

MatrixFlow システムを使って センサーデータの解析をやってみた

三島 健

技術本部

はじめに

近年、AI に対する注目は高まるばかりです。ほぼ毎日なんらかの AI に関するニュースを見かけます。私個人は異常検知または異常予知に興味があります。例えば、収穫したきゅうりが並べてあって、商品にならない異常な（と言っても形が悪いだけで味は問題ないのですが）きゅうりを自動的に抽出する AI があります。例えば、MRI の画像から病気（の可能性）があるか無いかを AI で判定したりします。例えば、鉄塔の画像から劣化具合を AI が判定しメンテナンスが必要かどうかを判定します。このように異常検知または異常予知ができると、様々な応用範囲が考えられて、世の中に貢献できそうです。このような背景から、異常検知または異常予知を AI でやってみたいと考えました。

AI のツールには、無償のものがたくさんありますし、有償のものもたくさんあります。果たしてどれを使ったら良いか、選ぶことが難しいです。これらの中で、MatrixFlow (<https://www.matrixflow.net/>)社が出している MatrixFlow という AI ツールを使う機会を得たので、これを利用してみる事にしました。MatrixFlow 社のホームページによると、「ノーコード AI」「プログラミング不要」「安価」「プロのデータサイエンティストが、現場の AI 活用をサポートいたします」などのキーワードが並んでいます。

トライする異常検知または異常予知のテーマですが、本稿ではパッケージマシンに取り付けた約 110 のセンサーからシステムダウンを予知する検討結果を中心に説明を書きます。

MatrixFlow システムとは？

MatrixFlow システム (<https://www.matrixflow.net/>) は MatrixFlow 社が出している有償のノーコード AI ツールの一つです。ホームページによると、以下のような特徴を持っています (2024/6/10 閲覧)。

1. 簡単に使える

様々な予測/分類テンプレートや、独自技術 AutoFlow が用意されているため、初心者でもマウス操作のみで簡単に AI モデルを構築できます。

2. 幅広い AI が作れる

MatrixFlow では、数値データ、テキストデータ、画像データのいずれも扱うことが可能です。

社内に複数のデータ分析ツールを導入する必要はありません。

3. 根拠が分かる

AI だからといってブラックボックス化してしまうわけではありません。要因説明機能が提供されているため、説明性の高い AI を構築することが可能です。

4. 社内で AI 活用が完結する

外注の時に必要な AI ベンダーとのコミュニケーションをなくすことが可能です。ビジネスの知識をそのまま AI 構築に活かせるため、短期間で良い AI を構築することが可能です。

