

AIの活用を目的としたIoTデータ分析基盤の構築と運用

常三島技術部門
情報システムグループ

辻 明典 (TSUJI Akinori)

1. はじめに

AIの技術革新が進み、さまざまな分野においてAIが活用され始めている。その中でもデータ分析はAIの活用に欠かせない技術の一つであり、データサイエンスに対応できるデジタル人材の育成においても重要となる基本技術である^[1]。データ分析基盤は、データ収集、データ加工・統合、データ分析モデルの構築・運用およびデータ可視化を一貫して行うための技術的な基盤である。本報告では、AIの活用を目的としたIoTデータ分析基盤の構築について、概要から具体的な実装手順、運用事例について幅広く説明することを目的とする。

2. IoTデータ分析基盤の概要

IoTデータ分析基盤は、センサデバイスから収集したデータを蓄積、加工、分析、可視化の一連の流れを実行するシステムである。AIを最大限活用するには、データ分析基盤をどのように構築して運用するかが重要である。ここでは、IoTデータ分析基盤の構成要素および課題について述べる。

2. 1 IoTデータ分析基盤の構成要素

IoTデバイスは、工場の生産現場やオフィスビル、商業施設など、さまざまな人やモノ、場所に設置され、温度や湿度、位置情報、振動などのデータ収集を行う小型のセンシングデバイスである。データを収集するだけでなく、データ分析を行い活用することで、はじめてデータに価値が与えられる。たとえば、工作設備

の故障予測や生産性の向上、人の行動分析や制御の自動化など、さまざまな課題の解決が期待できる。IoTデータ分析基盤は、このようなデータを効率的に分析し、データ活用を促すための技術的な基盤である。IoTデータ分析基盤の基本構成を図1に示す。IoTデータ分析基盤は大きく分けて、次の5つの機能により構成される。

- ・データ収集：センサデバイスからデータを収集する機能
- ・データ蓄積：収集したデータを一元的に管理・共有する機能
- ・データ加工・結合：プロジェクトの利用に最適なデータに変換し、データベースとして保存する機能
- ・データ分析：プロジェクトごとに必要なデータを抽出して、さまざまな分析手法を適用してデータ分析を行う機能
- ・データ可視化：データ分析した結果をわかりやすく視覚化して表示する機能

それぞれの機能について簡単に説明すると次のとおりである。データ収集では、IoTデバイスとデータ分析基盤を接続するための通信プロトコルの選択、データフォーマットの変換などの処理を行う。データ蓄積では、データを安全に保管するためのRAID機能やバックアップ機能などのストレージ冗長化やユーザとのデータ共有機能などを提供する。データ加工・結合では、収集したデータの前処理、データベースへの保存、データの検索、参照などを行う。データ分析では、統計解析や機械学習、深層学習などの人工知能による解析手法を用いてデータの傾向やパターンの抽出を行う。データ可視化では、分析結果を折れ線グラフや棒グラフ、散布図など、データの特徴に応じた形式でデータの見える化を行う。

