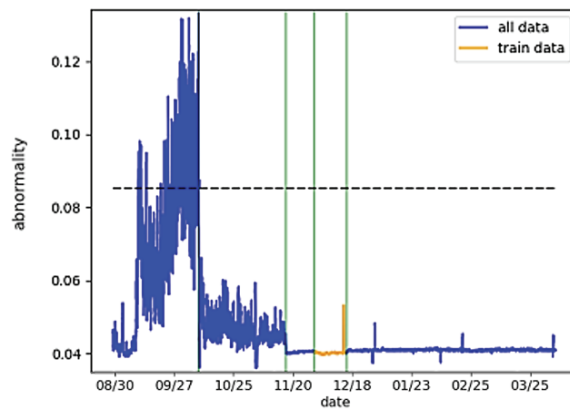


図 7.12 入力データ長 800 ms, 積層数 4, フィルタ数 (8,8,8,8) における異常度の推移

表 7.9 積層数・フィルタ数毎のパラメータ数

積層数・フィルタ数	(8)	(16)	(32)	(8,8)	(8,16)	(8,8,8)	(8,8,8,8)
パラメータ数	9,241	34,865	135,265	25,641	66,617	42,041	58,441
異常度算出速度/データ (μs)	6.593	6.784	8.439	8.340	8.664	9.451	10.733

この結果について、本節では積層数・フィルタ数の観点から考察を行う。図 7.10, 図 7.11, 図 7.12 に示す通り、異常度の算出結果に対して、積層数・フィルタ数による有意な差は見られず、おおむね同様の値となった。一方、表 7.9 より、パラメータ数や計算時間のより少ないパラメータの組み合わせは、積層数 1, フィルタ数 (8) の場合と積層数 2, (8,8) の 2 種類であり、後述する実験 3 ではこれらのパラメータを使用することとした。

入力データ長に関するパラメータの調査

図 7.9 と図 7.10 を見ても明らかなように，入力データ長が長ければ長いほど故障前後における異常度の差が大きくなり，またチェーン交換後の期間において発生する突発的な異常度の上昇も抑えられる傾向がみられた．これは，モデルに入力する情報が増えるほどモデルはより正確に正常状態における分布を学習でき，突発的に起こる異常も吸収できることが考えられる．そこで，入力データ長を更に延長した場合における異常度の推移や学習速度について比較を行った．実験条件を表 7.10 に示す．ただし，入力データ長および積層数・フィルタ数以外の条件は表 7.10 と同様である．異常度算出を行っ

表 7.10 AE の設定条件

条件	設定値
入力データ長	400/800/1,200/1,600 (ms)
encoder/decoder の 積層数・フィルタ数	(8)/(8,8)

表 7.11 入力データ長毎の異常度算出速度

積層数・フィルタ数	(8)				(8,8)			
入力データ長 (ms)	400	800	1,200	1,600	400	800	1,200	1,600
異常度算出速度/データ (μs)	4.902	6.784	7.843	10.407	6.326	8.340	10.111	11.786

た結果，異常度の推移については，入力データ長が 800 ms より長くなった場合，800 ms における推移とほぼ同じであることが分かった．今回用いたチェーンコンベアによる製品の搬送間隔が 670 ms 程度であることから，入力データ長が 800 ms 以上の場合に，異常度の算出に十分な情報をモデルが獲得できたと考えられる．なお，この搬送間隔については，データ解析を行うことで発見したパラメータであり，機械に関する事前知識が無くても獲得できる知見である．一方で，表 7.11 の通り，入力データ長を延長するにつれて学習および異常度算出に要する時間は長くなった．これらの結果および考察を踏まえて，後述する実験 3 においては，入力データ長は 800 ms とすることとした．

7.4 実験 3: 異常度算出手法間における特性の調査と精度比較

本節では、実験 2 において求められた 2 種類のパラメータを用いて、3 種類の AE により異常度算出を行い、構造による異常度の推移に対する特徴や、算出速度とパラメータ数などの特性の違いを調査する実験とその結果を説明する。

7.4.1 実験条件

本実験に用いるデータは、実験 1, 2 と同様にデータセット A である。本実験における 3 種類の AE の設定パラメータは、表 7.7 と同様のものである。ただし、先述の通りモデルの入力データ長には 800 ms を、TCN-AE の積層数・フィルタ数、および LSTM-AE の積層数・LSTM ブロック数には積層数 1, ブロック数 (8) と積層数 2, ブロック数 (8,8) の 2 種類のパラメータを使用した。また、FFNN-AE の積層数・ノード数については、いずれも積層数 2 で、ノード数は (512,128) と (256,128) の 2 種類パラメータを使用した。また、データの入力方法については、FFNN-AE の場合は始端から終端までの全ての時刻分のデータを同時に入力し、LSTM-AE と TCN-AE の場合は各時刻につき 1 点ずつ入力する。加えて、各モデルの性能評価のため、異常度算出時の速度を比較した。

7.4.2 実験結果

異常度算出結果のうち、LSTM-AE における積層数が 2、ブロック数が (8,8) である場合の結果を図 7.13 に、FFNN-AE における積層数が 2、ノード数が (256,128) の場合の結果を図 7.14 に示す。また、表 7.12 に、故障前 2 週間から故障 (図 7.3 中①) まで (区間 1)、故障後からチェーン交換 (②) まで (区間 2)、モデルの学習に用いた区間 (区間 3, ③-④)、学習に用いた区間以降の区間 (区間 4, ④以降) の 4 区間における異常度の平均を示す。最後に、各モデル、および各モデルにおける積層数・ノード数・ブロック数・フィルタ数における 1 データあたりの異常度算出速度、およびパラメータ数を表 7.13 に示す。

