

大規模時系列テンソルからの長期イベント予測

本田 崇人[†] 松原 靖子[†] 川畑 光希[†] 櫻井 保志[†]

[†] 大阪大学 産業科学研究所 〒567-0047 大阪府茨木市美穂ヶ丘 8-1

E-mail: †{takato88,yasuko,koki88,yasushi}@sanken.osaka-u.ac.jp

あらまし 本論文では、大規模時系列センサデータのためのイベント予測手法である SplitCast について述べる。SplitCast は (*facility, sensor, time*) の三つ組で構成される大規模時系列センサデータから、複数の観点に基づく多角的な時系列パターンを統合的に解析、要約し、長期的な将来イベント予測を行う。具体的には、工場設備に取り付けられた回転速度、稼働電圧、設備温度など、各設備におけるセンサデータの実測値で構成される時系列データが与えられたとき、提案手法は、(a) 基本的な時系列パターン、各設備間における共通パターンや設備固有のパターンを抽出し、統計的に要約することで、(b) 将来的なイベントを予測する。さらに、これらの処理は (c) データサイズに対して線形である。実データを用いた実験では、提案手法が時系列センサデータに含まれる特徴的な時系列パターンを多角的に捉え長期的なイベント予測を行うことを確認し、さらに、最新の既存手法と比較し大幅な精度、性能向上を達成していることを明らかにした。

キーワード 時系列テンソル, センサデータ, イベント予測

1 まえがき

近年、製造業においては製造工場のスマート化が推し進められている。大量のセンサを使用して生産ラインの稼働状況を常に監視し、その様子を蓄積、分析することにより、機器の異常検知 [26], [31] や品質管理 [15] 等、あらゆる側面から生産性を向上する取り組みが行われている。これらの取り組みに共通する重要な課題は、収集した大規模データからの効果的な知見獲得と、それに基づく将来予測技術の開発である。

特に、製造工場から得られる時系列データは複数のドメイン（設備、センサ、時間など）を持つ複雑なデータであり、多角的なパターンを持つことが多い。生産ラインにおいては、複数の作業工程（パターン）の時間遷移のみならず、複数ラインでの並列作業によって生まれる作業ラインごとに共通/相違なパターンを持つ。不良品や設備故障の要因を効果的に捉えるためには、このような多角的かつ動的なパターンを柔軟に表現すると同時に、それらの間に隠された因果関係を明らかにする必要がある。

工場設備で想定されるタスクでは、故障や不具合、加工精度の低下など、各イベントの発生をより早く把握することで対策の選択肢が広がり、工場稼働率の上昇や品質向上が見込まれる。つまり、大規模センサデータの将来予測技術は、より長期的な予測能力を有することが望ましい [16]。

本論文では、大規模時系列センサデータのためのイベント予測手法である SPLITCAST について述べる。SPLITCAST は、時系列データに含まれる典型的なパターン（本研究では、“レジーム”と呼ぶ）の数と変化点を多角的に捉え、システムの稼働状況を正確に把握することで、将来発生するイベントを予測する。より具体的には、次の問題を扱う。

問題: 複数箇所の設備で複数のセンサから収集された大規模時系列センサデータが与えられたとき、 l_s ステップ先のイベン

トを予測する。まず、(a) センサデータの中から多角的なパターンとその変化点を検出し、それらを要約情報としてまとめることで (b) 長期的かつ高精度な予測を実現する。さらに、(c) これらの処理は高速に行う。

1.1 具体例

図 1 工場設備データにおける SPLITCAST の出力結果を示している。図 1(a) はオリジナルのセンサデータを表し、5 つの設備における 3 つのセンサ値（回転速度、稼働電圧、設備温度）で構成される。黒矩形で塗られた箇所は設備が非常停止中（イベント発生中）であることを示す。

図 1(b) は SPLITCAST によるオリジナルデータからのパターン検出結果である。図中の縦線は時系列パターンが変化した時刻を示し、同一レジームに属するセグメントは同一色で塗られている。提案手法は、複数の設備から得られた時系列データを同時に解析することにより、多角的なパターン、すなわち、各設備内のパターンの時間遷移だけでなく、設備間で共通、あるいは相違なパターンを検出することが可能である。

図 1(c) に、オリジナルデータの中から、 $l_s = 200$ ステップ（約 17 分）後に非常停止した場合とそうでない場合の典型的な例を図示した。各図の左はセグメンテーション結果を示す。右の $\theta_1 \sim \theta_5$ はそれぞれ共通の時系列パターン（すなわちレジーム）を表し、それらの遷移の様子を可視化したものである。 p_{200} の値は、左図にある部分シーケンスとそのパターン検出結果が与えられたとき、提案手法が出力した 200 ステップ先での非常停止確率である。右図において、より多くの遷移が検出されたレジーム間には太い矢印が表示される。また、円の大きさはレジームの発生期間の大きさを示す。図 1(c-ii) を見ると、設備が非常停止する前に回転速度が上昇 (θ_5) しており、その傾向はレジーム θ_4 , θ_5 の遷移が現れることによって表現されている。実際に、提案手法は非常停止を正確に予測し、 p_{200} が高い値を

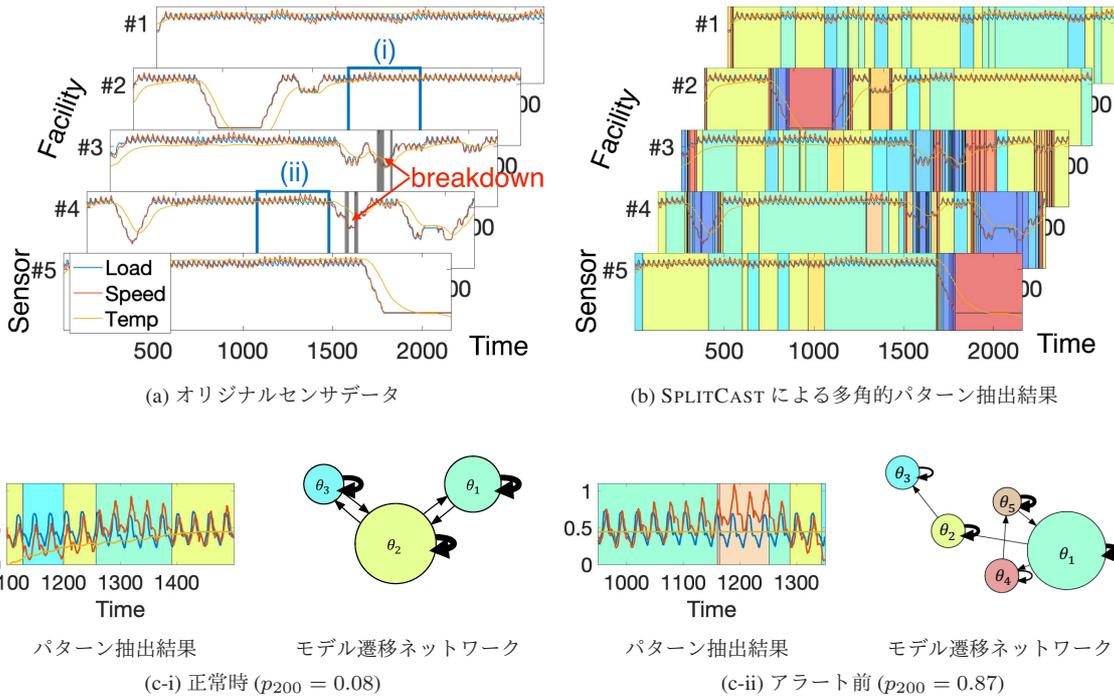


図 1: スマート工場センサデータにおける SPLITCAST によるパターン検出と 200 ステップ先のアラート予測 (p_{200}) 結果

示している。つまり、データに含まれる潜在的なパターンを検出することで、非常停止に至る過程を多角的に分析できるだけでなく、それらの要約情報を用いることで長期的かつ高精度な予測が可能となる。

