

日射量の実測値と計算値の比較

祖父江匠真

1 はじめに

前回, 任意の緯度経度と日時から日射量を計算するプログラムを開発したので, リサイクル館から送信された太陽光発電の日射量のデータとプログラムによって計算された日射量を比較, 評価する.

2 実測値データの取得

リサイクル館から送信された太陽光発電データを保存している Elasticsearch サーバーから任意の日付におけるドキュメントを全て取得するプログラムをソースコード 1 に示す. なお, Elasticsearch サーバーには 1 日あたり約 45000 件のドキュメントが保存されていたため, ソースコード 1 では, ScrollAPI を用いてドキュメントの取得を行っている.

ソースコード 1: Elasticsearch サーバーから任意の日付の環境データを取得するプログラム

```
1 from elasticsearch import Elasticsearch
2 import pickle
3 import datetime
4 import os
5
6
7 def fetchDocsByDatetime(dt, filePath):
8     es = Elasticsearch("http://133.71.201.197:9200", http_auth=("takenaka",
9         "takenaka"))
10
11     indexName = "pcs_recyclekan"
12
13     # すでにPickle ファイルが存在するなら ElasticSearch から取得しない
14     if os.path.isfile(filePath):
15         return
16
17     dt_next = dt + datetime.timedelta(days=1)
18     query = {
19         "query": {
20             "range": {
21                 "JPtime": {
22                     "gte": f"{dt.year}-{str(dt.month).zfill(2)}-{str(dt.day).zfill(2)}T00:00:00",
23                     "lt": f"{dt_next.year}-{str(dt_next.month).zfill(2)}-{str(dt_next.day).zfill(2)}T00:00:00",
```

```

23         },
24     }
25 }
26 }
27
28 num = 10
29 s_time = "2m"
30 data = es.search(
31     index=indexName,
32     scroll=s_time,
33     body=query,
34     size=num,
35     request_timeout=150,
36 )
37
38 s_id = data["_scroll_id"]
39 s_size = data["hits"]["total"]["value"]
40 result = data["hits"]["hits"]
41 while s_size > 0:
42     data = es.scroll(scroll_id=s_id, scroll=s_time, request_timeout=150)
43     s_id = data["_scroll_id"]
44     s_size = len(data["hits"]["hits"])
45     result.extend(data["hits"]["hits"])
46
47 with open(filePath, "wb") as f:
48     pickle.dump(result, f)
49
50 # 内部接続を閉じる
51 es.close()

```

3 実測値と計算値をプロットする

ソースコード 1 によって取得した Elasticsearch サーバーの日射量データと、計算によって求めた日射量データをプロットしたものを図 1 に示す。図 1 は Elasticsearch サーバーから取得した 2022 年 5 月 3 日の日射量データと、リサイクル館の緯度経度と日付情報 (2022 年 5 月 3 日) を入力として求めた日射量の計算値をプロットしている。また、Elasticsearch サーバーから取得したドキュメント数は約 45000 件であったが全てをプロットすると Matplotlib によって表示したグラフ画面がフリーズしてしまったので、1 分辺り 1 件にドキュメントを絞り込んでプロットしている。更に、実測値と計算値の日射量の最大値が等しくなるよう、計算値の縦軸のスケールを実測値と計算値の比率を元に変更している。