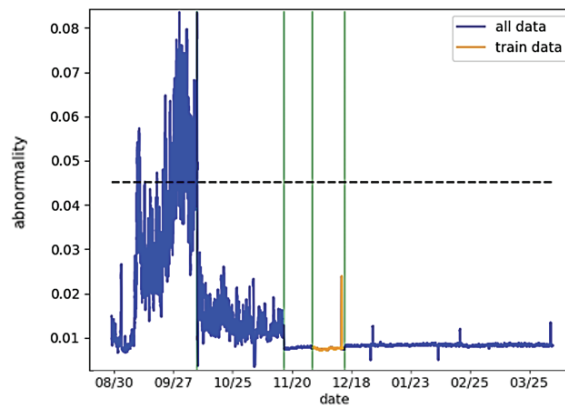


図 7.13 LSTM-AE, 積層数 2, ブロック数 (8,8) における異常度の推移

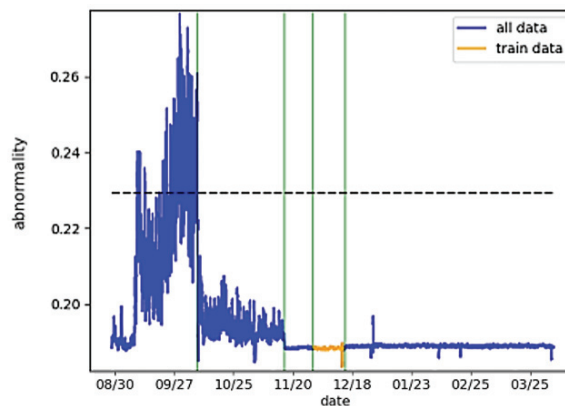


図 7.14 FFNN-AE, 積層数 2, ノード数 (256,128) における異常度の推移

表 7.12 各区間における異常度の平均, および故障前後の異常度の比. ただし, 故障前後の異常度の比 = (区間 1-区間 3)/(区間 2-区間 3)

モデル	区間 1	区間 2	区間 3	区間 4	異常度比
FFNN-AE	0.7789	0.7147	0.7055	0.7057	7.954
LSTM-AE	0.0313	0.0137	0.0077	0.0082	3.915
TCN-AE	0.0351	0.0133	0.0067	0.0073	4.278

表 7.13 モデル毎, および積層数・ノード数・ブロック数・フィルタ数毎のパラメータ数および 1 データあたりの異常度算出速度

モデル	FFNN-AE		LSTM-AE		TCN-AE	
積層数・ノード数など	(512,256)	(256,128)	(8)	(8,8)	(8)	(8,8)
パラメータ数	1,050,112	985,088	1,737	3,913	9,241	25,641
異常度算出速度/データ (μs)	3.129	3.050	272.207	543.781	6.593	8.340

7.4.3 考察

モデルにより算出された異常度の推移について、大きな挙動の差は見られなかった。ただし、表 7.12 より、故障前後の異常度の比は、FFNN-AE の場合が 7.954 なのに対し、TCN-AE が 4.278、LSTM-AE は 3.915 となり、故障後の異常度の下降の割合に差が生じた。LSTM-AE の持つゲート構造や、TCN-AE の持つ時間的畳み込み構造により、突発的な値変化への感度が FFNN-AE よりも高いことが要因であると考えられる。このことから、LSTM-AE および TCN-AE により非定常な突発的異常を捉えることで、劣化の進行と外れ値の出現を同時に検出することが期待できる。

続いて、パラメータ数および異常度算出速度の推移の観点から考察を行う。表 7.13 の通り、パラメータ数は LSTM-AE が最小、FFNN-AE が最大となった。異常度算出時の速度は、LSTM-AE が最大、FFNN-AE が最小となり、パラメータ数が 5.320 倍多い TCN-AE 構造と比較しても 41.28 倍の時間を要した。パラメータが FFNN-AE において増大する理由は、LSTM-AE や TCN-AE の場合、入力データの持つ時刻情報に基づいて順番に入力されていくのに対し、FFNN-AE の場合にはデータ長分のすべてのデータを同時に入力する必要があることが考えられる。一方で、データを系列情報として扱う LSTM-AE が、TCN-AE よりも異常度算出に要する時間が増大する理由は、TCN-AE が複数の時刻分で一度に畳み込みを行う構造を持つのに対し、LSTM-AE は時刻順に系列データを再帰する構造を取っていることが考えられる。

以上の考察に加え、LSTM-AE および TCN-AE は系列データのみを使用できるのに対し、FFNN-AE は時系列以外のデータにも使用できることも踏まえると、FFNN-AE は、学習データが十分に確保できる場合には高い異常度算出性能を発揮することが期待でき、データの形態に対しても柔軟に対応した異常度算出器となることが期待できる。一方で、モニタリング初期段階や、新品から劣化までのサイクルが短い設備など、獲得できるデータのサンプル数が限定される場合には、FFNN-AE の学習に必要なデータが確保できない、ということが考えられる。そういった場合には、よりパラメータの少ない TCN-AE および LSTM-AE が有効であると考えられる。特に、TCN-AE については、少ないパラメータでありながら、FFNN-AE のような大きな表現力を有するモデルと同様の異常度の推移を出力できている。

FFNN、および LSTM をはじめとした RNN 構造において、モデルの性能はパラメータのチューニングによって大きく左右されるのに対し、TCN は比較的単純なチューニングで高い性能を発揮できる [43] ことも考慮すると、TCN-AE はデータサンプルが限られる場合や、メモリや時間に制約があり大きなサイズのモデルを使用できない状況下においては、TCN-AE が最も性能の高い異常度算出器となることが期待できる。

以上の結果および考察を、表 7.14 にまとめる。本表は、振動データをはじめとした、観測データに基づく故障予測を実施する場合に、どのようなモデル・入力情報を用いることでより有効な異常度算出が可能となるかを示すものといえる。