1.2 本研究の貢献

本研究では、大規模時系列データのためのイベント予測手法である SPLITCAST を提案する。本手法は次の特徴を持つ。

- データ内に含まれる時系列パターンに関する事前知識を必要とせず、パターン(レジーム)の変化点と潜在的な振る舞いを、時間遷移と設備間の多角的な観点から把握する。
- 大規模時系列データに含まれる複雑なパターンを要約し、要約情報に基づく効果的なイベント予測を行う。
- モデル推定とイベント予測に必要な計算コストはデータサイズに対して線形である。

2 関連研究

センサデータの解析に関する研究は、データベースやデータマイニング等、様々な分野で進められている [2], [18], [20], [23], [26]。自己回帰モデルや線形動的システムは代表的な技術であり、これらに基づくセンサデータの解析と予測手法が数多く存在する [14]。RegimeCast [16] は大量に生成され続ける多次元センサデータから非線形動的システムをリアルタイムに推定し、適応的に将来を予測し続ける能力を持つ。本手法はセンサストリームを入力とし、センサデータの実測値の予測において高い性能を示すが、ラベル付けされたイベントデータの予測には対応していない。

また、時系列ビッグデータを対象としたパターン発見とクラスタリングも重要な課題である [8], [11], [12], [17], [27], [28], [30]。

Matsubara ら [19] は大規模イベントテンソルの解析手法として TriMine を提案した。TriMine は与えられたデータを複数のトピックに分類し、潜在的なトレンドやパターンを検出するが、Web クリックログのような離散イベントデータを対象としており、IoT センサデータのような時系列シーケンスの動的パターンやそのグループ(レジーム)を表現することができず、扱う問題が異なる。加えて、TriMine はイベント予測能力を持たない。

Deep Neural Network に基づく非線形動特性の解析に関する研究も盛んである [3], [9], [10]。Qin らは文献 [22] において、入力時系列の中で重要な次元と次元削減後の特徴空間で重要な次元を 2 階層にわたりモデル化することで、高精度に株価を予測する手法を提案した。一方で、本研究のように、不連続に発生するイベントを予測するタスクではイベントの発生強度(Intensity)をモデル化する手法が主流である [5], [6], [21], [29]。例えば、RMTTP [5] は、過去のイベント履歴から次に発生するイベントの時刻と種類を予測するための非線形モデルを提案している。しかし、これらの手法はイベント履歴のみから構成されるカテゴリカルデータを対象としており、センサからの実測値で構成される連続データによるイベント予測を行うことはできない。結論として、時系列テンソルを対象としたイベント予測手法はこれまで存在せず、提案手法が世界初の試みである。

3 問題定義

本研究で扱う工場設備センサデータは、三菱重工エンジン & ターボチャージャ株式会社で 2017 年 10 月 1 日に稼働した 55 設備における 3 種類のセンサデータで構成される。本データは (*facility, sensor, time*) の三つ組で表現され、それぞれ、 w 個の設備、 d 種のセンサ、 n の期間 (5 秒単位) から成る。このセンサデータは、3 階のテンソル $\mathcal{X} \in \mathbb{R}^{w \times d \times n}$ として表現するこ

とができ、 \mathcal{X} の要素 $x_{ij}(t)$ は時刻 t における i 番目の設備の j 番目のセンサでの計測値を示す。本論文では、このセンサデータを多次元時系列テンソルと呼ぶ。本研究の最終目的は、与えられた時系列テンソル \mathcal{X} から l_s ステップ先のイベントを予測することであり、取り組む問題を以下のように定義する。

[問題 1] 時系列テンソル $\mathcal{X}(t_s : t_e)$ が与えられたとき、 l_s ステップ先のイベント $\mathcal{Y}(t_e + l_s)$ を次式に基づいて予測する。

$$\mathcal{Y}(t_e + l_s) \approx F(\mathcal{X}(t_s : t_e)). \quad (1)$$

ここで、 $t_s : t_e$ は予測に使用するテンソルのウィンドウを表し、 F を提案モデルとする。

本論文では、 $\mathcal{Y}(t_e + l_s)$ を高精度に予測するために、確率モデルと深層学習に基づくモデルを構築し、与えられたセンサデータから故障の要因となる高次元かつ非線形な動的特性を抽出する。具体的には、提案モデルは次の 3 つの能力を有する。

- (P1) 潜在的な動的パターンの多角的な検出
- (P2) 動的パターンに基づく特徴抽出
- (P3) l_s ステップ先の長期イベント予測

4 提案モデル

4.1 (P1) 潜在的な動的パターンの検出

多次元時系列テンソル \mathcal{X} が与えられたとき、提案手法はまず、 \mathcal{X} を m 個のセグメント集合 $\mathcal{S} = \{s_1, \dots, s_m\}$ に分割してその特徴をとらえる。 s_i は i 番目のセグメントの開始点 t_s 、終了点 t_e 、設備番号で構成され (つまり、 $s_i = \{t_s, t_e, facilityID\}$)、各セグメントは重複がないものとする。そして、発見したセグメント集合を類似セグメントのグループに分類する。本論文ではこれらのグループをレジームと呼ぶ。

[定義 1] (レジーム) r を最適なセグメントグループの個数とする。それぞれのセグメント s はセグメントグループの 1 つに割り当てられる。

さらに、各セグメントが所属するレジームを表現するため、新たにセグメントメンバーシップを定義する。

[定義 2] (セグメントメンバーシップ) 時系列テンソル \mathcal{X} が与えられたとき、 $\mathcal{F} = \{f_1, \dots, f_m\}$ を、 m 個の整数列とし、 f_i を i 番目のセグメントが所属するレジームの番号とする。

これにより、多次元時系列テンソル \mathcal{X} を m 個のセグメントと r 個のレジームで $\{m, r, \mathcal{S}, \Theta, \mathcal{F}\}$ として表現することができる。次に提案手法は、得られたレジーム情報に基づき、多次元時系列テンソルを統計モデル化し重要な特徴を抽出する。

