

AIを用いた設備電力データの可視化・分析ツールの構築に向けて

発表主旨

本研究では、AIを用いた電力データの可視化・分析手法を開発し、ユーザーによる現場改善を支援するツールを提案する。多変量の電力データを整理・分析し、得られた気づきをユーザーにフィードバックし、改善や最適化に向けた具体的な提案を行うことを目的としている。発表の要点は以下の通りである。

・生成AIの活用

開発に生成AIを活用することで、AIやデータ分析の知識が十分でない状態からでも、比較的短期間で実用的なツールを構築できることを示す。

・データの可視化

matplotlib等のライブラリは、エクセルでのグラフ作成と比較して、特に多量のデータを扱う際に、メリットが多いことを示す。

・種々の分析手法

k-means や階層的クラスタリングによるデータの可視化

isolation forest による外れ値検知

周期性の抽出による長期的なパターン把握

・ツール化

ユーザーインタフェースを備え、分析結果の提示に加えて具体的な改善提案を提示する。

・セキュリティ対応

データ活用に伴う情報管理のためのセキュリティガイドラインについて触れる。

発表者紹介

氏 名	木村 耕太 主任コンサルタント コンサルティング統括事業部 生産・業務改革コンサルティング部
専門分野	現場改善、設計・開発力の強化、原価管理、5S、IEによる現場改善、生産管理
コンサルティング歴	産業機械、自動車部品、素材、食品製造業などを中心に多数の企業の生産革新や原価管理を支援

AIを用いた設備電力データの 可視化・分析ツールの構築に向けて

主任コンサルタント 木村 耕太



一般社団法人 中部産業連盟

AIを用いた設備電力データの可視化・分析ツールの構築

導入・背景

設備の消費電力の時系列データを使用

- ・省エネの高いニーズ
- ・比較的、取得しやすい



AIを用いた可視化・分析

- ・人手では捉えにくいパターンや異常を発見し、改善や最適化に役立つ視点を得ることが可能かどうか、探索

生成AI活用による開発の加速

知識が少ない状態から、短期間で開発が可能

•知識不足からのスタートを補完

AIや分析の専門知識が少なくても、生成AIへ質問することで、実装方法やコード例が提示され、学びながら開発を進められる。

•短期間で成果物を構築

従来なら時間を要するデータ処理や可視化について、生成AIの支援により、数週間程度で形にできた。

•試行錯誤の効率化

プログラムのエラー修正や改良もAIが行ってくれるため、開発サイクルを高速に回せる。



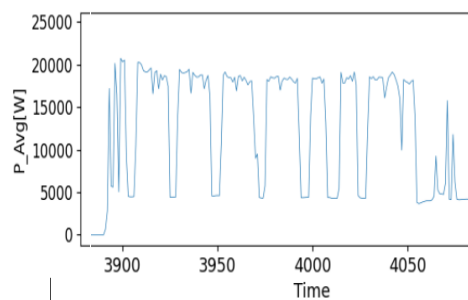
3

matplotlibによるデータの可視化

エクセルでのグラフ作成との比較

観点	Excel グラフ	Matplotlib (Python)
お手軽	直感操作で即作図	コードを書く必要あり
自動化	VBAで可能	ループで数百図を量産
データ連携	表ベースで軽～中規模は簡単	pandas/Numpy/SQL/CSV/Parquet/と直結
大規模データ	行数・応答性に限界	数十万～千万点でも工夫次第で処理可能
カスタマイズ自由度	凡例/軸/注釈などはGUI範囲で制約	目盛・補助線・投影・複合軸・等高線・カラーマップ等、細部まで可能
配布・権限	Excelで配布	画像/PDF配布

プレス機の消費電力



4

matplotlibによるデータの可視化

PythonやAIとの親和性

- **データ処理～AI分析～可視化がワンフロー**
→ 切り貼り不要、属人化を防止。
- **モデルの挙動をグラフで直感的に検証**
→ 学習曲線や予測分布を可視化して判断。
- **AIとノーコード連携可能**
→ ChatGPTなどからコードを自動生成し、即実行できる。
- **再現性と自動化の高さ**
→ AIの結果を含む報告書を定期的に自動生成できる。