表 7.14 3 種類の AE ベース異常度算出手法の比較

モデル	利点	欠点
FFNN-AE	<ul style="list-style-type: none"> ● 振動・音声・画像データなど、様々なデータ形式に対応しているため、汎用性が高い ● 十分なデータが揃っている場合、高い異常度算出性能を発揮する 	<ul style="list-style-type: none"> ● モデルのパラメータ数が増大する ● 複雑な特徴表現の獲得のため積層数・ノード数を増加する必要がある
LSTM-AE	<ul style="list-style-type: none"> ● 再帰構造を持つため、積層数・ノード数を増加してもパラメータ数が少なく、小さいモデルが構築可能 	<ul style="list-style-type: none"> ● 入力データ長が長い場合、再帰構造により学習および異常度算出に要する時間が大幅に増大する ● 最適な構造の発見のためのチューニングに時間を要する
TCN-AE	<ul style="list-style-type: none"> ● データが少ない場合でも、単純なチューニングのみで高い異常度算出性能を発揮する ● 畳み込み構造を持つため、入力データ長による異常度算出時間への影響が少ない 	<ul style="list-style-type: none"> ● 積層数およびノード数を増加させた場合、モデルのパラメータ数が急激に増大する

7.5 実験 4: 長期間データによる故障予測実験

本節では、データセット B を用いた異常度算出を行い、異常度がどのように推移するかを調査する実験について、その条件と結果を示し、考察を行う。なお、本節で用いる提案手法は手法 2 のみとし、また使用する AE は TCN-AE とする。

7.5.1 実験条件

本実験における実験条件を表 7.15 に示す。ただし、先述の通りモデルの入力データ長には 800 ms を、TCN-AE の積層数・フィルタ数には積層数 2、ブロック数 (8,8) の 2 種類のパラメータを使用した。ただし、本実験においては、フレームシフト長は 100 ms とした。また、本実験では、図 7.3 中の初期伸びカット (③) から 1 年間分の稼働中のデータをモデル学習のためのデータとして用いた。更に、本データセットは長期的な計測データに基づく実験であるため、異常度の算出を行う際にデータの計測時の不具合などによる意図しない異常度の外れ値が発生したため、この外れ値を除外する処理を行っている。実験結果の評価は、第 6 章と同様に異常度の推移の可視化による主観的な評価、および故障前の異常度を 1 クラス目、故障後の異常度を 2 クラス目とする ROC 曲線および AUC による客観評価の双方を実施した。

表 7.15 実験 4 実験条件

条件	設定値
ダウンサンプリング後のサンプリング周波数	1,280 Hz
ファイル数	3,933
入力データ (フレーム) 長	800 ms
フレームシフト	100 ms
総データ数	393,293
学習用データ数	178,193
TCN-AE の encoder/decoder の積層数 / フィルタ数	2 / (8,8)
移動平均・移動分散を算出するフレーム幅	3,000

7.5.2 実験結果

本節における実験結果について、データごとの異常度とした異常度の推移を示す線グラフを図 7.15 に、異常度の移動平均を示す線グラフを図 7.16 に、異常度の移動分散を示す線グラフを図 7.17 にそれぞれ示す。ただし、図中の 4 本の縦線は、実験 1, 2, 3 で用いた日付と同じ位置に描画されている。また、図中の赤色で描画した線グラフおよびグラフ区間が、学習に用いたデータおよびデータ区間に相当する。続いて、故障付近の異常度の推移を確認するため、データの収集開始直後からデータの変化(④)までの期間のみの異常度を抽出してグラフ化した。その異常度の推移と移動平均、および移動分散をそれぞれ、図 7.18, 図 7.19, 図 7.20 に示す。最後に、異常度と移動平均、移動分散の ROC 曲線と AUC を図 7.21 に示す。3 種類のグラフについて、左から異常度、移動平均、移動分散の ROC 曲線および AUC をそれぞれ示している。