5. すぐに始められる

必要なのはブラウザとデータのみ。クラウドで提供しているサービスの為、面倒な機械学習の環境構築は一切ありません。導入時の時間的、経済的コストを軽減します。

課題設定

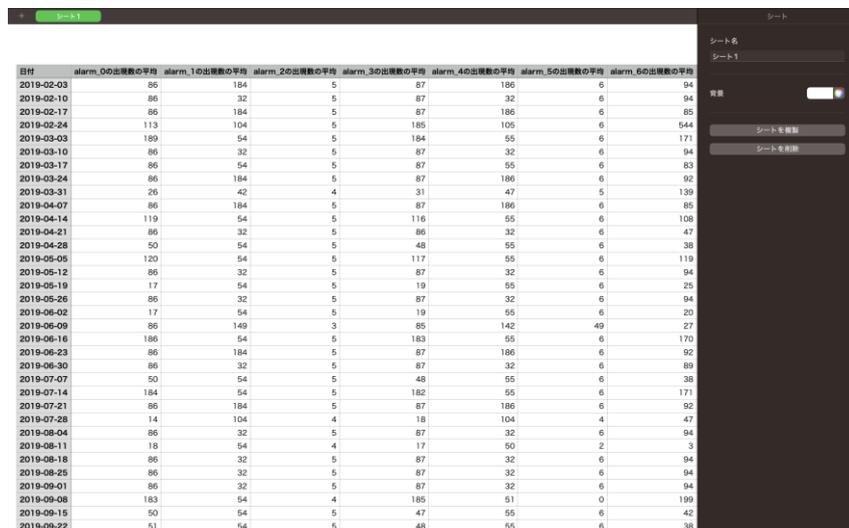
マシンに複数のセンサーを取り付けてアラームが上がった時、何月何日何時何分何秒にどのセンサーのアラームが上がった、というペアのデータを蓄積します。これらのセンサーデータを教師データとして学習したモデルを作成し、マシンダウンが起きるかどうかを予測します。もし、高い精度でマシンダウンを予測することが可能になれば、マシンダウンが起きる前に計画的にメンテナンスを実行する事ができ、マシンダウンによる損失を最小化する事ができるというメリットがあります。急に工場のラインが停止してしまうと、生産損失（商品が製造できなくなってしまう）、設備の修理費用の支払い、納期遅延（お客様からの信頼を失う可能性があります）、生産停止中の人件費の支払い（工場が停止している間も従業員には給与を支払う必要があります）、再開に伴う稼働コスト（機械の再設定や人員の再配置などの作業が必要になり無視できないコストになります）などの損失があります。例えば、自動車工場では1分間の突然の稼働停止で約 240 万円の損失が出る場合や約 550 万円もの損失が出る場合があるそうです（<https://monoist.itmedia.co.jp/mn/articles/2104/02/news013.html>）。これらの損失をなるべく小さくすることができれば、多くの企業に多くのメリットがあると思われます。

教師データ

前記課題を達成するためにはモデルが学習するための教師データが必要です。今回はこのページから（<https://ieee-dataport.org/open-access/alarm-logs-packaging-industry-alpi>）オープンデータであるセンサーデータをダウンロードしました。生データは 13.2MB あります。

シナリオ

前章でダウンロードしたデータはパッケージマシンにアラーム 0 から 114 のセンサーが取り付けてあって、何月何日何時何分何秒にどのセンサーのアラームが上がった、というペアのデータです。これを一週間毎に集計して、何月何日から何月何日までの一週間にどのアラームが何回出て、次の一週間にマシンダウンがあったかなかったか、というデータに加工しました。34KB になりました。これらのデータを教師データとして学習したモデルを作成し、ある時点で発生したアラームに対して、次の一週間以内にマシンダウンが起きるかどうかを 2 値分類問題として予測します。データは図 1 のように、あらかじめ CSV ファイルに保存しておきます。



日付	alarm_0の出現数の平均	alarm_1の出現数の平均	alarm_2の出現数の平均	alarm_3の出現数の平均	alarm_4の出現数の平均	alarm_5の出現数の平均	alarm_6の出現数の平均
2019-02-03	86	184	5	87	186	5	94
2019-02-10	86	32	5	87	32	6	94
2019-02-17	86	184	5	87	186	6	85
2019-02-24	113	104	5	185	105	6	544
2019-03-03	189	54	5	184	55	6	171
2019-03-10	86	32	5	87	32	6	94
2019-03-17	86	54	5	87	55	6	83
2019-03-24	86	184	5	87	186	6	92
2019-03-31	26	42	4	31	47	5	139
2019-04-07	86	184	5	87	186	6	85
2019-04-14	119	54	5	116	55	6	108
2019-04-21	86	32	5	86	32	6	47
2019-04-28	50	54	5	48	55	6	38
2019-05-05	120	54	5	117	55	6	119
2019-05-12	86	32	5	87	32	6	94
2019-05-19	17	54	5	19	55	6	25
2019-05-26	86	32	5	87	32	6	94
2019-06-02	17	54	5	19	55	6	20
2019-06-09	86	149	3	85	142	49	27
2019-06-16	186	54	5	183	55	6	170
2019-06-23	86	184	5	87	186	6	92
2019-06-30	86	32	5	87	32	6	89
2019-07-07	50	54	5	48	55	6	38
2019-07-14	184	54	5	182	55	6	171
2019-07-21	86	184	5	87	186	6	92
2019-07-28	14	104	4	18	104	4	47
2019-08-04	86	32	5	87	32	6	94
2019-08-11	18	54	4	17	50	2	3
2019-08-18	86	32	5	87	32	6	94
2019-08-25	86	32	5	87	32	6	94
2019-09-01	86	32	5	87	32	6	94
2019-09-08	183	54	4	185	51	0	199
2019-09-15	50	54	5	47	55	6	42
2019-09-22	51	54	5	48	55	6	38

