

天気予報による発電量予測をもとにユーザとセンサーデータのサービスレベルを合意する手法

杉本 一彦¹ 串田 高幸¹

概要: 本稿では、天気予報による発電量予測をもとにセンサーデータの送信間隔をユーザと合意をした上で変更し、消費電力の制御手法を提案する。予測された発電量もとに、IoT デバイスが電池不足とならない消費電力となるセンサーデータ送信間隔を設定し、ユーザへ提示する。発電量予測は IoT デバイスから得られる発電量と OpenWeatherAPI から取得される天気予報の雲量データを用いる。雲量ごとに発電量を分け、各雲量における発電量の確率分布を求めることで雲量データをもとに発電量の予測を行う。実験では実際の発電量と雲量データの関係性や発電量予測の精度について分析を行った。その結果、発電量予測の精度では 1 日前に取得した雲量では誤差 9.06%となり、3 日前では誤差-48.86%となった。このことから 1 日前の天気予報の利用、もしくは 1 日～3 日前の予想発電量の確率分布の和を取ることでより精度の高い動作可能確率をユーザへ提示することができ、センサーデータ送信間隔の変更することが可能となる。

1. はじめに

近年、モノのインターネット (Internet of Things : IoT) は、有益なサービスを提供するのに役立つ膨大な量のデータを生成している [1]。IoT の発展に伴い、さらに多種多様なデータが必要になっていく中で IoT デバイスは急激に増加をしている [2][3]。その中でネットワーク、データ処理 (ビッグデータ)、電力、メンテナンス、輻輳制御、セキュリティ、認証問題、効率的な情報発信、エネルギー効率といった課題が各業界で見受けられるようになってきた [4]。

1.1 背景

IoT デバイスはセンサーモジュールと電子回路で接続されている。図 1 はセンサーデータの一連の流れを示したものである。IoT デバイスはセンサーモジュールからセンサーデータを取得し、インターネットを介してサーバへ送る。送られたセンサーデータはサーバで蓄積され、分析や解析のために利用される。分析や解析をするのはユーザ (エンドユーザ) である。つまり、IoT デバイスの目的はユーザへセンサーデータを届けることである。

IoT デバイスはセンサーデータを 5 分に 1 回や 1 時間に 1 回のような決められた間隔で送信する。そしてセンサーデータの送信間隔はユーザが設定を行う。ユーザは望んでいる間隔やタイミングでセンサーデータが送られてくる事



図 1 センサーデータの一連の流れ

を求めている。そのため IoT デバイスはセンサーデータを正しく送信するために安定し、継続した動作を求められる。また、センサーデータを送る正確性がサービスの信頼性となる。IoT デバイスの監視は必須となる。

IoT デバイスが安定した稼働を行うための 1 つの要素として電力が挙げられる。通常、IoT デバイスは AC 電源に接続されることで安定した電力を得ることを可能にしている。しかし、IoT の発展に伴い IoT デバイスの設置場所は拡大し AC 電源に接続できない場所への設置も求められるようになってきた [5]。そこで、IoT デバイスの電源として 1 次電池や 2 次電池を用いる場合もある。1 次電池や 2 次電池は電力を安定して一定時間供給することは可能であるが、電池残量が 0 となった場合は電池交換や充電作業が必要となる。多くの IoT デバイスに用いることはメンテナンスコストが増加することに繋がる可能性がある [6]。そのため IoT デバイスには外部から電力供給がなされなくても継続した動作をするエコシステムが求められる。図 2 は IoT デバイスの電力エコシステムに必要な 3 つの要素を示す。発電は、外部からの電力供給をせずとも動作するために必要不可欠な要素である。発電は自然エネルギーを用いた発

¹ 東京工科大学コンピュータサイエンス学部
〒192-0982 東京都八王子市片倉町 1404-1

電方法が一般的である。蓄電は発電量が変化しても IoT デバイスへ一定の安定した電力を供給するためのものである。そして溜められた電力を IoT デバイスが消費をする。発電、蓄電、消費のバランスを適切なものにする事で IoT デバイスは外部電源なしに動作することが可能となる。そのためには発電量や蓄電状況の監視、消費電力を制御することが必要となる。