2. 2 IoTデータ分析基盤の課題

IoTデータ分析基盤の導入によって、データ

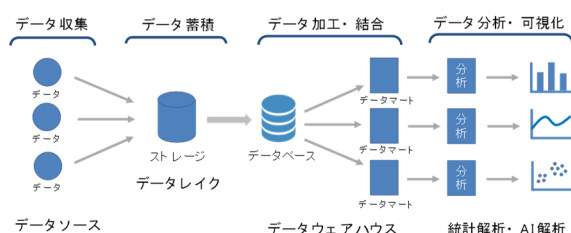


図1 データ分析基盤の基本構成

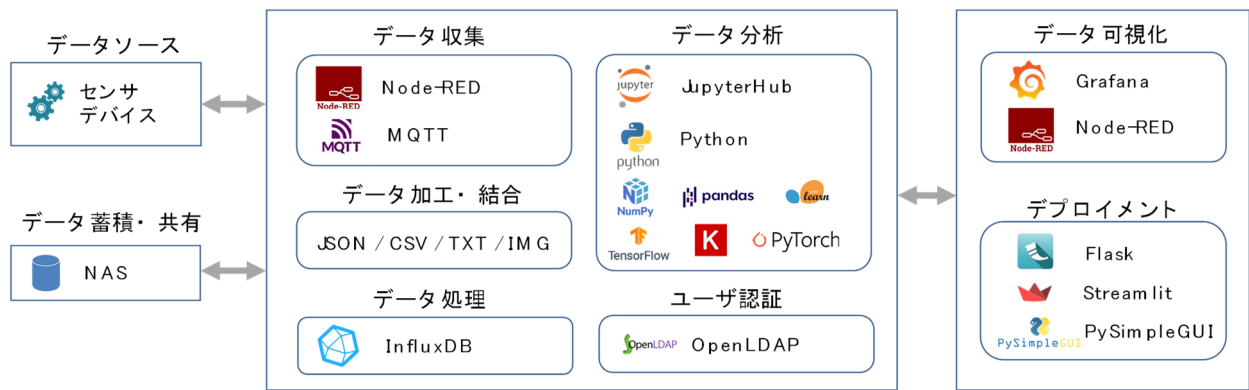


図2 AIの活用を目的としたIoTデータ分析基盤の構成

に基づく意思決定を正確にでき、また新たな価値の創出や作業の効率化をはじめとした様々なメリットが得られる。それらを実現するには、データ収集、蓄積、加工、分析、および可視化の各機能に適したツールやサービスの選定が特に重要である。これらの機能の一部またはすべては、クラウドサービスとして提供される機会が多くなり、導入に係る敷居が低くなっている。一方で、クラウドサービスの利用は、従量課金やサブスクリプション契約のため、ユーザ数が多い場合やデータ量が急激に増加するなどにより、インフラコストの増大や定常的なコスト管理が困難となることがある。また、提供するサービスの運用がクラウドサービスの稼働に影響を受けるクラウドサービス依存の課題があることが知られている。

そこで本研究では、無償のオープンソースアプリケーションを活用して、数十人規模のユーザの利用を想定したIoTデータ分析基盤をオンプレミスにて構築を行った。オンプレミスによる内製化は、サーバ構築に係る時間や手間、運用に係るコストがかかるものの、外部クラウドサービスに依存しないことは基より、サーバ構築に係る技術の蓄積を図ることができ、技術継承にも繋げることができる。また、データを外部に出さないため、データの安全性や機密性も同時に確保ができ、さらには、最新の技術導入や試験的な実証実験PoCについても迅速に対応できるなどの利点がある。

3. AIを活用したIoTデータ分析基盤の構築

AIを活用したIoTデータ分析基盤の構築に

際して、各機能に応じたサービスの検討を行った。図2に、構築したオンプレミスのIoTデータ分析基盤の構成について示す。本研究では、データ分析基盤のすべての機能をサーバ上に実装し、ユーザのコンピュータ環境に依存することなく、ブラウザのみでデータ分析が行える開発環境の構築を目標とした。ここでは、本データ分析基盤におけるデータ収集、蓄積、加工・結合、分析および可視化、これら5つの機能の詳細について述べる。

3. 1 データ収集

データ収集において、センサからのデータ受信にはMQTTプロトコルを採用した^[2]。MQTT通信では、非同期にデータを受信できるため、センサデバイスはサーバ名とトピック名を指定するのみで良くファームウェアの開発が容易になるという利点がある。また、MQTTブローカによって、センサごとに割り当てたトピック名のみでデータの識別ができるため、IPアドレスやホスト名、MACアドレスなどのリソース管理が必要なくなる。MQTTブローカの起動、データの受信処理にNode-REDを用いた。Node-REDは、ノーコード/ローコードで開発ができ、さまざまな機能ノードを繋ぎあわせることで視覚的にプログラムができる開発環境である^[3]。

3. 2 データ蓄積

データ蓄積にはネットワークに対応した市販のNAS (Network Attached Storage) を用いた。サーバとは別にNASを設置して、サーバからNASのディスクをマウントして使用する。