図 1

MatrixFlow システムにデータや使うアルゴリズムをセットする

MatrixFlow システムは WEB ブラウザを使って操作します。基本的には、マウス操作だけでほとんど完結します。AI と言っても、たくさんのアルゴリズムが存在します。どれを使えば良いか？を考えて答えを出すのはなかなか大変です。どうやって選びましょうか？ MatrixFlow システムでは、図 2 のように、「テンプレートを使用して AI を構築する」と「自分で AI を構築する」の大きく分けて 2 パターンを選ぶことができます。

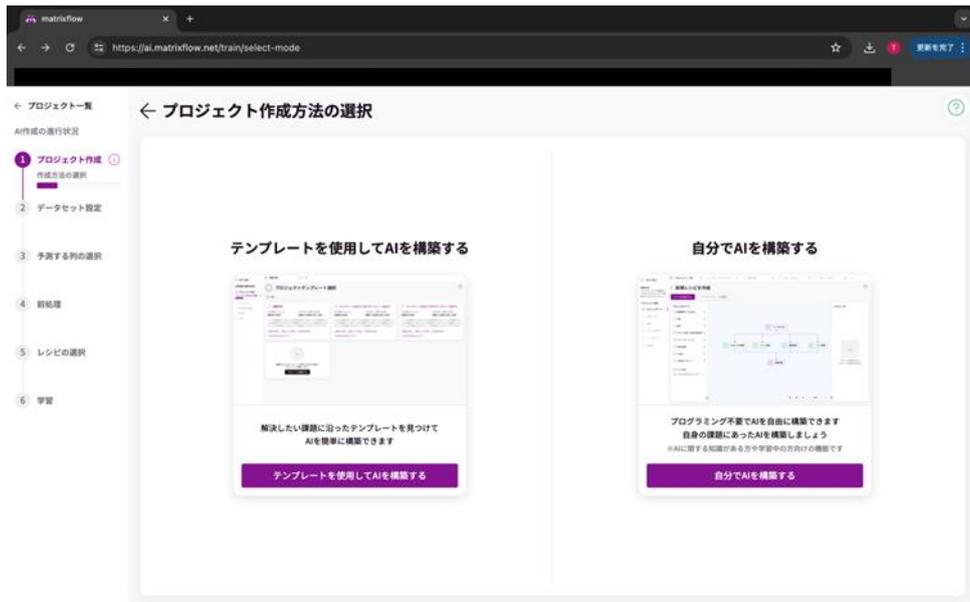


図 2

「テンプレートを使用して AI を構築する」を選ぶと、図 3 のように複数のテンプレートが表示されます。自分が解析したい目的に近いテンプレートを選べば、その目的に最適なアルゴリズムが選択できるようになっています。なんと便利です！



図 3

上級者はおそらく「自分で AI を構築する」を使うと思いますが、初級者にも便利な機能がありますので、順に説明します。まず、図4の画面から、前記 CSV ファイルを読み込みます。