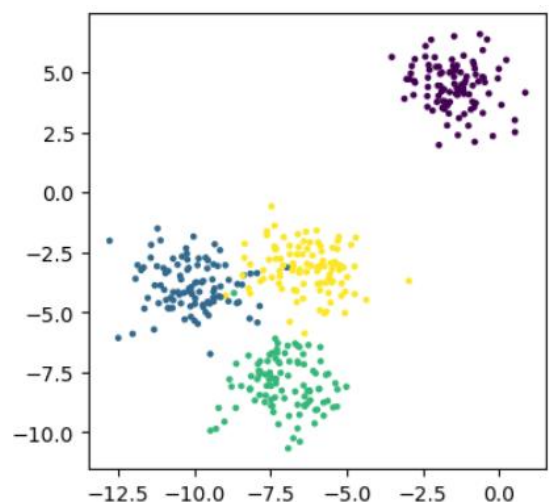
5

分析方法 ① クラスタリング

k-means法

- **データを「似たもの同士」でグループ分けする方法。**

1. グループ数 (K) を決める : 4グループに分けたい → $K=4$
2. 仮の中心点を置くデータの中に「このあたりが中心かな?」という点をランダムに置く。
3. それぞれのデータを近い中心に割り当てる → 「どの中心に一番近いか」で振り分け。
4. 中心を計算し直す → グループに属した点の“平均”を取り、それを新しい中心とする。



6

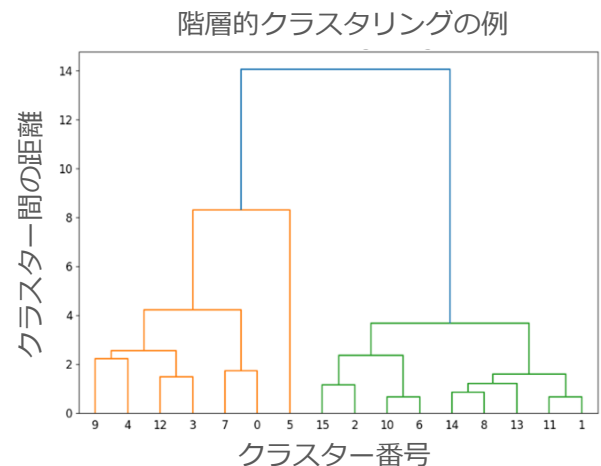
分析方法 ① クラスタリング

階層的クラスタリング

• データを「木（ツリー）」のようにまとめていく方法

（代表的な「凝集型クラスタリング」の場合）

1. 全データを一人ずつバラバラにする→ 最初は「N個のグループ（クラスター）」
2. 一番近いもの同士をくっつける→ 2点がまとまり、クラスターになる
3. 次に近いクラスター同士をくっつける→ だんだん大きなクラスターにまとめていく



7

分析方法 ② 外れ値検出

isolation forestによる外れ値検知

• 外れ値（異常値）を見つけるための機械学習アルゴリズム
名前の通り、「孤立させる」ことで外れ値を検知

1. データをランダムに分割する

→ 例えば「温度が50℃より高い？低い？」というルールで枝分かれ

2. 分割を繰り返していく

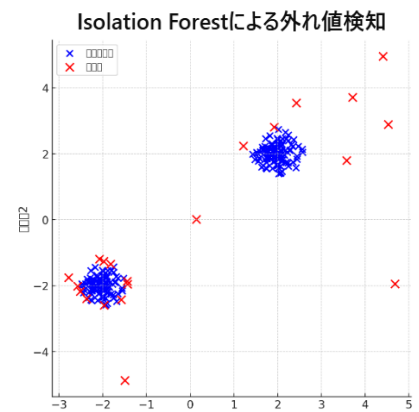
→ 木（decision tree）のように、データが分類されていく

3. 外れ値はすぐ孤立する

- 普通のデータ：他と似ているので何度も枝分かれしないと分離できない
- 外れ値：他と違うので、数回の分割ですぐに孤立する。

4. 孤立の速さをスコア化

→ すぐ孤立したデータほど「外れ値度が高い」と判断する。



8

分析方法 ③周期性の発見

周期性の発見による長期傾向の抽出

「毎日、何時ごろ消費電力が高くなる」「曜日によって故障率がちがう」など、一定のリズムで繰り返す“波”をとらえる

・フーリエ変換やスペクトル解析

→ 周波数成分を調べ、どんな周期（例：7日周期、12カ月周期）が強いかを定量的に把握。

・自己相関（ACF: Autocorrelation Function）

→ 「何日／何か月ずらすとデータ同士が似ているか？」を調べて、繰り返し周期を発見。



9

ツール化とユーザーインターフェース

操作画面のイメージ

--- 1. CSV File Loading ---

ファイル名:

スキップ行...