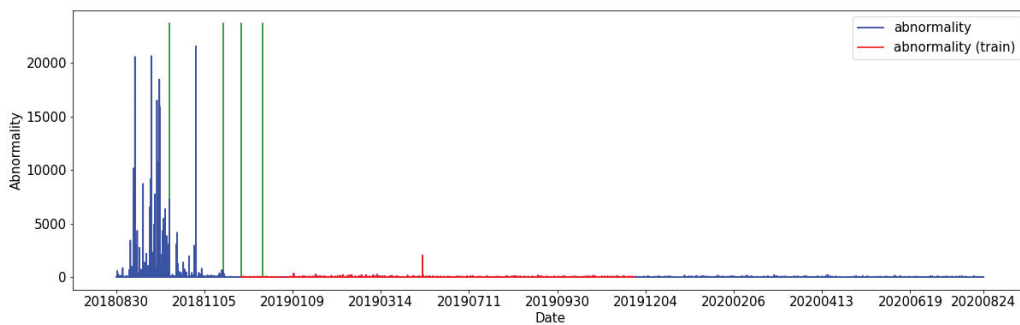


図 7.15 データセット B における異常度算出結果

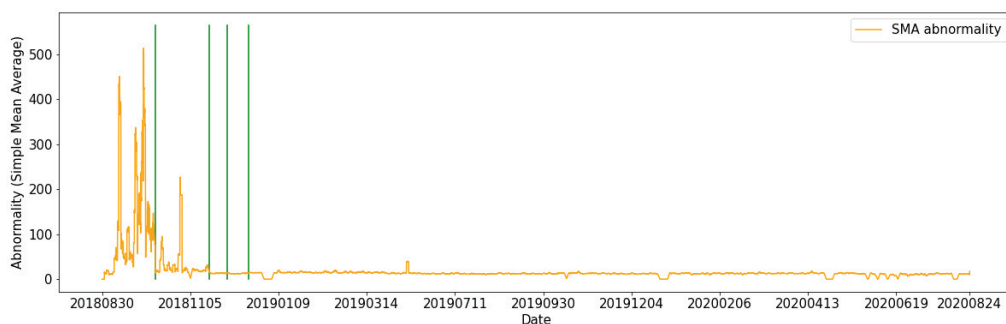


図 7.16 データセット B における異常度算出結果の移動平均

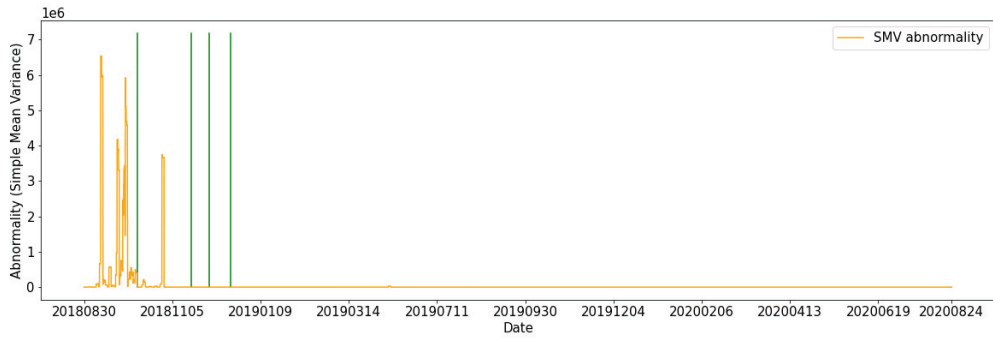


図 7.17 データセット B における異常度算出結果の移動分散

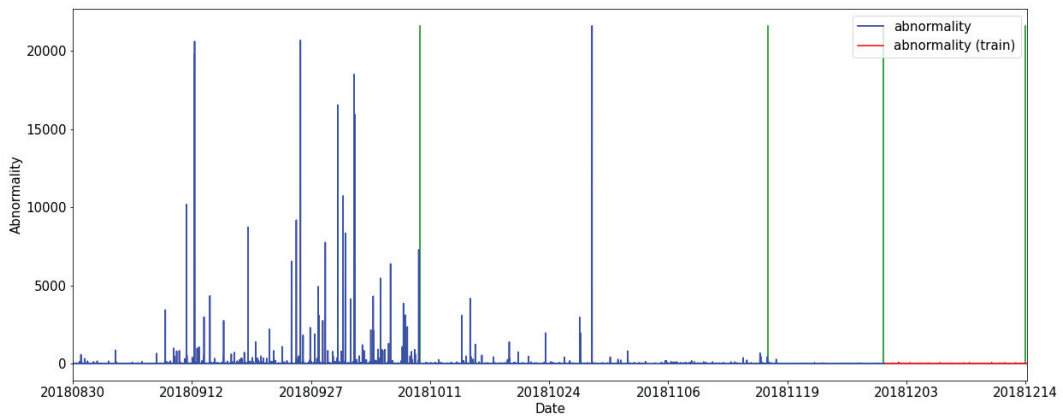


図 7.18 データセット B における異常度算出結果（一部区間のみ）

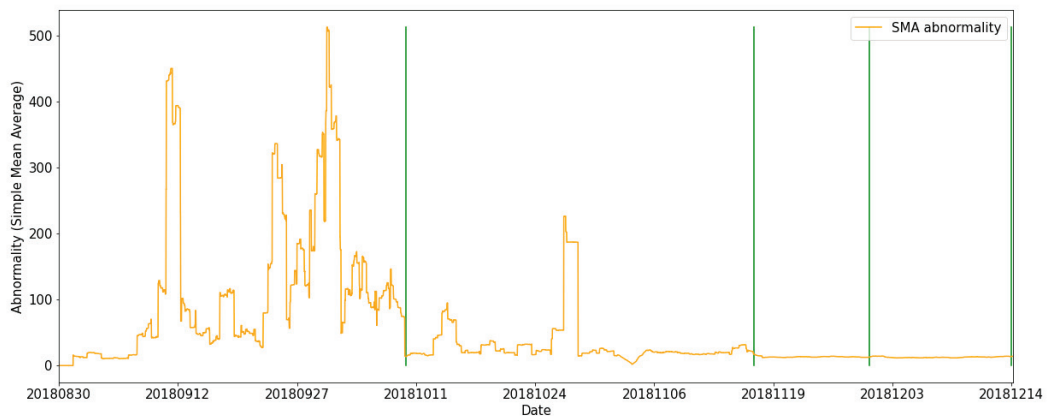


図 7.19 データセット B における異常度算出結果の移動平均（一部区間のみ）

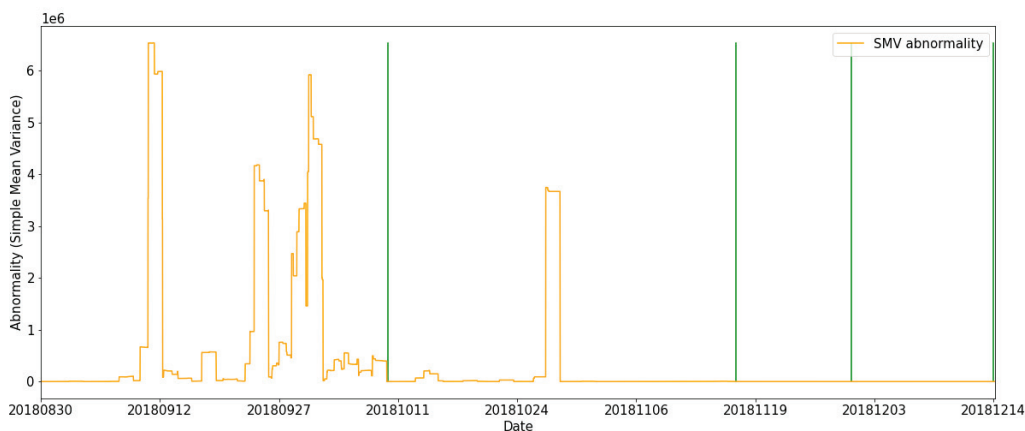


図 7.20 データセット B における異常度算出結果の移動分散（一部区間のみ）

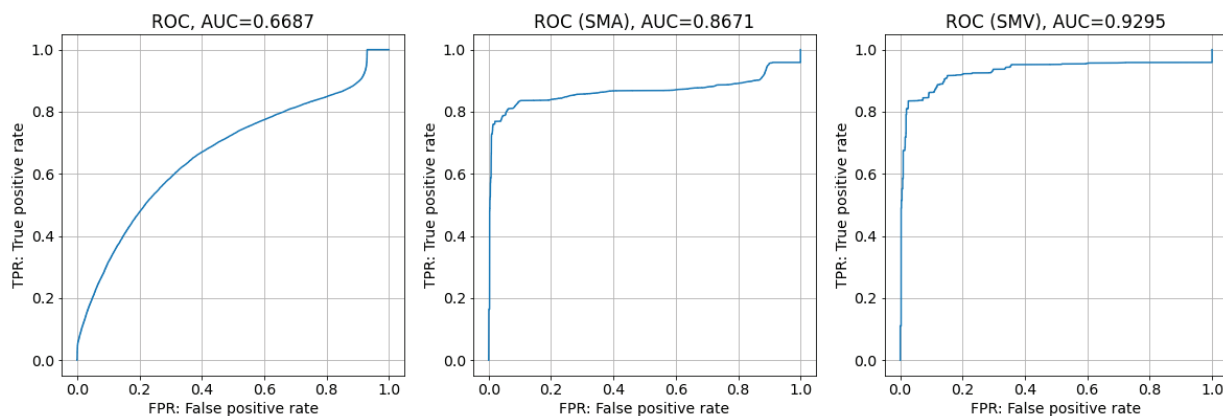


図 7.21 データセット B における異常度算出結果の ROC 曲線および AUC. 左から, 異常度, 移動平均, 移動分散の ROC 曲線および AUC をそれぞれ示す.

7.5.3 考察

図 7.21 より, 移動分散を用いた場合の AUC が 0.9295 であり, 本データセットを用いた故障前後での分類は高い精度で行えていると考えられる. このことから, 実験 1 および実験 2 と同様に, 提案手法による故障予測の有効性が示唆された. また, このことより, 本異常度算出手法による劣化状況の把握, および閾値処理による故障予測の判定の可能性および有効性も示唆された. 一方で, 本データセットにおけるチェーン交換後の異常度の段階的な上昇は確認されなかった. これは, 7.1.1 項でも触れたとおり, 実際のチェーンコンベアにおいて異常度の上昇の原因となりうる劣化が発生していないことが理由である. 本コンベアの耐用年数は最低でも 3 年であることから, さらに長期的なデータを収集し, 耐用年数以上のデータを取得した上で本手法を適用する必要がある, 本研究における今後の課題といえる.

第 8 章

結論

8.1 まとめ