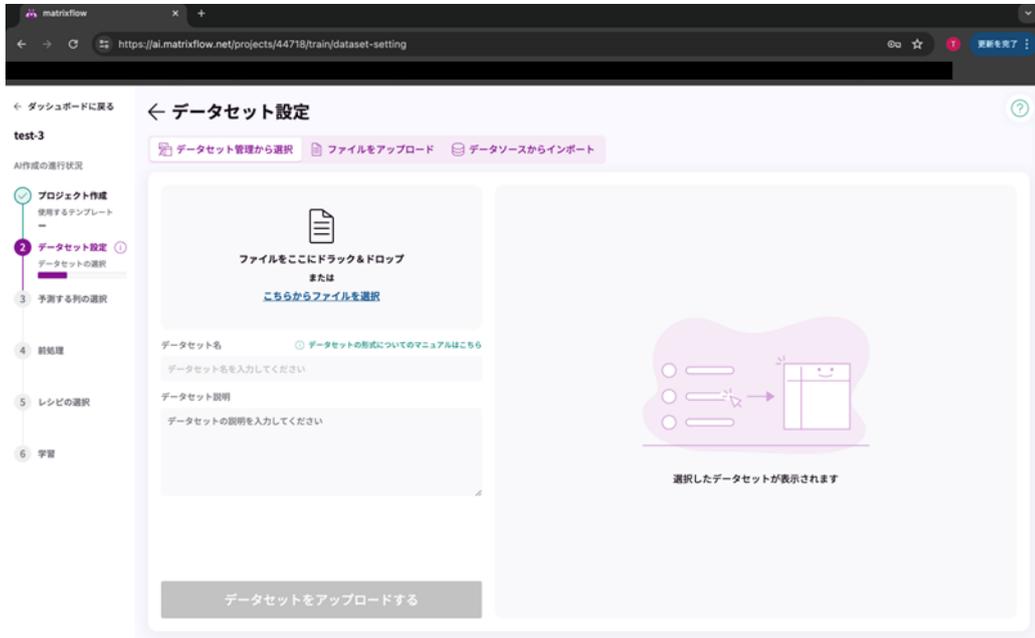


図 4

次に、図5のように列名が表示されるので、どの列を予測するのかを画面から選択します。今回は最後の列の次の週でマシンダウン（動作不良）が発生したかどうかを選択します。

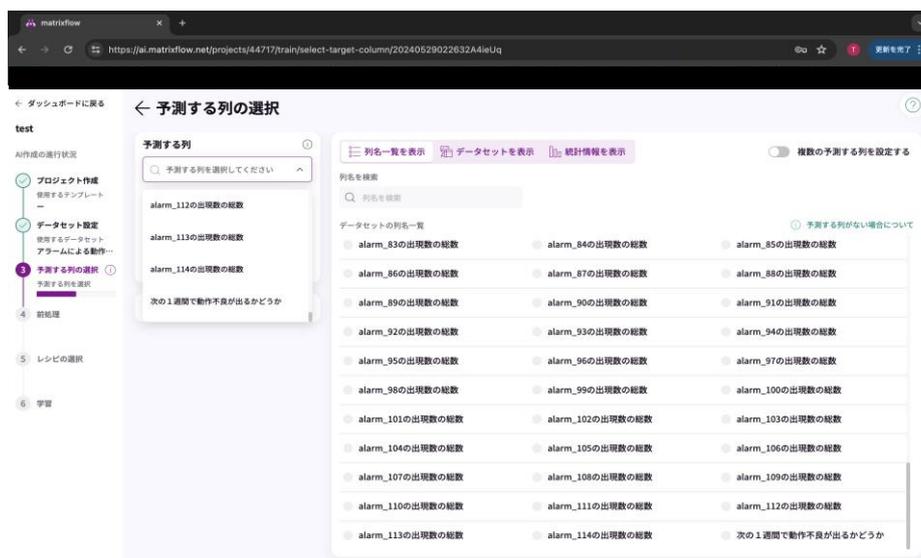


図 5

次に、どのレシピを使うかを選択します。今回は自動で AI を構築する AutoFlow という機能を選択しました。これは MatrixFlow がサポートしている AI のアルゴリズムとそのハイパーパラメータを同じ教師データに対して実行し、最も良い結果が出たものを採用する、という便利機能です。デフォルトでは 200 のアルゴリズムとパラメータをトライします。なお、元々のデータファイルからデフォルトでは 10% のデータをランダムにテストデータとして抽出し、それ以外の 90% のデータを学習用に使います。