4.2 (P2) 動的パターンに基づく特徴抽出

それぞれのレジームは統計モデル $\Theta = \{\theta_1, \dots, \theta_r, \Delta_{r \times r}\}$ として表現される。本研究では、多次元時系列テンソルの振る舞いを表現するため、隠れマルコフモデル (HMM: Hidden Markov Model) を用いる。HMM は隠れ状態を持つマルコフ過程を仮定した確率モデルの一種であり、音声認識を含む様々な分野において、時系列処理手法として広く利用されている。HMM は初期確率 $\pi = \{\pi_i\}_{i=1}^k$ 、遷移確率 $\mathbf{A} = \{a_{ij}\}_{i,j=1}^k$ 、出力確率 $\mathbf{B} = \{b_i(\mathbf{x})\}_{i=1}^k$ の三つ組で表現される (すなわち、

$\theta = \{\pi, \mathbf{A}, \mathbf{B}\}$)。ここで、 k は HMM の潜在状態数を示す。本論文では出力確率 \mathbf{B} が多次元ガウス分布から生成されるものとする。これにより多次元ベクトルのシーケンスを確率モデルで表現する (つまり $\mathbf{B} \sim \{\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)\}_{i=1}^k$)。HMM のモデルパラメータ $\theta = \{\pi, \mathbf{A}, \mathbf{B}\}$ と、あるユーザのシーケンス \mathbf{X} が与えられた時、 \mathbf{X} の尤度 $P(\mathbf{X}|\theta)$ は次のように計算される：

$$P(\mathbf{X}|\theta) = \max_{1 \leq i \leq k} \{p_i(n)\}$$

$$p_i(t) = \begin{cases} \pi_i \cdot b_i(\mathbf{x}_1) & (t = 1) \\ \max_{1 \leq j \leq k} \{p_j(t-1) \cdot a_{ji}\} \cdot b_i(\mathbf{x}_t) & (2 \leq t \leq n) \end{cases}$$

ここで、 $p_i(t)$ は時刻 t における潜在状態 i の最大確率を示し、 n は \mathbf{X} のシーケンス長である。この尤度は図 3 に示す遷移図に基づき、動的計画法の一種であるビタビアルゴリズム [7] を用いて計算される。さらに、新たな概念としてレジーム遷移行列 $\Delta_{r \times r}$ を導入する。

[定義 3] (レジーム遷移行列) $\Delta_{r \times r}$ を r 個のレジーム群の遷移行列と呼ぶ。ここで、要素 $\delta_{ij} \in \Delta$ は i 番目のレジームから j 番目のレジームへの遷移確率を示す。

上記のモデルを用いて、時系列テンソル \mathcal{X} を、以下に示す HMM の潜在状態系列 \mathcal{Z} とモデル化した際の誤差 \mathcal{E} で要約し、特徴量化することで、高精度かつ長期的な予測を実現する。

[定義 4] (潜在状態テンソル) 各設備ごとの HMM の潜在状態系列 $\mathcal{Z} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_w\}$ を潜在状態テンソルと呼ぶ。ここで、 $\mathbf{Z}_i = \{z_{ij}(1), \dots, z_{ij}(n)\}_{j=1}^d$ であり、 $z_{ij}(t)$ は自身と同じ状態であるデータ集合 \mathbf{x} の平均と分散の組 $\{\mu, \sigma\}$ で構成される。

[定義 5] (誤差テンソル) 多次元時系列テンソル \mathcal{X} を潜在状態テンソル \mathcal{Z} でモデル化した際の誤差 $\mathcal{E} = \{\mathbf{E}_1, \dots, \mathbf{E}_w\}$ を誤差テンソルと呼ぶ。本論文では HMM の出力確率 \mathbf{B} が多次元ガウス分布に従うと仮定しているため、 i 番目の設備の j 番目のセンサにおける時刻 t での誤差 $e_{ij}(t) \in \mathbf{E}_i$ は、以下のように表現される。 $e_{ij}(t) = P(\mathcal{X}_{ij}(t)|z_{ij}(t)) \sim \mathcal{N}(\mu, \sigma^2)$
 $\mu, \sigma \in z_{ij}(t)$

すなわち、時系列テンソル \mathcal{X} を、(P1) で得られたレジーム情報 $\{m, r, \mathcal{S}, \Theta, \mathcal{F}\}$ に基づき $\mathcal{X} \approx \text{IGPDF}(\mathcal{Z}, \mathcal{E})$ となるようなテンソル \mathcal{Z} と \mathcal{E} によって要約し、重要な特徴を抽出する。ここで IGPDF (Inverse Gaussian Probability Density Function) はガウス分布における確率密度関数の逆関数を表す。

4.3 (P3) l_s ステップ先の長期予測

結論として、式 1 は、

$$\mathcal{Y}(t_e + l_s) \approx F(\{\mathcal{Z}(t_s : t_e), \mathcal{E}(t_s : t_e)\}) \quad (2)$$

のように言い換えられる。ここで、 F は予測モデルを表す。

5 アルゴリズム

これまで、多次元時系列テンソル \mathcal{X} を要約し、効果的に予測するための提案モデルについて述べた。本章では、問題 1 を解決するためのアルゴリズムについて説明を行う。ここで問題となるのは、どのようにレジームやセグメントの数を決定するかである。提案手法は、最小記述長の概念に基づき、適切なモデルを生成するための基準となる符号化スキームを導入する。

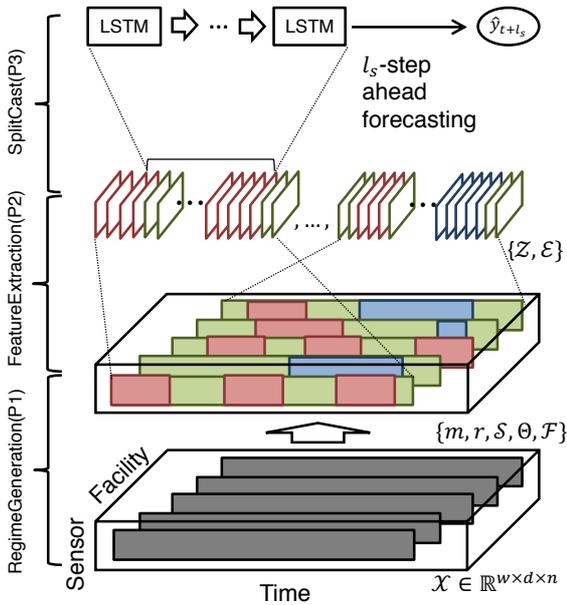


図 2: SPLITCAST のアルゴリズム概要.

5.1 モデル選択とデータ圧縮

直感的には、データが与えられたときのモデルのよさは次の式で表現できる.

$$Cost_T(\mathcal{X}; \mathcal{M}) = Cost_M(\mathcal{M}) + \alpha \cdot Cost_C(\mathcal{X}|\mathcal{M}) \quad (3)$$

ここで、 $Cost_M(\mathcal{M})$ はモデル \mathcal{M} を表現するためのモデルコストを示し、 $Cost_C(\mathcal{X}|\mathcal{M})$ は \mathcal{M} が与えられたときのテンソル \mathcal{X} の符号化コストを示す。 α は符号化コストに対する重み¹であり、 α の値が大きいくほどより実データに正確なモデルを生成する (セグメントの数 m 、レジームの数 r が大きくなる).