本論文では、機械工具の摩耗に伴う異常検知、および劣化に伴う機械設備の故障予測において使用する異常度算出を、音声やセンサデータなどの波形信号情報を持つ時系列データを入力とし、深層学習モデルを用いて行う手法を 2 種類提案した。

1 つ目は、深層学習の一手法である再帰型 Autoencoder (AE) による特徴抽出と、機械学習の一手法である Gaussian Mixture Model (GMM) による異常度算出の 2 段階による故障予測手法 (手法 1) である。この手法では、入力には時系列データを振幅スペクトルに変換したものを利用した。

2 つ目は、AE のみを故障予測に用いる手法 (手法 2) である。センサから収集した生波形情報を入力とし、データの前処理や特徴抽出などのプロセスを内包した end-to-end 方式を採用した。構築した手法では、AE の再構成誤差を異常度の算出に利用した。

提案手法を異常検知および故障予測に適用する際、学習時には正常状態におけるデータ群のみを利用した。モデルが正常状態におけるデータの分布を学習するため、故障に近いデータを入力すると、異常度が正常状態のデータよりも高く算出されることが期待できる。

構築した手法の性能を評価するため、2 種類のデータセットによる異常検知・故障予測タスクについて、異常度を算出し異常検知・故障予測の有効性や頑健性を検証する実験を実施した。

はじめに、穴あけパンチ加工に用いられるプレス機に設置した複数のセンサデータを用いて、穴あけパンチ加工時のパンチ工具の摩耗の検知を行う実験を実施した。結果、TCN-AE により構築したモデルを用いた場合に、比較手法として使用した機械学習ベースの異常度算出モデルよりも高速に、かつ精度を維持した状態で異常検知が可能であることを示した。また、構築したモデルのうち、FFNN-AE モデルよりも TCN-AE モデルの精度が高く、かつ異常度の分散が抑えられているという結果も得られた。これは、TCN-AE を用いた場合、周波数情報と時間情報の双方をより加味した学習が可能であるため、より機械の状態に近い異常度算出が可能であることが考えられる。

つづいて、実振動データを用いて 2 種類の故障予測手法による異常度算出を行い、故障を未然に予測できるか、また各ネットワーク構造、および入力データの持つ特性により異常度の推移やモデル性能にどのような差がみられるかを検証する実験を行った。結果、2 種類の手法とも実際の故障から一定