NASを使用するのは、ホットスワップ機能（故障時に電源を入れたままディスクの入れ替え可能）を有し、RAID構成によりディスクが冗長化されるため、サーバ上にデータを保存するよりも安全にデータを保管できるためである。MQTTブローカにおいて、センサより受信した生データは前処理等を行う前にこのNASに保存する。また、一般ユーザに対して、NASに保存されたデータにアクセスできる権限を付与することで、指定のディレクトリ下のファイルの共有が可能となる。これにより、ユーザ間のデータ受け渡しに係る時間の短縮やデータ紛失などのリスクも低減できる。

3. 3 データ加工・結合

NASに保存されたセンサの生データは、JSON、CSVなどのテキストや画像、映像といったデータが含まれる。すべての生データをNAS上に保存した上で、データを受信するごとに、サーバにセンサに応じたデータベースを作成する。ここで、データベースには、時刻情報を主キーとしたデータを扱うInfluxDBを用いた^[4]。データベース上でのコマンド操作により、データの参照・抽出・結合・挿入・削除などが実行できる。プロジェクトごとに収集したデータをひとまとめにしてデータマートの作成を行う。このデータマートを基本として、次のデータ分析および可視化を行った。

3. 4 データ分析

データ分析の基盤には、PythonをベースとしたJupyterhubを用いた^[5]。Jupyterhubは、ウェブインタフェースを備えており、ユーザはブラウザを準備するだけでOSに依存することなくデータ分析が行える。データの基本処理にはNumPy, Ppandas, Matplotlib, AIに使用される統計解析にScikit-learn, Seaborn, 深層学習用にTensor Flow, Keras, PyTorchをはじめとして、Pythonで扱えるライブラリが使用できる。さらに、Jupyterhubで作成したプログラムをウェブアプリとしてデプロイメントするために、Flask, Streamlit, PySimpleWebのライブラリを用いた。これらライブラリを使用することで、元となるPythonプログラムに大きな修正の必要がなく、ウェブ上で動作するアプリの作成

および実行ができる。Jupyterhubのユーザ管理にはOpenLDAPを用いた^[6]。OpenLDAPサーバによるユーザ認証によって、効率的に多くのユーザの運用が行える。

3. 5 データ可視化

データの可視化には、データベースからリアルタイムで取得した情報を可視化できるGrafanaを用いた。Grafanaはデータ分析およびインタラクティブな視覚化を可能にするマルチプラットフォームで動作するウェブアプリである^[7]。InfluxDBやMySQLなどのデータソースに接続することで、ウェブブラウザ上でチャートやグラフ、アラート生成などをダッシュボード上で行える。また、GrafanaのAPIトークン生成機能を使用して、登録されたデータソースにAPI経由にて他システムとのデータ連携ができる。また、簡易にデータの可視化を行うには、Node-REDのダッシュボードノードを用いてノーコードによるウェブアプリの開発も可能である。

4. AIを活用したデータ分析基盤の運用事例

本研究において構築したIoTデータ分析基盤の運用事例として、水産養殖場に設置した水質を計測する複数のセンサのデータ収集を行い、AIによるデータ分析および可視化を行った結果を次に示す。

4. 1 センサデバイス

水産養殖場に水質を計測するセンサ（水温、pH、塩分濃度、濁度、溶存酸素）を設置した。市販の水質計の多くはスタンドアロンで動作する計測器である。そこで、無線Wi-Fiに接続できるコントローラを開発して、無線による

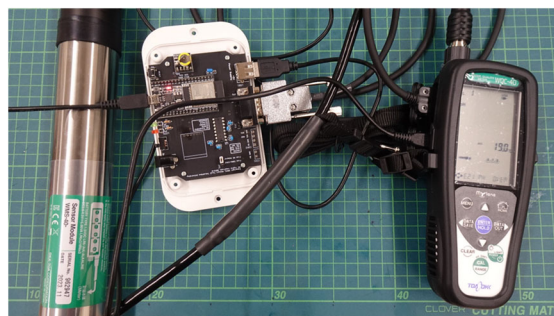


図3 水質計と開発した無線コントローラ

データ収集を可能とした。図3に、水質計と開発した無線コントローラを示す。1分ごとにセンサよりデータを取得し、無線APを通じて、サーバへデータの送信を行った。一般的なセンサデバイスや市販のセンサは、無線対応したものが少なく、有線シリアル通信規格RS-232C, RS-485, Modbus, USB, LANなどのポートを有することが多い。この場合、これらの通信ポートをコンバータによりさまざまな無線通信プロトコルに変換をし、センサを無線化してデータ通信を行う必要がある。