モデルコスト. 具体的には、SPLITCAST の全パラメータ集合の表現コストは以下の要素で構成される: 時系列テンソル \mathcal{X} のサイズ: $\log^*(w) + \log^*(d) + \log^*(n)$ ビット², セグメント集合 \mathcal{S} : $\log^*(m) + \sum_{i=1}^{m-1} \log^* |s_i|$ ビット, レジーム割り当て \mathcal{F} : $m \log(r)$ ビット, レジームパラメータ集合 Θ : $\sum_{i=1}^r Cost_M(\theta_i) + Cost_M(\Delta)$ ビット. ここで、浮動小数点のコストを c_F とすると、 k 個の状態を持つ単一のレジームパラメータ θ は $Cost_M(\theta) = \log^*(k) + c_F \cdot (k + k^2 + 2kd)$, レジーム遷移行列 Δ は $Cost_M(\Delta) = c_F \cdot r^2$ のコストを要する.

符号化コスト. モデルが与えられたときの \mathcal{X} の符号化コストは、ハフマン符号を用いた情報圧縮により、負の対数尤度を用いて次のように表現することができる: $Cost_C(\mathcal{X}|\theta) = \sum_{i=1}^m Cost_C(\mathcal{X}[s_i]|\Theta) = \sum_{i=1}^m -\ln(\delta_{vu} \cdot (\delta_{uu})^{|s_i|-1} \cdot P(\mathcal{X}[s_i]|\theta_u))$. ここで、 i と $(i-1)$ 番目のセグメントはそれぞれ u と v 番目のレジームに所属するものとし、 $\mathcal{X}[s_i]$ は \mathcal{X} に含まれるセグメント s_i で構成される部分シーケンスを表す。 $P(\mathcal{X}[s_i]|\theta_u)$ は θ_u が与えられたときの $\mathcal{X}[s_i]$ の尤度とする.

結論として、提案アルゴリズムは式 (3) を最小化するように、

1: デフォルトでは $\alpha = 1.0$.

2: ここで、 \log^* は整数のユニバーサル符号長を表す: $\log^*(x) \approx \log_2(x) + \log_2 \log_2(x) + \dots$ [24].

\mathcal{X} に含まれる時系列パターンとその変化点の数 r, m を決定する. 次節では、データをコスト関数に基づき要約しながら、長期的なイベント予測を実現するための具体的なアルゴリズムについて詳述する.

5.2 概要

提案手法は、以下のアルゴリズムで構成される.

- **REGIMEGENERATION (P1):** テンソル \mathcal{X} に含まれる時系列パターンの種類と変化点を検出する. 各時系列パターンのダイナミクスをモデルパラメータ Θ として表現し、モデルパラメータ集合 $\{m, r, \mathcal{S}, \Theta, \mathcal{F}\}$ を得る.
- **FEATUREEXTRACTION (P2):** 時系列パターンの要約情報 $\{m, r, \mathcal{S}, \Theta, \mathcal{F}\}$ を用いて、多次元時系列テンソル \mathcal{X} を潜在状態テンソル \mathcal{Z} と誤差テンソル \mathcal{E} で表現する.
- **SPLITCAST (P3):** $\mathcal{X}' = \{\mathcal{Z}, \mathcal{E}\}$ のうち、あるウィンドウ $t_s : t_e$ の部分テンソル $\{\mathcal{X}'(t_s : t_e)$ からイベントの予兆となる特徴を抽出し、 l_s 先のイベントラベル $\mathcal{Y}(t_e + l_s)$ を予測する.

図 2 に提案モデルの概要を示す. テンソル \mathcal{X} が与えられたとき、提案手法は \mathcal{X} の時系列パターンの時間遷移と設備固有のパターンを捉え、それらに基づいて \mathcal{X} を $\{\mathcal{Z}, \mathcal{E}\}$ で要約する. 最終的に、得られた $\{\mathcal{Z}, \mathcal{E}\}$ から l_s ステップ先でのイベントラベルを予測し、出力する.

5.3 RegimeGeneration (P1)

時系列解析における根本的な問題は、時系列データに内在する隠された構造があるかどうかである. 我々が扱う多次元時系列テンソルは、複数の観点からの特徴を持つ. すなわち、時間ドメインの特徴と設備ドメインの特徴である. 具体的には、スマート工場から得られる時系列センサデータは、各工程の時間遷移パターンと、設備固有のパターンを持つ. そこで本研究では、与えられた時系列テンソルの根底にある構造を簡潔に要約した、多角的なパターン発見とグループ化を同時に行う.

ここで、時系列テンソルの多角的解析のためのアルゴリズムである V-Split と H-Split を提案する. V-Split は時間方向の観点からレジームを推定し、H-Split は設備ごとの特性をレジームとして表現する. これら二つのアルゴリズムを任意方向に行うことで、効率的かつ効果的に重要なパターンを多角的に発見し、レジームとして要約する. 具体的には、式 3 に基づき、以下の二つのアルゴリズムを繰り返す.

- **V-Split:** テンソル \mathcal{X} から時間遷移するパターンとその変化点を検出し 2 つのレジームに分割する. それら 2 つのレジームに対し、モデルパラメータ $\{\theta_1, \theta_2, \Delta\}$ を推定する.
- **H-Split:** テンソル \mathcal{X} に表れるある 1 つのレジームから設備ごとの特徴を抽出し、2 つのレジームに分割後、それらのレジームのモデルパラメータを推定する.

上記のアルゴリズムにより、 $r = 1, 2, \dots$ とレジーム数が増加していく. もしレジーム θ_0 を 2 つのレジーム $\{\theta_1, \theta_2\}$ に分割した際、コスト関数 (式 3) の値が大きくなれば θ_0 は最適とみなし、これ以上分割しない. 生成された全てのレジームについて同様にコスト計算を繰り返し、コストが下がらなくな

Algorithm 1 V-Split (\mathcal{X})

```

1: Input: Tensor  $\mathcal{X}$ 
2: Output:  $\{m_1, m_2, S_1, S_2, \theta_1, \theta_2\}$ 
3: Initialize models  $\theta_1, \theta_2, \Delta_{2 \times 2}$ ;
4: while improving the cost do
5:    $\{m_1, m_2, S_1, S_2\} = \text{V-Assignment}(\mathcal{X}, \theta_1, \theta_2, \Delta)$ ;
6:    $\theta_1 = \text{ModelEstimation}(S_1); \theta_2 = \text{ModelEstimation}(S_2)$ ;
7:   Update  $\Delta$  from  $S_1, S_2$ ;
8: end while
9: return  $\{m_1, m_2, S_1, S_2, \theta_1, \theta_2\}$ ;