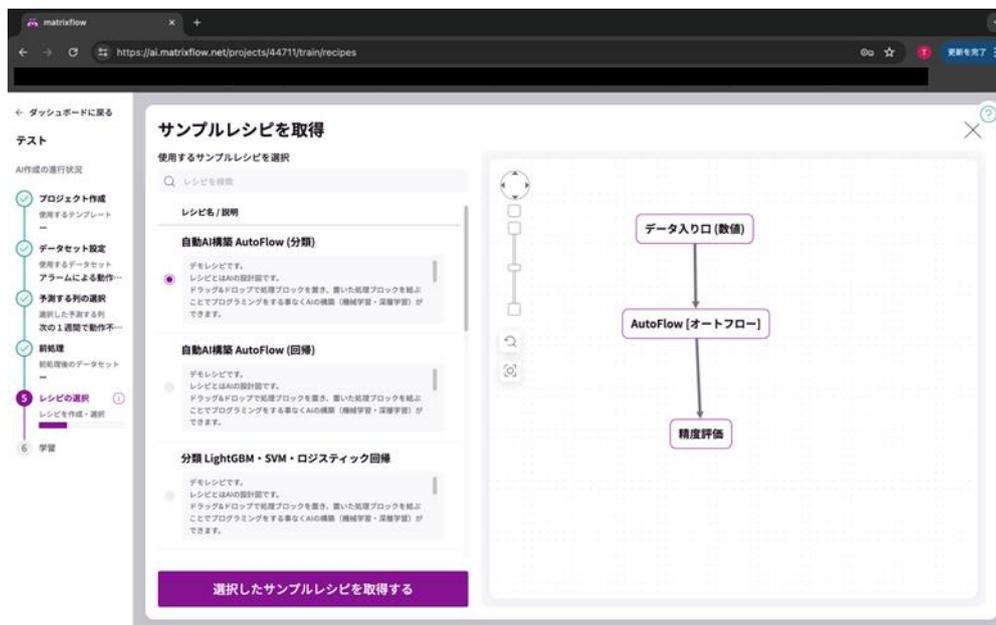


図 6

最後に学習で使う列名を選択します。今回は予測する列以外の全てのアラームのデータを使いました。

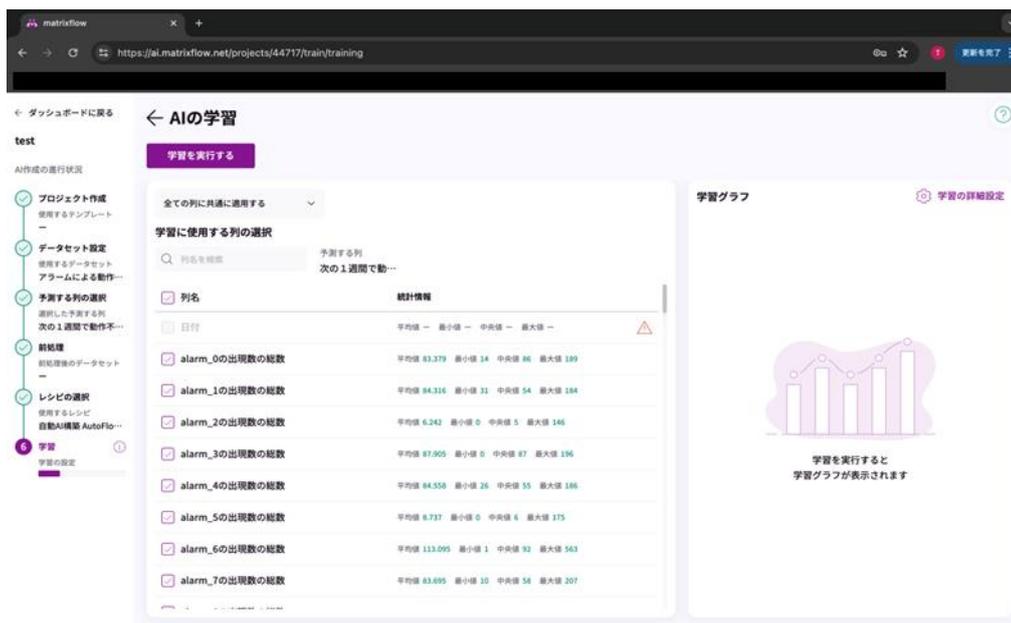


図 7

これにて学習とテストを開始します。

結果

結果が出ました。「概要」の画面では、テストデータの正解率と正解を出すために重要な列を表示します。なかなか良い精度です。

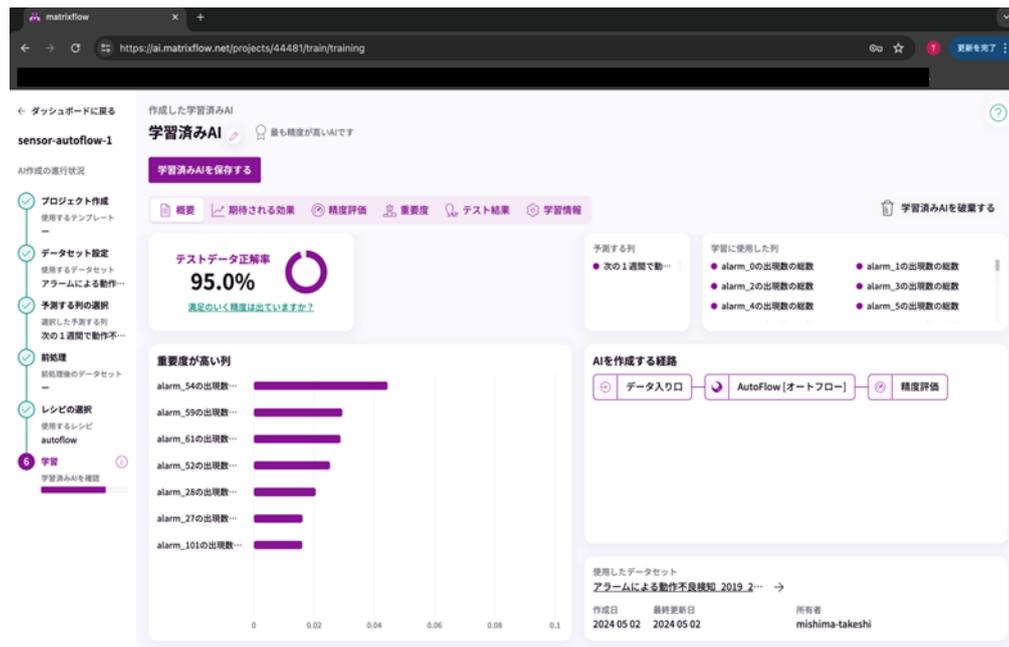


図 8

「精度評価」の画面では、左側にテストデータ正解率、適合率、再現率、F 値が表示され、右側には混同行列が表示されます。正解「0」とは次の一週間ではマシンダウンが起きていないことを表し、正解「1」は次の1週間でマシンダウンが起きたことを表します。つまり、テストデータのうち17回はマシンダウンが起きていなかったが、17回全てマシンダウンは起きないと予測したため、予測が当たったことを表します。マシンダウンが起きたケースは3回ありましたが、そのうち1回は「0」と間違って予測し、2回は「1」と予測が当たっています。正解率 100%とまではいきませんが、なかなか良い結果になったと思います。