4.2 データ収集

データ収集はNode-REDのフローにより実装した。図4に、ノーコードでプログラムしたNode-REDのフローを示す。Node-REDでは、この図のように機能ノードを接続するだけで、プログラムに相当するデータ処理ができる。ここでは、MQTTの機能ノードを用いて、MQTTブローカの起動を行った。センサからのデータはMQTT-INノードのトピック名により

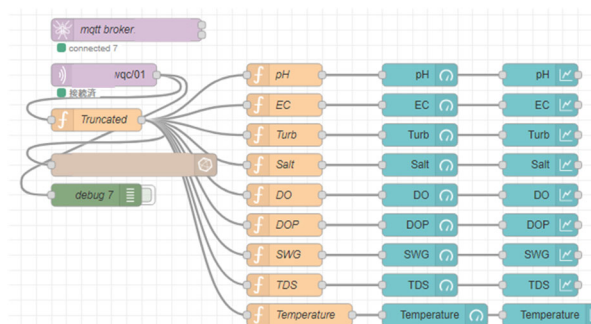


図4 ノーコードによるデータ収集，前処理，データ変換，データベース記録

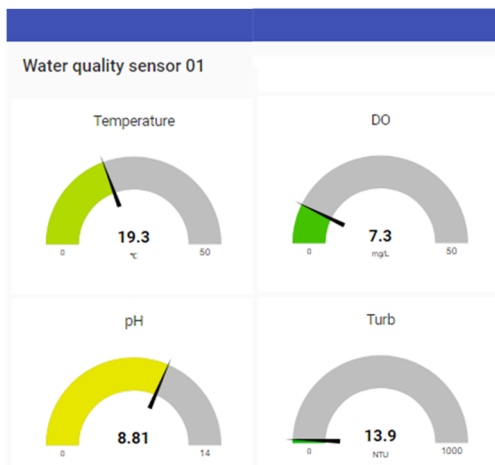


図5 Node-RED のダッシュボードノードによるデータ可視化

処理され、データをJSON形式に変換した上でInfluxDBデータベースに保存した。さらに図5のように各センサのデータをゲージ表示として可視化するために、Node-REDのダッシュボードノードによりノーコードによる実装を行った。このデータ可視化サイトは、ブラウザの機能によりウェブアプリとして実行できるため、パソコンやスマホなどOSに依存せず閲覧ができる。

4.3 データ加工・結合

MQTT入力を受信したセンサのデータは浮動小数点形式のデータである。そのため、各センサの値について小数点以下2桁, 1桁など、センサの測定精度にあわせてデータを整形する必要がある。データ整形処理について、図6のようにNode-REDのFunctionノードを用いてプログラムを行った。続けて、このセンサのデータについて、センサに付与された時刻データ(YYYY/MM/DD HH:MM:SS)に基づいて、各センサのデータ列とともにInfluxDBに保存した。データベース操作についても、InfluxDBノードを用いて、ノーコードでデータ挿入ができる。このようにNode-REDを用いることで、データの前処理をローコードで実装したり、データベースの操作などをノーコードで実行できるため、視覚的に理解しやすく保守性の高いプログラム開発が行える。

4.4 データ分析

データベースに保存されたセンサデータ(時刻, 水温, pH, 塩分濃度, 濁度, 溶存酸

```

function ノードを編集
  削除 中止 完了
  プロパティ
  名前 Truncated
  設定 初期化処理 コード 終了処理
  1 msg.payload.pH = Math.round(msg.payload.pH * 100) / 100;
  2 msg.payload.EC = Math.round(msg.payload.EC * 100) / 100;
  3 msg.payload.Turb = Math.round(msg.payload.Turb * 10) / 10;
  4 msg.payload.Salt = Math.round(msg.payload.Salt * 100) / 100;
  5 msg.payload.TDS = Math.round(msg.payload.TDS * 10) / 10;
  6 msg.payload.SWG = Math.round(msg.payload.SWG * 10) / 10;
  7 msg.payload.DO = Math.round(msg.payload.DO * 10) / 10;
  8 msg.payload.DOP = Math.round(msg.payload.DOP * 10) / 10;
  9 msg.payload.Temperature = Math.round(msg.payload.Temperature
  10
  11 return msg;
  12
  