```

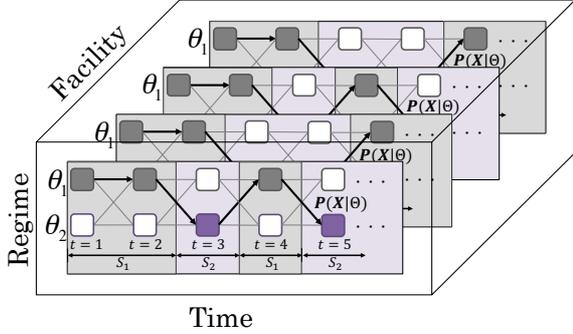


図 3: V-Assignment の概要

るまで上記の分割アルゴリズムを繰り返す。最終的に、コストが収束したときのセグメント、レジーム、モデルパラメータ $\{m, r, S, \Theta, \mathcal{F}\}$ を出力し、RegimeGeneration を終了する。

5.3.1 V-Split

多次元時系列テンソル \mathcal{X} が与えられたとき、V-Split は時間遷移の観点から 2 つのレジームを検出し、それらのモデルパラメータ $\{\theta_1, \theta_2, \Delta\}$ を推定する。高精度なモデルを生成するため、提案手法はセグメント/レジームの検出とモデルパラメータの更新を以下のように反復する。

- (Phase 1) V-Assignment: 2 つのモデルパラメータ $\{\theta_1, \theta_2, \Delta\}$ が与えられたとき、それらに基づいて 2 つのセグメントセット $\{S_1, S_2\}$ とパターンの変化点を抽出する。
- (Phase 2) ModelEstimation: 2 つのセグメントセット $\{S_1, S_2\}$ が与えられたとき、それらに基づいてモデルパラメータ $\{\theta_1, \theta_2, \Delta\}$ を更新する。

V-Split の概要をアルゴリズム 1 に示す。上記のアルゴリズムは期待値最大化法 (EM: Expectation maximization) に基づいており、それぞれのフェーズが E, M ステップに対応している。

まず、最も単純な部分問題として、テンソル \mathcal{X} と 2 つのモデルパラメータ $\{\theta_1, \theta_2, \Delta\}$ が与えられている場合を考える。V-Assignment はレジームのモデルパラメータに基づき、 \mathcal{X} のパターンの変化点の変化点を検出することができる。提案アルゴリズムの基本的な概念を説明するため、図 3 の遷移図を示す。2 つのレジーム $\{\theta_1, \theta_2\}$ の遷移を接続し、各時刻ごとに 2 つのレジームの符号化コストを比較しながら、与えられたレジーム間のパターン遷移を推定する。本アルゴリズムは動的計画法の一種であるビタビアルゴリズム [7] に基づき、符号化コスト $Cost_T(\mathcal{X}|\Theta) = -\ln P(\mathcal{X}|\Theta)$ を計算する。具体的には、尤度 $P(\mathcal{X}|\Theta)$ は次のように計算される: $P(\mathcal{X}|\Theta) = \max_{i=1,2} \{P(\mathcal{X}|\Theta)_i\}$ 。こ

Algorithm 2 H-Assignment ($\mathcal{X}, \theta_1, \theta_2, \Delta$)

```

1: Input: Tensor  $\mathcal{X}$ , model parameters  $\{\theta_1, \theta_2, \Delta\}$ 
2: Output:  $\{m_1, m_2, S_1, S_2\}$ 
3:  $m_1 = 0; m_2 = 0; S_1 = \emptyset; S_2 = \emptyset$ ;
4: for  $i = 1$  to  $w$  do
5:   if  $Cost_C(\mathcal{X}[i]|\theta_2, \Delta) > Cost_C(\mathcal{X}[i]|\theta_1, \Delta)$  then
6:      $S_1 = S_1 \cup \mathcal{X}[i]$ ;
7:      $m_1 = m_1 + |\mathcal{X}[i]|$ ;
8:   else
9:      $S_2 = S_2 \cup \mathcal{X}[i]$ ;
10:     $m_2 = m_2 + |\mathcal{X}[i]|$ ;
11:   end if
12: end for
13: return  $\{m_1, m_2, S_1, S_2\}$ ;

```

ここで $P(\mathcal{X}|\Theta)_i$ は i 番目のレジーム θ_i へ遷移する時の尤度を示す。例として、 $P(\mathcal{X}|\Theta)_1$ は次のように計算される:

$$P(\mathcal{X}|\Theta)_1 = \max_{1 \leq i \leq k_1} \{p_{1;i}(t)\}$$

$$p_{1;i}(t) = \max \begin{cases} \delta_{21} \cdot \max_u \{p_{2;u}(t-1)\} \cdot \pi_{1;i} \cdot b_{1;i}(\mathbf{x}(t)) \\ // \text{regime shift from } \theta_2 \text{ to } \theta_1 \\ \delta_{11} \cdot \max_j \{p_{1;j}(t-1)\} \cdot a_{1;j,i} \cdot b_{1;i}(\mathbf{x}(t)) \\ // \text{staying at regime } \theta_1 \end{cases}$$

ここで、 $p_{1;i}(t)$ は時刻 t でのレジーム θ_1 の潜在状態 i の最大確率を表し、 δ_{21} はレジーム θ_2 から θ_1 へのレジーム遷移確率、 $\max_u \{p_{2;u}(t-1)\}$ は前時刻 $t-1$ での θ_2 の尤もらしい潜在状態である確率、 $\pi_{1;i}$ は θ_1 の潜在状態 i の初期確率、 $b_{1;i}(\mathbf{x}(t))$ は θ_1 の潜在状態 i に対する $\mathbf{x}(t)$ の出力確率、そして $a_{1;j,i}$ は θ_1 の潜在状態 i から潜在状態 j への遷移確率を表す。ここで、時刻 $t=1$ において、レジーム θ_1 である確率は $p_{1;i}(1) = \delta_{11} \cdot \pi_{1;i} \cdot b_{1;i}(\mathbf{x}(t))$ で与えられる。

なお、モデルパラメータの推定には BaumWelch アルゴリズム [1] を用い、レジーム遷移確率 $\Delta = \{\delta_{11}, \delta_{12}, \delta_{21}, \delta_{22}\}$ を次のように計算する: $\delta_{11} = \frac{\sum_{s \in S_1} |s| - N_{12}}{\sum_{s \in S_1} |s|}$, $\delta_{12} = \frac{N_{12}}{\sum_{s \in S_1} |s|}$ 。ここで、 $\sum_{s \in S_1} |s|$ はレジーム θ_1 に所属するセグメントの長さの総和を表し、 N_{12} は θ_1 から θ_2 へのレジームの切り替え回数を示す。 δ_{21}, δ_{22} についても同様に計算できる。