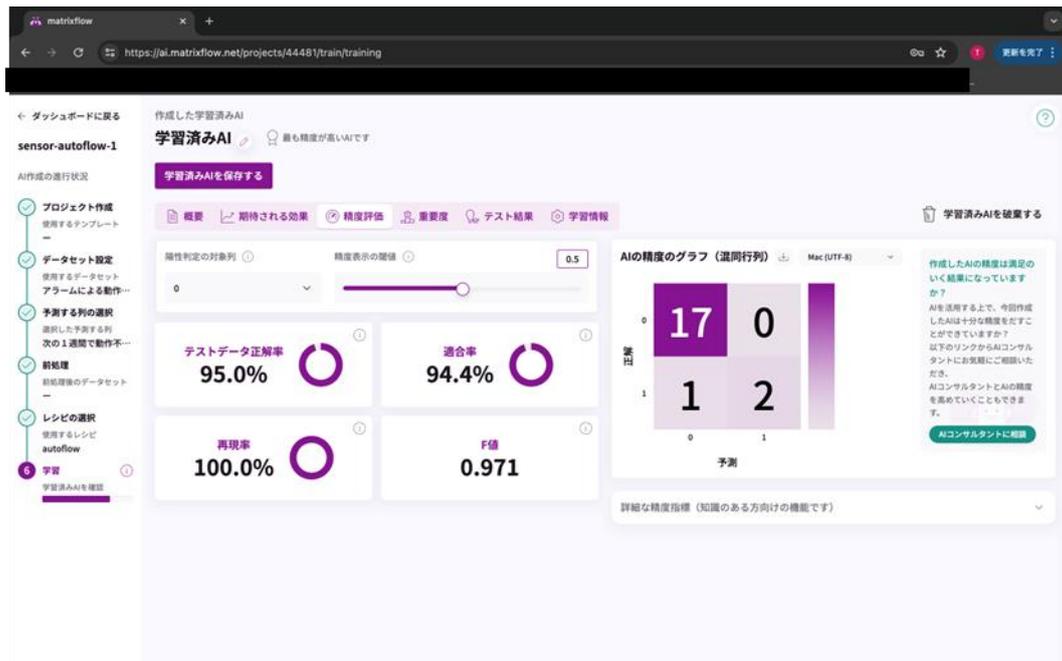


図 9

こんな便利な画面もあります。この結果を出すためにどの列のデータが重要だったかを示すものです。この図を見ると、アラーム 5 4 が最も重要なデータであり、次に 5 9、6 1 と続きます。また、右のグラフはアラーム 5 4 が 150 回を超えると、より次の週にマシンダウンが発生していることを表します。