期間前には異常度が閾値を超える傾向が見られることから、提案手法によって故障をあらかじめ予測できる可能性を示した。本データセットにおける故障予測の性能向上のため、はじめに TCN-AE における積層構造および入力データ長におけるパラメータに関する考察のための比較実験を行った。その結果から、振動周期など振動データの特徴を事前に解析し、その特徴を加味した入力データ長を選択することで、効果的な異常度算出が可能であることが示された。つづいて 3 種類の AE により異常度算出を行い、その構造の違いや異常度の推移における特徴の違いについて調査した。結果、ネットワーク構造の持つ特徴に加え、入力データの特徴を加味したモデルおよびパラメータの選択を行うことで、実際の機械設備の状態をより明瞭に可視化できる異常度算出モデルを構築できることが分かった。最後に、より長期的な収集データによるデータセットを用いた実験の結果から、本異常度算出手法による劣化状況の把握、および閾値処理による故障予測の判定の有効性も示唆された。

8.2 今後の課題

今後の課題として、次の 2 点が挙げられる。

はじめに、より長期的なデータ収集を行い、劣化に伴う故障が発生しているデータを用いた故障予測の実施が挙げられる。7.5 節で使用したデータは 2 年分であり、7.2 節から 7.4 節までに利用していたデータよりは長期間のデータではあるものの、劣化に伴う故障が新たに発生していないデータであった。対象となるチェーンコンベアの耐用年数は最低でも 3 年であるため、今回取得したデータよりも更に長期的なデータを収集し、耐用年数以上のデータを取得した上で再度本手法を適用することにより、本手法の有効性を示す必要がある。

また、複数の Autoencoder や機械学習手法によるアンサンブル異常検知および故障予測手法の構築も、今後の課題として挙げられる。本論文で提案した 2 種類の異常度算出手法を比較した場合、手法 1 は異常度算出手法に GMM を用いていることから、現状の手法 2 では困難である異常判定の理由の説明可能性をもつという利点を持つ一方で、2 段階手法であることから異常度算出までの時間が手法 2 と比較して長くなるという欠点がある。また、本論文で扱った 3 種類の AE 構造についても、それぞれ特性や利点を持っている。これらの手法の特性を組み合わせ、アンサンブル的にデータの持つ分布を学習することで、より頑健な、かつ説明可能な異常度算出を行うことが期待できる。

謝辞

はじめに、本研究をはじめ、学部3年から博士課程までの6年半もの間、研究方針や計画に関するご指導をいただきました岐阜大学工学部 速水悟教授，田村哲嗣准教授に深く感謝申し上げます。また、本論文を作成するにあたり、ご指摘および助言をいただきました、岐阜大学工学部 原武史教授，古屋耕平准教授に深く感謝申し上げます。

本研究を進めるにあたり、共同で研究を行わせていただいた KYB 株式会社様，および井指諒亮氏，古川輝氏，内藤孝昌氏には，データを提供いただき，研究成果に関して議論を行い，数多くの助言をいただきました。深く感謝申し上げます。また，共同で研究を行わせていただいた Stuttgart University Institute for Metal Forming Technology Prof. Mathias Liewald, Dr. Celalettin Karadogan には，データを提供いただき，また数多くの議論を通して研究成果に関するご指摘および助言をいただきました。深く感謝申し上げます。

博士課程において異常検知研究を行うにあたり，岐阜大学スマート金型開発拠点におけるデータ分析を担当させていただいた経験は，本論文を記述する上で非常に重要な知見となりました。岐阜大学工学部 王志剛教授，箱山智之助教，仲井朝美教授，および参画企業である株式会社デンソー様，株式会社チュキョー様，アサヒフォージ株式会社様，株式会社岐阜多田精機様，扶桑工機株式会社様，日本ユニシス・エクセリョーションズ株式会社様，ユニアデックス株式会社様の関係者の皆様には，多大なご助言，ご指導を賜りましたことに，深く感謝申し上げます。

岐阜大学キッズプログラミング研究会を5年にわたってリードいただき，プログラミング教育に関する多くの経験やノウハウ，知見を習得する機会を与えていただきました，株式会社タノキバ 村山聡江氏，岐阜大学工学部 加藤邦人准教授，寺田和憲准教授，山口忠教授，深井英和助教に深く感謝申し上げます。

修士課程より研究を行っておりますピアノ学習支援システムに関する研究において，データの撮影に協力いただき，また議論を通して多大なご指導，ご指摘，助言をいただきました，中部学院大学短期大学部 杉山祐子教授，岐阜大学教育学部 今井亜湖准教授に深く感謝申し上げます。

また，学部当時の研究室の同期であり，博士課程に入学後ともに株式会社 Lukos を設立した宮崎晃一氏には，特に会社設立および運営に関して数多くの議論を重ね，岐阜大学発ベンチャーとして活動することができたことを，大変光栄に思っております。また，会社設立および運営に関して多くの助言をいただきました，岐阜大学 上原雅行准教授，および岐阜県産業経済振興センターの関係者の皆様にも，深く感謝申し上げます。そして，岐阜大学起業部の創設に携わった岐阜大学 藤田一實氏，長曾

我部竣也氏，神谷海渡氏，伊藤彰彦氏には，深く感謝申し上げるとともに，起業部の更なる発展を願っております。

そして，速水・田村研究室 異常検知班のメンバーである松井彩華氏，元水健人氏，森駿登氏，山下純也氏には，ミーティングをはじめ，様々な場面において貴重なご意見，ご指摘をいただき，かつ同じ研究分野において多分にご助力いただきましたことに，深く感謝申し上げます。

最後に，速水・田村研究室の現役生の皆様および OB・OG の皆様とともに，6 年半に渡る研究生生活を通し，数えきれないほどの知識やノウハウを学習・経験することが出来ました。この知識や経験は，今後の私の人生において基軸となると確信しております。皆様に，深く感謝申し上げます，