5.3.2 H-Split

現実問題として、時系列テンソル \mathcal{X} はパターンの時間遷移だけでなく、設備ごとの個体差を持っている。例えば、ある 2 つの設備において、同じ部品を加工する場合であっても、工程ごとに設備間でのセンサデータの振る舞いに個体差が生じる。そこで本研究では、このような設備固有の特徴を捉え、効果的にモデル化するためのアルゴリズムである H-Split を提案する。直感的には、提案アルゴリズムは V-Split と同様に (Phase 1) レジーム分割と (Phase 2) モデル推定の 2 つのフェーズを繰り返すことで、適切なレジームとそのモデルパラメータを推定する。V-Split と異なるのは、設備固有の特徴を捉えるためのアルゴリズム H-Assignment (Phase 1) である。H-Assignment の概要をアルゴリズム 2 に示す。これまでの典型的なクラスタリングアルゴリズムと異なり、H-Assignment は効果的に設備固有のパター

ンを抽出する．具体的には，テンソル \mathcal{X} とモデルパラメータ $\{\theta_1, \theta_2\}$ が与えられたとき，提案アルゴリズムは設備 i のセグメントをあるレジーム θ に割り当てたときの符号化コストを以下のように計算し，よりコストが小さくなるレジームに設備 i のセグメントを割り当てる： $\{\mathcal{S}_\theta\} = \arg \min_{\theta \in \{\theta_1, \theta_2\}} Cost_C(\mathcal{X}[i]|\theta, \Delta)$ ．ここで， $\mathcal{X}[i] = \{s_1, s_2, \dots\}$ は設備 i のセグメント集合であり同一設備のセグメントは同一レジームとなるよう制約される．

5.4 FeatureExtraction (P2)

前節では，時系列テンソルから時系列パターンを多角的に検出するためのアルゴリズムについて述べた．次の目的は，イベント発生の長期的な予測を実現するために，時系列データからイベントの原因，あるいは予兆を示す特徴を抽出することである．一般に，高いサンプリングレートで収集されるセンサデータは多くのノイズを含み，監視するシステムが複雑であるほどその正確なふるまいをモデル化することが難しくなる．そこで本論文では，時系列パターンの特徴を利用して \mathcal{X} を抽象化し，イベントの予兆を効果的に抽出する手法を提案する．具体的には，時系列テンソル \mathcal{X} とモデルパラメータ集合 $\{m, r, \mathcal{S}, \Theta, \mathcal{F}\}$ が与えられたとき， \mathcal{X} を時系列パターンに基づく潜在状態テンソル \mathcal{Z} とモデル化した際の誤差テンソル \mathcal{E} に分割する．

今， r 個のレジーム集合 $\Theta = \{\theta_1, \dots, \theta_r\}$ が与えられたとすると，各時刻 t における，設備 i のデータ $\mathbf{x}_i(t) = \{x_{ij}(t)\}_{j=1}^d$ は Θ 内のレジームのいずれかの状態 $z_i(t)$ に変換される．ここで， $z_i(t)$ は自身と同じ状態に属する全データポイントの平均と分散の組 $\{\mu, \sigma\}$ を示す．つまり，潜在状態テンソルの次元は $\mathcal{Z} \in \mathbb{R}^{w \times 2d \times n}$ となる．続いて， Θ が与えられたときの，時刻 t における設備 i のセンサ j の計測値 $x_{ij}(t) \in \mathcal{X}$ の符号化誤差を事後確率 $p(x_{ij}(t)|\theta)$ で表現する．すなわち， \mathcal{X} 全体の符号化誤差は $\mathcal{E} \in \mathbb{R}^{w \times d \times n}$ である．最終的に，2つの特徴を結合した系列 $\mathcal{X}' \in \mathbb{R}^{w \times 3d \times n}$ を出力する．以上の処理により，入力データの情報を失うことなく，学習モデル推定の際に時系列方向の潜在的なふるまいを考慮することができる．

5.5 SPLITCAST (P3)

提案手法の最終的な目標は，与えられた時系列テンソル \mathcal{X} から， l_s ステップ先の長期的な予測を高精度に行うことである．ラベル予測タスクの典型的な方策として，近年では深層学習に基づく手法が数多く提案されている．これらは，中間層を多層にしたり，中間層のユニット数を増やすことで柔軟な学習を実現できる一方，層数やユニット数が増えるほど学習パラメータが多くなり計算時間が長くなる．また，過学習の問題もあり，問題を解決するためのテクニックが数多く存在する一方，どれも経験則に基づくものであり，人手を介した非常に細かいチューニングが必要となる．そこで提案手法 SPLITCAST は，確率モデルに基づく特徴抽出手法と深層学習手法を組み合わせ，実データから抽出された特徴的な時系列パターンを学習することで，より小さなネットワークで学習でき，過学習の問題を軽減しながら効率かつ効果的なイベント予測を実現する．

具体的には，テンソル $\mathcal{X}' = \{\mathcal{Z}, \mathcal{E}\}$ の時間発展の様子をモデ

ル化するために，Long-short term memory (LSTM)[10] を適用する．LSTM は，入力サンプルを時系列データとして扱い，高次元の非線形ダイナミクスを学習可能な深層学習モデルのひとつである．RNN の中間層のユニットをメモリユニットと呼ばれる特殊な構造に置き換えたもので，入力ゲート，出力ゲート，忘却ゲートの3種類を使用して時刻 t のユニット値 c_t とユニットの出力値 h_t を制御する．各ゲートの出力値をそれぞれ i_t, o_t, f_t とすると，LSTM の順伝播は以下の式で表される．

$$\begin{aligned} h_t &= o_t \odot \sigma(c_t) \\ o_t &= \sigma(W^{ox}x_t + W^{oh}h_{t-1} + W^{oc}c_t + b^o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \sigma(W^{hx}x_t + W^{ch}h_{t-1} + b^c) \\ i_t &= \sigma(W^{ix}x_t + W^{ih}h_{t-1} + W^{ic}c_{t-1} + b^i) \\ f_t &= \sigma(W^{fx}x_t + W^{fh}h_{t-1} + W^{fc}c_{t-1} + b^f) \end{aligned}$$

ここで， \odot は各要素の積を示し， $\sigma(\cdot)$ は活性化関数を示す．本研究では，活性化関数に *sigmoid* 関数を使用する．LSTM はメモリユニットによって与えられた入力系列の長期依存性を学習することができるため，レジームとその内部の状態遷移の過程で設備稼働状況に対して特に重要な特徴を記憶しながら，最新の設備稼働状況を要約した特徴ベクトルを抽出すると考えられる．最後に， h_t を用いて l_s ステップ先のイベント予測を行う．

本研究では，時刻 t における最新の部分シーケンスからの l_s 先故障予測を2クラス分類タスクとして扱い，出力を時刻 $t + l_s$ における故障発生確率とする．したがって，SPLITCAST の最終的な出力は， $y_{t+l_s} = \text{sigmoid}(W^{yh}h_t + b^y)$ となる．また，提案モデルが最小化すべき目的関数は Binary cross entropy (BCE) となり，モデル学習時のバッチサイズを N ，各入力サンプル i に対する SPLITCAST の出力値を \hat{y}_i とすると， $\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)$ で表される．ここで重要な点として，提案手法は比較的小さなユニット数 (= 10) かつシンプルな構造のモデルを用いながら，非常に高い性能を示している (6章を参照)．