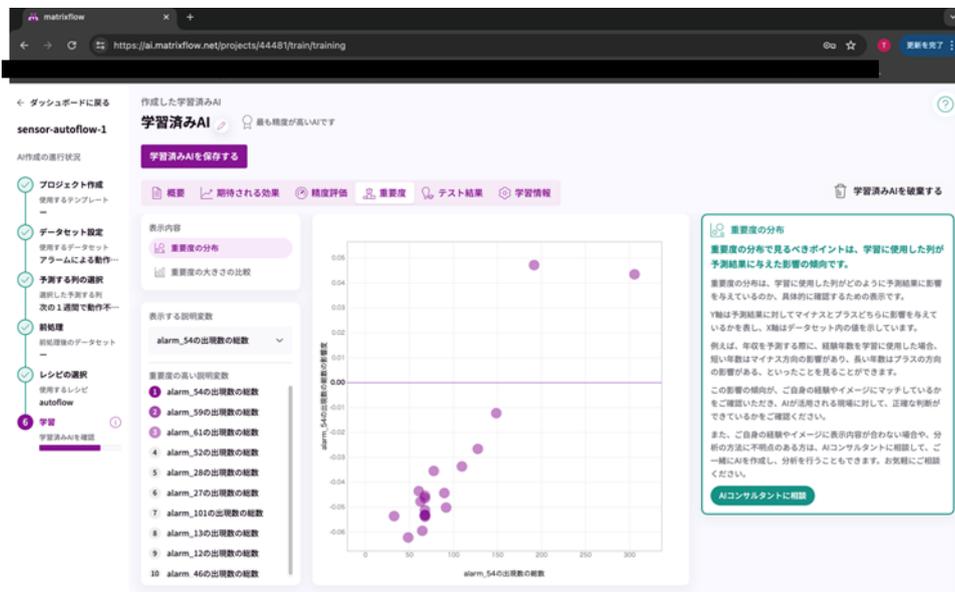


図 10

最後の図は 200 回のトライの中でどの実験がどのくらいの正解率だったかをグラフに示します。左側にはこの 200 回のうちテストデータで最も精度が高かったモデルの情報を表示

しています。つまり、アルゴリズムは「ランダムフォレスト」を使って「ブートストラップサンプリング」を使った場合が最も精度が高かったことを表しています（ランダムフォレストにはブートストラップサンプリング以外にオプションが無いようですが、他のアルゴリズムの時にはオプションも変えて実験するようです）。

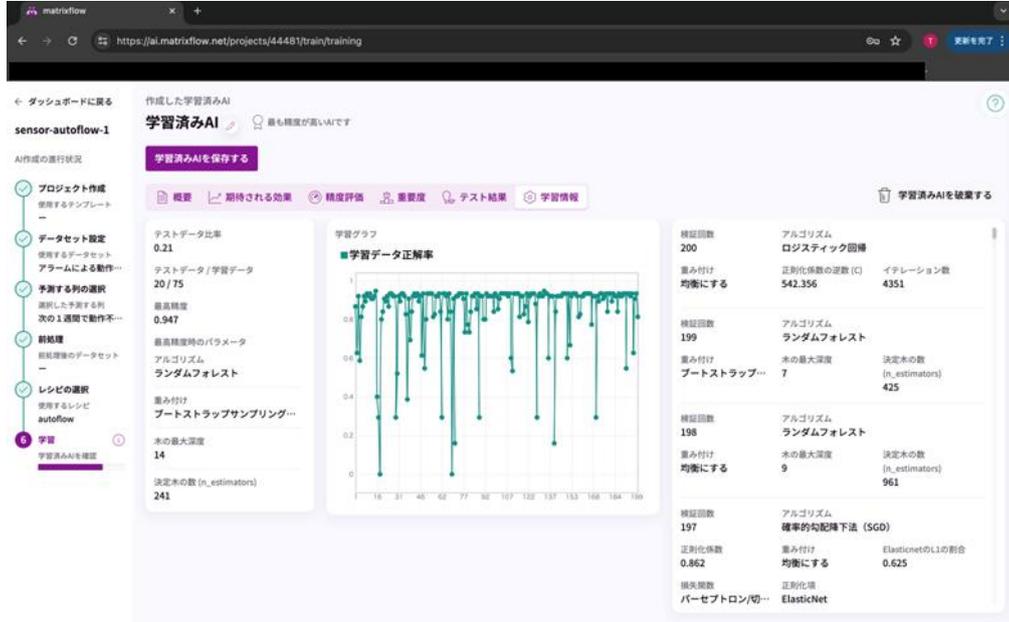


図 11

まとめ

今回はセンサーデータと MatrixFlow を使って異常予知ができるかどうかをやってみました。オープンデータを扱ったデータに対してまあまあ良さそうな結果が得られました。これからも異常検知または異常予知の検討をして極めたいと思います。

GSLetterNeo Vol.190

2024年5月20日発行

発行者 株式会社 SRA 技術本部 先端技術研究室

編集者 熊澤努 方学芬

バックナンバー <https://www.sra.co.jp/public/sra/gsletter/>

お問い合わせ gsneo@sra.co.jp

 **株式会社SRA**

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。



夢を。Yawaraka Innovation
やわらかいのべーしょん