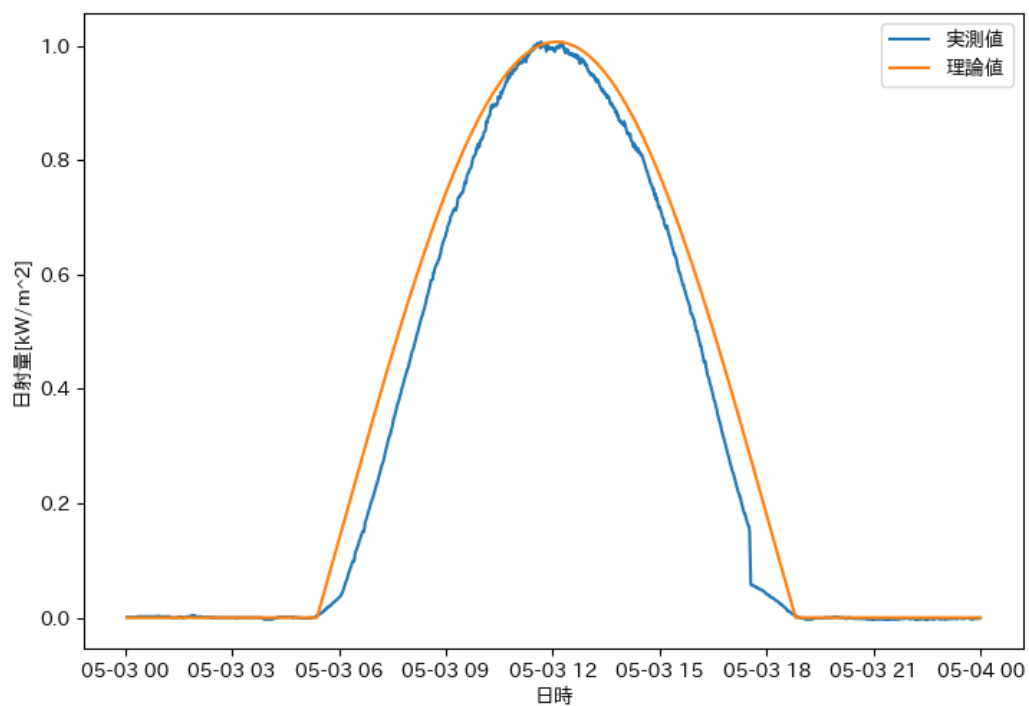


図 1: 2022 年 5 月 3 日の日射量の実測値と計算値をプロットしたもの

次に, 計算値と同じ値を取る実測値の時刻と, 計算値の時刻の差を計算することで, 計算値を用いて時刻を予測した際の予測誤差を各計算値ごとに求めたものを図 2 に示す. 図 2 より, 計算によって求めた日射量による時刻の推定には最大で約 59 分の誤差が生じることが分かった.

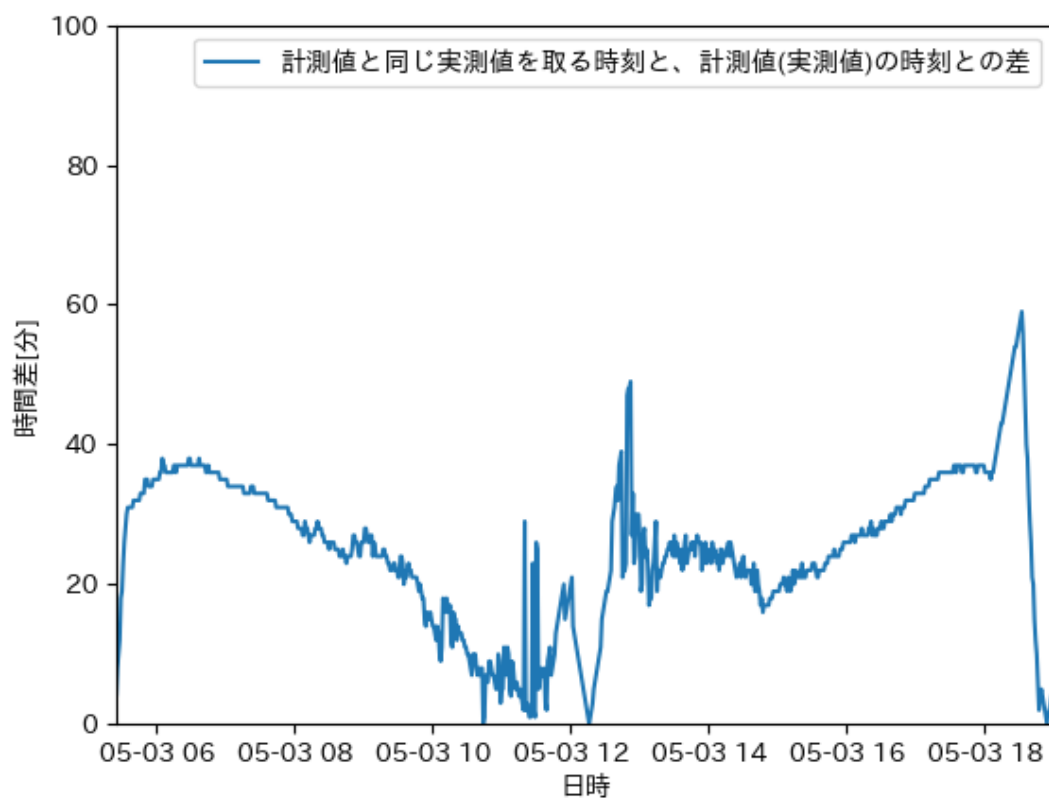


図 2: 計測値と同じ実測値を取る時刻と、計測値の時刻との差

4 おわりに

今回は、実測した日射量のデータと計算によって求めた日射量を比較した。計算値による時刻の推定に最大で約 59 分の誤差が生じていることが分かったので、次回までに日射量の計算式を再検討する。

参考文献

- [1] Blacknon, "Python で Elasticsearch5 から scroll を使って検索結果をすべて取得する", "<https://orebibu.com/ja/home/201803/20180302.001/>", 参照 May 30, 2022.