5.5.1 理論的な分析

[補助定理 1] 提案手法の計算量はデータサイズに対し線形 ($O(wdn)$) である．

[証明 1] 各反復処理において，V-Assignment, H-Assignment および ModelEstimation は符号化コストとモデルパラメータの推定のために $O(wdnk^2)$ の計算量を要する．ここで， w は設備数， d は次元数， n は時系列の長さ， k はレジーム $\{\theta_i\}_{i=1}^r$ 中の隠れ状態の数を示す．よって，RegimeGeneration (P1) の計算量は $O(\#iter \cdot wdnk^2)$ である．ここで，反復回数 $\#iter$ と隠れ状態の個数 k は非常に小さい定数であるため，無視することができる．よって，RegimeGeneration の計算量は $O(wdn)$ である．FeatureExtraction (P2) においては，各設備，各センサ，各時刻の潜在状態とモデル化した際の誤差を出力するため，計算量は $O(wdn)$ である．最終的に，得られたモデルをユニット数 u の LSTM で学習する際，計算量は $O(u^2 \cdot wdn)$ となる．ここで，提案手法においては，複雑なニューラルネットワークを想定しておらず，ニューラルネットワークのユニット数 u は非常

に小さい定数であるため無視できる。ゆえに、提案手法の計算量は $O(wdn)$ である。

6 評価実験

本論文では、SPLITCAST の有効性を検証するため、実データを用いた実験を行った。図 1 において提案手法の具体例をすでに示した。本章では、イベントの長期的予測に対する提案手法の精度と計算時間の検証を行う。実験は 128GB のメモリ、NVIDIA TITAN V 12GB の GPU を搭載した Linux (Ubuntu 18.04 LTS) マシン上で実施した。データセットは平均値と分散値で正規化して使用している。

6.1 提案手法の予測精度

次に、与えられた時系列テンソルに対する提案手法の故障予測精度について検証する。比較手法には、一般的な 2 値予測モデルであるロジスティック回帰 (LR: Logistic regression) [1] と再帰型ニューラルネットワークモデルである RNN (Recurrent neural network) [25], GRU (Gated recurrent unit) [4], LSTM と比較した。LR では、他の再帰型モデルを推定する際にミニバッチとして与える部分シーケンスから平均値、分散値、最大値、最小値を算出し、4 次元の特徴ベクトルとしてラベル予測を行う。RNN, GRU, LSTM では、実データを入力とする。

提案手法に関して、予測ステップ数 200, ウィンドウサイズ 400, 符号化コストの重み 1.0 をデフォルトとして実験を行なっている。また、提案手法を含むすべての再帰型モデルについては、中間層のユニット数を 10, 出力層のユニット数は 5 とし、最適化アルゴリズムには Adam [13] を使用した。評価指標には Accuracy を使用し、5 分割交差検証を行なった。

使用したデータセットは、三菱重工エンジン&ターボチャージャ株式会社で 2017 年 10 月から 3 ヶ月間実際に稼働し、ベアリング・ハウジング加工を行っていた 55 の工場設備に取り付けられた、回転速度 (Speed), 稼働電圧 (Load), 設備温度 (Temp) の 3 つのセンサによって 5 秒間隔で取得されたものである。スライディングウィンドウで学習用サンプルを生成しており、設備自体が稼働していないときのサンプルは省いている。正常稼働時のサンプル数が 62983, 非常停止前のサンプル数が 1069 あり、学習に偏りが生じるため、非常停止時のサンプル数に正常稼働時のサンプル数を揃え、結果として 1069×2 サンプルを用いた実験を行なった。

予測先ステップ数を変化させたときの予測精度。図 4 は予測先のステップ数 l_s を変化させたときの精度の比較である。本実験は異なる l_s ごとにサンプルを生成し、学習と予測を行なっている。比較手法は、ランダムに予測した場合と同程度 ($Accuracy = 0.5$) の予測精度を示す一方で、SPLITCAST はいずれの条件下でも優れた性能を示している。この結果から、非常停止の要因は温度の上昇や稼働電圧の低下といった単純なものではなく、非線形性を有する複雑な事象だと考えられる。提案手法は実データに含まれる時系列パターンを考慮して各時刻のダイナミクスを捉えることができるため、他の再帰型モデル

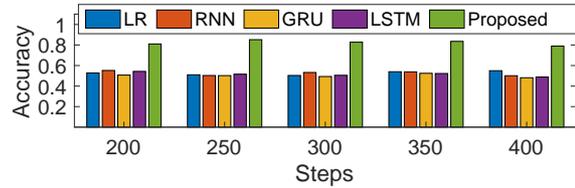


図 4: l_s ステップ先の予測精度の比較

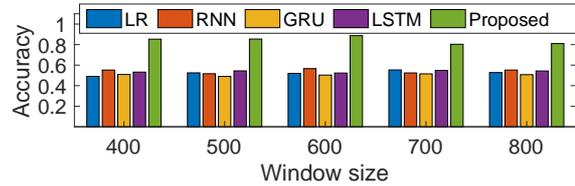


図 5: ウィンドウ幅を変化させたときの予測精度の比較

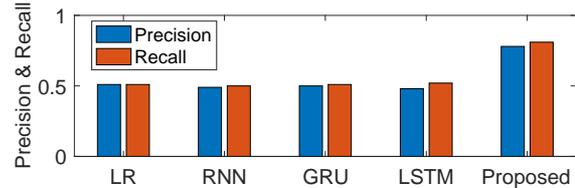


図 6: 予測結果の適合率 (Precision) と再現率 (Recall)

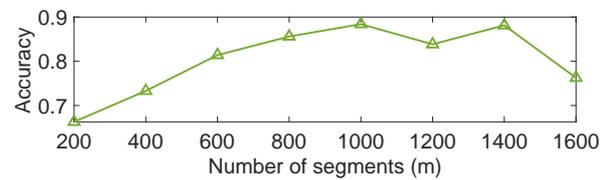


図 7: 発見セグメント数に対する予測精度の変化

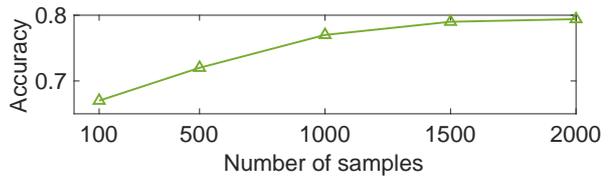


図 8: 学習サンプル数と予測精度の関係

と比べて効果的に非常停止の要因を抽出することに成功した。ウィンドウサイズを変化させたときの予測精度。図 5 は、ネットワーク学習時に使用するミニバッチのウィンドウ幅を変化させたときの予測精度の比較である。提案手法は、異なるウィンドウ幅のデータに対しても安定して高い性能を示している。予測結果の適合率と再現率。図 6 に、予測結果の適合率 (Precision) と再現率 (Recall) を示す。適合率は、予測されたイベントの合計数とそのうち正解であったイベントの合計数の割合を示す。再現率は、全てのイベントの正解値の数と予測されたイベントの中で正解した合計数の割合を示す。両者とも、精度が高い場合には 1 に近づく。提案手法はどちらの指標に対しても優れた性能を示している。