```

図6 Node-RED の Function ノードによるセンサデータの整形

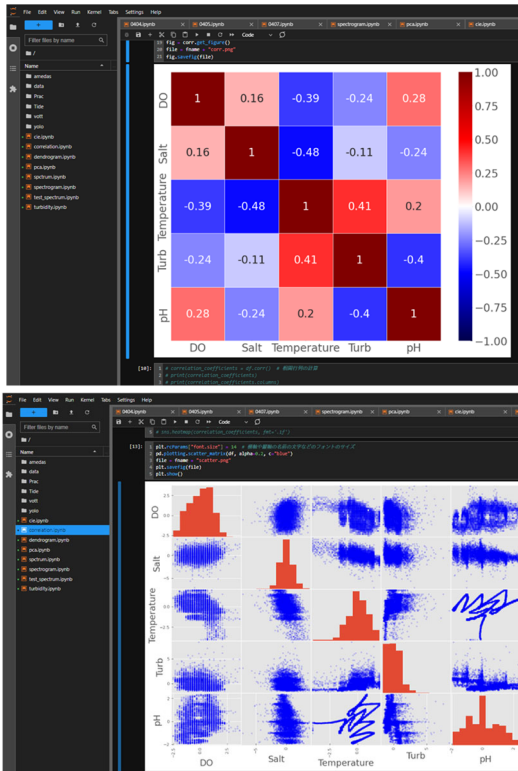


図7 データ分析結果（多変量データの相関分析）

素)を、Jupyterhub上のiPythonにおいてデータ列の読み込みを行った。iPythonは、Pythonを対話的に実行するためのシェル環境であり、特別な環境を用意せずともウェブ上でプログラムの開発及び実行ができる。図7は、データ分析に多変量解析を適用した結果である。ここでは、統計解析のライブラリSeabornを用いて分析結果をグラフ表示した。データ分析では、分析結果の視覚化が非常に重要であるが、既存ライブラリの活用によって比較的短いコードで視覚化が行えることを確認した。

4. 5 データ可視化

Jupyterhub環境で、データベースのデータを読み込んだ後、前処理として外れ値や異常値、不正値などを抽出し、データクレンジングを行った。その後、データ可視化のために、Grafanaにデータを取り込み、ウェブ上に各センサのデータを表示した。図8に各センサのデータを表示した結果を示す。Grafanaで生成できるグラフの種類、データの表示期間、アラート設定、およびダッシュボード画面やデータ共有機能について、それぞれの検証を行い、良好に動作することを確認した。



図8 Grafanaによるセンサの時系列データの可視化（水温、pH、塩分濃度、濁度、溶存酸素）

5. まとめ

本研究では、AIの活用を目的としたIoTデータ分析基盤の構築を行った。ウェブ上でデータ分析できる仕組みを実装することで、パソコンがあれば、誰でも何処でも簡単にデータにアクセスをして、データ分析の課題に取り組むことができる。さらには、分析結果や計測データのユーザ間での共有も容易にできるようになる。AIやデータサイエンスの教育・研究のみならず、AIの社会実装を実現する上においても自由度の高いデータ分析基盤の運用は非常に重要な位置づけにある。今後は、教育・研究環境への導入や生産現場におけるDX化の推進を目標として、データ分析基盤の実践的な活用を進めることとする。

謝辞

本研究はJSPS科研費23H05234の助成を受けて実施しました。ここに謝意を表します。

参考文献

- [1] 経済産業省, 2023年版ものづくり白書
- [2] MQTT, <https://mqtt.org/>
- [3] Node-RED, <https://nodered.org/>
- [4] InfluxDB, <https://www.influxdata.com/>
- [5] JupyterHub, <https://jupyter.org/>
- [6] OpenLDAP, <https://www.openldap.org/>
- [7] Grafana, <https://www.grafana.com/>