参考文献

- [1] "ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)", <http://image-net.org/challenges/LSVRC/2012/>, 参照 Oct. 28, 2020,
- [2] Krizhevsky, A., Sutskever, I. and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097-1105, 2012.
- [3] "Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring," <http://dcase.community/challenge2020/task-unsupervised-detection-of-anomalous-sounds>, 参照 Oct. 19, 2020.
- [4] 西田勇, 立川宗治, 佐藤隆太, 白瀬敬一, "切削トルクの予測結果と実測結果の定量比較による加工異常検出システム (第1報 工具摩耗による加工異常検出)," *日本機械学会論文集*, Vol.84, No.857, 2018, DOI:10.1299/transjsme.17-00433.
- [5] 近藤稔, "振動のオクターブバンド分析を用いた異常検知法による車両機器の診断," *日本機械学会論文集*, Vol.84, No.862, 2018, DOI:10.1299/transjsme.17-00594.
- [6] 中塚俊介, 相澤宏旭, 加藤邦人, "少数不良品サンプル下における Adversarial AutoEncoder による正常モデルの生成と異常検出," *精密工学会誌*, Vol.84, No.12, pp.1071-1078, 2018.
- [7] Chalapathy, R. and Chawla S., "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.
- [8] Matsui, A., Asahi, S., Tamura, S., Hayamizu, S., Isashi, R., Furukawa, A. and Naito, T., "Anomaly detection in mechanical vibration using combination of signal processing and autoencoder," *Proc. RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing 2020*, pp.139-142, 2020.
- [9] Asahi, S., Matsui, A., Tamura, S., Hayamizu, S., Isashi, R., Furukawa, A. and Naito, T., "Failure Prediction Method for Mechanical Equipment from Vibration Data With LSTM Autoencoder and GMM," *Proc. RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing 2020*, pp.139-142, 2020.
- [10] 井手剛, 杉山将, "異常検知と変化検知," 講談社, 2015.
- [11] Baldi, P., "Autoencoders, unsupervised learning, and deep architectures," *Proc. International Conference on Machine Learning*, pp.37-49, 2012.

-
- [12] Pang, G., Shen, C., Cao, L. and Hengel, A. V. D., "Deep learning for anomaly detection: A review," arXiv preprint arXiv:2007.02500, 2020.
- [13] Ding, N., Ma, H., Gao, H., Ma, Y. and Tan, G., "Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model," *Computers & Electrical Engineering*, Vol.79, 2019.
- [14] Singh, K., Rajora, S., Vishwakarma, D. K., Tripathi, G., Kumar, S. and Walia, G. S., "Crowd anomaly detection using aggregation of ensembles of fine-tuned convnets," *Neurocomputing*, Vol.371, pp.188-198, 2020.
- [15] Chalapathy, R., Menon, A. K. and Chawla, S., "Anomaly detection using one-class neural networks," arXiv preprint arXiv:1802.06360, 2018.
- [16] Kingma, D. P. and Welling, M., "An introduction to variational autoencoders," arXiv preprint arXiv:1906.02691, 2019.
- [17] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial networks," arXiv preprint arXiv:1406.2661, 2014.
- [18] Zimmerer, D., Kohl, S. A., Petersen, J., Isensee, F. and Maier-Hein, K. H., "Context-encoding variational autoencoder for unsupervised anomaly detection," arXiv preprint arXiv:1812.05941, 2018.
- [19] Kaplan, M. O. and Alptekin, S. E., "An improved BiGAN based approach for anomaly detection," *Procedia Computer Science*, Vol.176, pp.185-194, 2020.
- [20] Le Roux, J., Hershey, J. R. and Wainwright, F., "Deep NMF for speech separation," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing 2015*, pp. 66-70, 2015.
- [21] Li, H. C., Yang, G., Yang, W., Du, Q. and Emery, W. J., "Deep nonsmooth nonnegative matrix factorization network with semi-supervised learning for SAR image change detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol.160, pp.167-179, 2020.
- [22] de La Bourdonnaye, F., Teuliere, C., Chateau, T. and Triesch, J., "Learning of binocular fixations using anomaly detection with deep reinforcement learning," *Proc. International Joint Conference on Neural Networks 2017*, pp. 760-767, 2017.
- [23] Yu, M. and Sun, S., "Policy-based reinforcement learning for time series anomaly detection," *Engineering Applications of Artificial Intelligence*, Vol.95, 2020, DOI:10.1016/j.engappai.2020.103919.
- [24] 吉澤正, "高速フーリエ変換とその応用," *計測と制御*, Vol.8, No.12, pp.851-860, 1969.
- [25] Miyake, A. and Shinmura, S., "An algorithm for the optimal linear discriminant functions," *Proc. International Conference on Cybernetics and Society*, pp.1447-1450, 1978.
- [26] Cover, T., Peter, H., "Nearest neighbor pattern classification," *IEEE transactions on infor-*