発見セグメント数に対する予測精度。図 7 に検出セグメント数 m に対する SPLITCAST の予測精度を示す。符号化コストの重みである α を 0.1 ~ 10 まで変化させながら、検出セグメント数を増減させた。図に示すように、SPLITCAST によって分割したセグメントの数によって予測精度は大きく変化する。 m が小さい場合、時系列データから十分な要約情報を得ることができず予測精度が低下する。 m が大きい場合にも同様に、要約情

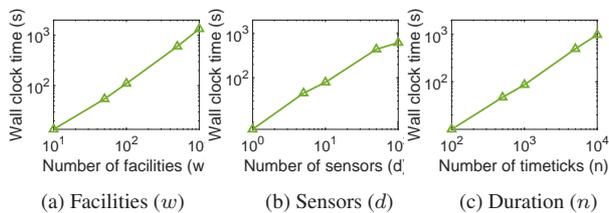


図 9: 提案手法の計算速度

報が実データに近づくため予測精度が低下する。この結果からも、時系列テンソルからのパターン検出が故障予測の精度向上に有効であると言える。本実験では、 $m = 1000$ のとき最も良い結果 ($Accuracy = 0.88$) が得られた。結論として、提案手法は比較手法に対し、平均して約 62%もの精度向上を達成した。学習サンプル数と予測精度の関係。実運用において、学習サンプルが少ない場合、十分な精度を得られない可能性がある。図 8 に学習サンプル数と予測精度の関係性を示す。提案手法は、小さいサンプル数においても比較手法より高い性能を示しており、学習サンプル数が増大するにつれより高い精度で故障イベントを予測することができる。

6.2 提案手法の計算速度

図 9 は設備数 w 、センサ数 d 、シーケンス長 n をそれぞれ変化させたときの SPLITCAST の計算コストを示す。より具体的には、入力データを時系列パターンに分割し、モデルの学習を 10 epoch 終えたときの計算時間である。SPLITCAST は与えられた時系列テンソルから効率的に時系列パターンを検出するため、すべての実験においてデータサイズに線形な計算量であり、大規模データの解析に適した手法である。

7 むすび

本論文では大規模時系列データのための予測アルゴリズムとして SPLITCAST を提案した。SPLITCAST は、与えられた時系列テンソルに含まれる特徴的なパターンやその変化点を多角的に抽出し、レジームとして要約することで、長期的なイベント予測を実現する。工場設備で得られた実データを用いて実験を行い、SPLITCAST が複雑な時系列パターンを適切にモデル化し、長期的なイベント予測能力を持つことを確認した。また、既存手法と比較して大幅な精度と性能の向上を達成した。

文 献

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.
- [2] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice Hall, Englewood Cliffs, NJ, 3rd edition, 1994.
- [3] P. Chen, S. Liu, C. Shi, B. Hooi, B. Wang, and X. Cheng. Neucast: Seasonal neural forecast of power grid time series. In *IJCAI*, pages 3315–3321, 2018.
- [4] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv e-prints*, page arXiv:1409.1259, Sep 2014.
- [5] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, pages 1555–1564, 2016.
- [6] N. Du, Y. Wang, N. He, and L. Song. Time-sensitive recommenda-

- tion from recurrent user activities. In *NIPS*, pages 3492–3500, 2015.
- [7] J. G. DAVID FORNEY. The viterbi algorithm. In *Proceedings of the IEEE*, pages 268–278, 1973.
- [8] D. Hallac, S. Vare, S. Boyd, and J. Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. In *KDD*, pages 215–223, 2017.
- [9] C. Han, K. Murao, T. Noguchi, Y. Kawata, F. Uchiyama, L. Rundo, H. Nakayama, and S. Satoh. Learning more with less: Conditional pggan-based data augmentation for brain metastases detection using highly-rough annotation on MR images. In *CIKM*, pages 119–127, 2019.
- [10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.
- [11] T. Honda, Y. Matsubara, R. Neyama, M. Abe, and Y. Sakurai. Multi-aspect mining of complex sensor sequences. In *ICDM*, 2019.
- [12] K. Kawabata, Y. Matsubara, and Y. Sakurai. Automatic sequential pattern mining in data streams. In *CIKM*, pages 1733–1742, 2019.
- [13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.
- [14] L. Li, J. McCann, N. Pollard, and C. Faloutsos. Dynammo: Mining and summarization of coevolving sequences with missing values. In *KDD*, 2009.
- [15] Y. Li, J. Wang, J. Ye, and C. K. Reddy. A multi-task learning formulation for survival analysis. In *KDD*, pages 1715–1724, 2016.
- [16] Y. Matsubara and Y. Sakurai. Regime shifts in streams: Real-time forecasting of co-evolving time sequences. In *KDD*, 2016.
- [17] Y. Matsubara, Y. Sakurai, and C. Faloutsos. Autoplait: Automatic mining of co-evolving time sequences. In *SIGMOD*, pages 193–204, 2014.
- [18] Y. Matsubara, Y. Sakurai, and C. Faloutsos. The web as a jungle: Non-linear dynamical systems for co-evolving online activities. In *WWW*, pages 721–731, 2015.
- [19] Y. Matsubara, Y. Sakurai, C. Faloutsos, T. Iwata, and M. Yoshikawa. Fast mining and forecasting of complex time-stamped events. In *KDD*, pages 271–279, 2012.
- [20] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *KDD*, pages 6–14, 2012.
- [21] H. Mei and J. Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NIPS*, pages 6757–6767, 2017.
- [22] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *IJCAI*, pages 2627–2633, 2017.
- [23] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*, pages 262–270, 2012.
- [24] J. Rissanen. A Universal Prior for Integers and Estimation by Minimum Description Length. *Ann. of Statist.*, 11(2):416–431, 1983.
- [25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [26] Y. Sakurai, S. Papadimitriou, and C. Faloutsos. Braid: Stream mining through group lag correlations. In *SIGMOD*, pages 599–610, 2005.
- [27] P. Wang, H. Wang, and W. Wang. Finding semantics in time series. In *SIGMOD Conference*, pages 385–396, 2011.
- [28] S. Wang, K. Kam, C. Xiao, S. R. Bowen, and W. A. Chaovalitwongse. An efficient time series subsequence pattern mining and prediction framework with an application to respiratory motion prediction. In *AAAI*, pages 2159–2165, 2016.
- [29] S. Xiao, J. Yan, X. Yang, H. Zha, and S. Chu. Modeling the intensity function of point process via recurrent neural networks, 2017.
- [30] R. Zhao and Q. Ji. An adversarial hierarchical hidden markov model for human pose modeling and generation. In *AAAI*, 2018.
- [31] Y. Zhou, H. Zou, R. Arghandeh, W. Gu, and C. J. Spanos. Non-parametric outliers detection in multiple time series A case study: Power grid data analysis. In *AAAI*, 2018.