CSVを読み込む

--- 2. Analysis Selection ---

☐ 時系列グラフ

☐ クラスタリング

☐ 外れ値検出

☐ 周期性分析

--- 3. Analysis Setup and Execution ---

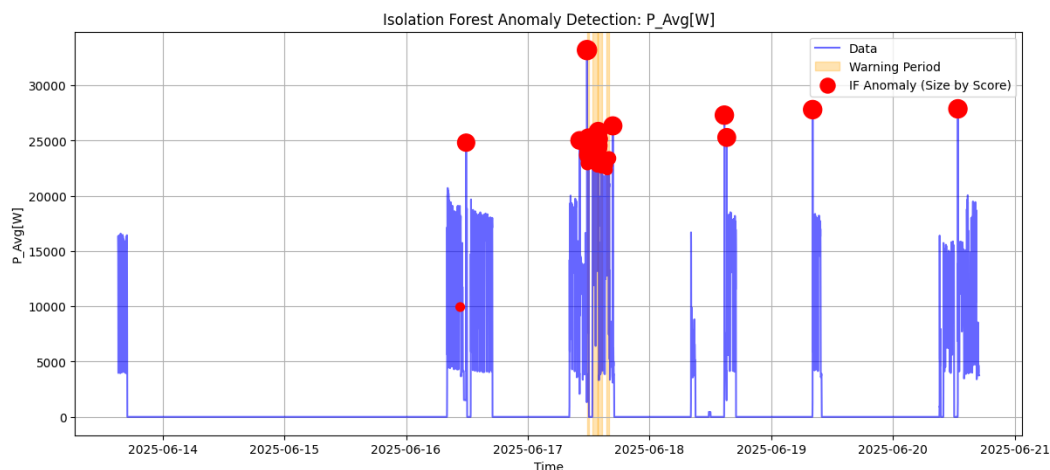
選択した分析の入力設定

--- 4. Result Summary ---

結果を表示

ツール化とユーザーインターフェース

分析結果の提示



11

ツール化とユーザーインターフェース

AIによる分析結果の提示

- ・ 1時間平均電力消費量は、各ファイルで明確な周期的なパターンを示し、ピーク時と谷時の区別が明確です。例えば、ピーク電力消費は通常10時頃に発生し、最低電力消費は0時頃です。
- ・ 1日平均電力消費量にもパターンが見られます。ピーク電力消費は金曜日に発生し、最低電力消費は日曜日に発生する傾向があります。

12

開発環境

- Google Colab

Googleが提供する、ブラウザ上で機械学習・データ分析用のPythonコードを実行できる無料のクラウドベース開発環境



- Google Drive

ファイルの置き場所として使用



- 言語：Python

シンプルで読みやすいオープンソースプログラミング言語。AIに関する多数のライブラリが使用できる。



- Gemini

Googleが提供している生成AIサービス。上記の環境との親和性が高い。



13

セキュリティ対応

ガイドラインの制定

- ガイドラインの制定を実施
- 「学習しない設定」にする
- 生成AIへ入ってはならない情報を決めておく
- ガイドラインに沿って教育を実施

14

まとめ・今後の展望

本研究の成果と実用性

- ・データの周期性や外れ値を捉える手法を用いて、**データの見える化、異常検知や長期傾向の把握**を可能にした
- ・**生成AIを活用**することで、短期間での開発を行うことが出来た

エネルギー利用効率化への貢献

- ・エネルギー消費の無駄やピークの把握が容易になり、効率的な運転計画や改善施策に直接つながれることが期待される

今後の改善や発展の方向性

- ・さらに長期かつ多様なデータを取得・活用し、AI・機械学習との統合によって、自動化や予測精度の向上を高めることで、実運用における意思決定の支援を強化する