- mation theory, Vol.13, No.1, pp.21-27, 1967.
- [27] Tax, D. M., Duin, R. P., "Support vector data description," *Machine learning*, Vol.54, No.1, pp.45-66, 2004.
- [28] Pang, G., Yan, C., Shen, C., Hengel, A. V. D. and Bai, X., "Self-trained deep ordinal regression for End-to-End video anomaly detection," *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*, pp. 12173-12182, 2020.
- [29] Perera, P., Nallapati, R. and Xiang, B., 2019 "Ocgan: One-class novelty detection using gans with constrained latent representations," *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019*, pp. 2898-2906, 2019.
- [30] Frénay, B. and Verleysen, M., "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, Vol.25, No.5, pp.845-869, 2013.
- [31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., "Attention is all you need," *Advances in neural information processing systems*, pp.5998-6008, 2017.
- [32] Srivastava, N., Mansimov, E. and Salakhudinov, R., "Unsupervised learning of video representations using lstms," *Proc. International Conference on Machine Learning 2015*, pp.843-852, 2015.
- [33] Welch, P., "The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms", *IEEE Transactions on Audio and Electroacoustics*, Vol.15, No.2, pp.70-73, 1967.
- [34] 井手剛, "入門 機械学習による異常検知—R による実践ガイド," コロナ社, 2015.
- [35] Khreich, W., Khosravifar, B., Hamou-Lhadj, A., Talhi, C., "An anomaly detection system based on variable N-gram features and one-class SVM," *Information and Software Technology*, Vol.91, pp.186-197, 2017.
- [36] Werbos, P. J., "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, Vol.78, No.10, pp.1550-1560, 1990.
- [37] Gers, F. A., Schmidhuber, J. and Cummins, F., "Learning to forget: Continual prediction with LSTM," *Neural computation*, Vol.12, No.10, pp.2451-2471, 2000.
- [38] Hochreiter, S., Schmidhuber, J., "Long short-term memory," *Neural computation*, Vol.9, No.8, pp.1735-1780, 1997.
- [39] C. Lea, M. D. Flynn, R. Vidal, A. Reiter and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," *Proc. IEEE Conference on Computer Vision and Pattern Recognition 2017*, pp.156-165, 2017.
- [40] Srivastava, N., Mansimov, E. and Salakhudinov, R., "Unsupervised learning of video representations using lstms," *Proc. International Conference on Machine Learning 2015*, pp.843-852, 2015.

-
- [41] Abadi, M., Barham, P., et al, "TensorFlow: a system for large-scale machine learning," Proc. 12th USENIX conference on Operating Systems Design and Implementation, pp.265-283, 2016.
- [42] "Keras Documentation," <https://keras.io/>, 参照 Oct. 27, 2020.
- [43] S. Bai, J. Z. Kolter and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.

発表文献

- [1] 朝日翔太, 田村哲嗣, 速水悟, ”楽曲における印象に対する個人差の影響の調査, ” 日本音響学会 2015 年秋季研究発表会, pp.761-762, 2015.
- [2] 朝日翔太, 杉山祐子, 田村哲嗣, 速水悟, ”ピアノ初学者のための練習支援動画『映像テキスト』の制作, ” 日本教育工学会第 32 回全国大会, pp.995-996, 2016.
- [3] Asahi, S., Tamura, S., Hayamizu, S. and Sugiyama, Y., “Estimation of tempo, timing, and melody for piano practice support systems,” The Journal of the Acoustical Society of America, Vol.140, No.4, pp.3429-3429, 2016.
- [4] 朝日翔太, 杉山祐子, 田村哲嗣, 速水悟, ”練習支援動画『映像テキスト』による演奏イメージの可視化 一視線動向追跡実験による, ” 日本教育工学会第 33 回全国大会, pp.881-882, 2017.
- [5] 朝日翔太, 田村哲嗣, 速水悟, 杉山祐子, ”ピアノ学習支援システムのための LSTM を用いたタイミング・ピッチ推定法, ” 日本音響学会 2017 年秋季研究発表会, pp.1279-1280, 2017.
- [6] 朝日翔太, 高橋和之, 村山聡江, 寺田和憲, 加藤邦人, 山口忠, 今井亜湖, 速水悟, ”初等教育におけるテキスト型プログラミング言語 Python によるプログラミング教育の効果検証, ” 日本教育工学会第 34 回全国大会, pp.231-232, 2018.
- [7] Asahi, S., Tamura, S., Sugiyama, Y. and Hayamizu, S., ”Toward a High Performance Piano Practice Support System for Beginners,” Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2018, pp.73-79, 2018.
- [8] 赤澤玲, 朝日翔太, 高橋和之, 村山聡江, 寺田和憲, 加藤邦人, 山口忠, 今井亜湖, 速水悟, ”初・中等プログラミング教育におけるテキスト型プログラミング言語 Python の実用可能性, ” 日本教育工学会 2019 年秋季全国大会, pp.597-598, 2019.
- [9] Asahi, S., Matsui, A., Tamura, S., Hayamizu, S., Isashi, R., Furukawa, A. and Naito, T., ”Failure Prediction Method for Mechanical Equipment from Vibration Data With LSTM Autoencoder and GMM,” 2020 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing 2020, pp.65-68, 2020.
- [10] Matsui, A., Asahi, S., Tamura, S., Hayamizu, S., Isashi, R., Furukawa, A. and Naito, T., ”Anomaly Detection in Mechanical Vibration Using Combination of Signal Processing and Autoencoder,” 2020 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing 2020, pp.139-142, 2020.

-
- [11] Matsui, A., Asahi, S., Tamura, S., Hayamizu, S., Isashi, R., Furukawa, A. and Naito, T., "Anomaly Detection in Mechanical Vibration Using Combination of Signal Processing and Autoencoder," *Journal of Signal Processing*, Vol.24, No.4, pp.203-206, 2020, DOI: 10.2299/jsp.24.203.
- [12] 朝日翔太, 松井彩華, 田村哲嗣, 速水悟, 井指諒亮, 古川輝, 内藤孝昌, "再帰型オートエンコーダを用いた振動データによる工場設備の故障予測手法の提案," *日本機械学会論文集*, Vol.86, No.891, pp.20-00020, 2020, DOI:10.1299/transjsme.20-00020.
- [13] 朝日翔太, 松井彩華, 田村哲嗣, 速水悟, 井指諒亮, 古川輝, 内藤孝昌, "工場設備の故障予測における Autoencoder を用いた異常度算出手法の分析," *電子情報通信学会論文誌*, Vol.J104-D, No.04, 2021, DOI: 10.14923/transinfj.2020PDP0042.