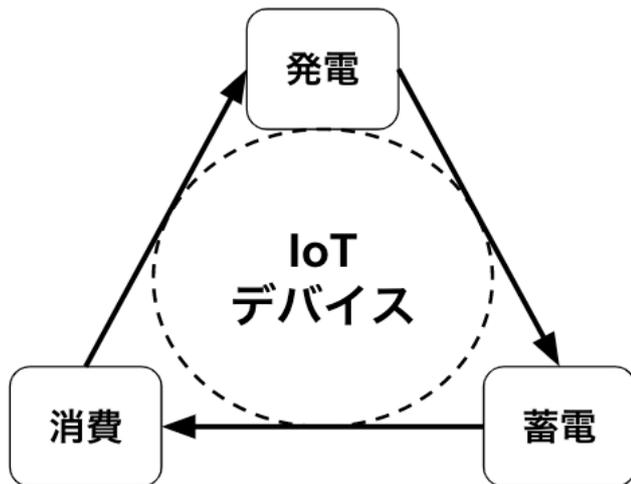


図 2 IoT デバイスの電力エコシステム

本稿では、発電として太陽光パネルを用いる。太陽光発電の発電量は時間や天気に影響される。太陽光を用いて発電するという性質上、収集された発電量は予測できる可能性が低い [7]。そのため継続した運用を目指すとした場合、発電量が変化する中において電池の残量が不足しないよう消費電力制御を含めた電力制御管理が必要となる。

IoT デバイスの消費電力はセンサーデータの送信間隔によって変わる。センサーデータを送信する際はネットワークに接続されており、消費電力は高くなる。例として、マイクロコントローラーである ESP32 の Wifi 接続時の平均消費電力は約 1.0W、スリープ時は約 0.05W である [8]。そのため、IoT デバイスは通常はスリープモードとし、データを送る場合のみネットワークに接続することで消費電力を下げる。しかし、センサーデータの送信間隔が短くなることは、より長い時間ネットワークに接続する事となるため、消費電力の増加へ繋がる [8]。したがって、IoT デバイスの消費電力を制御するためにはセンサーデータの送信間隔を変更する必要がある。

センサーデータの送信間隔はユーザが設定している。そのため、送信間隔の変更にはユーザへ変更の許可や変更の指示、変更の要請をしなくてはならない。さらにその上で、ユーザが変更を加える。つまりユーザの合意が得られなければ送信間隔の変更が行えない。そのためシステムはユーザとセンサーデータの送信間隔を予め合意の上設定することが必要になる。

1.2 課題

ユーザとセンサーデータの送信間隔を決定する場合、現状ではセンサーデータの送信間隔を含めた動作スケジューリングの決定手順が考えられていない場合が多い [9]。また IoT デバイスにおける太陽光発電を用いた電力パフォーマンスの改善を述べる研究では、センサーデータの取得間隔が静的でかつシステムによって設定されておりユーザの考慮がなされていない [10][11]。したがって、実際に IoT デバイスを動作させる場合において、ユーザはそもそもセンサーデータの送信間隔の設定ができず、送信間隔はシステムによって変更されてしまう。このような場合ではサービスとして成り立たず、ユーザが利用することにはならない。

2. 関連研究

太陽光発電を用いた IoT によるアプローチでは、公共の天気予報に基づく IoT デバイスでの太陽エネルギーに関する予測を行っている論文がある [11]。簡単に入手できる公共の気象データに基づき、複数の機械学習手法を用いて太陽光発電の予測について調べている。しかし発電量の予測を行っているものの、ユーザを考慮した IoT デバイスの動作スケジューリングや消費電力についての記載がない。また、IoT デバイスのエネルギー問題に関するアプローチでは、エネルギーハーベスティングセンシングでバッテリーを再考する論文がある [12]。非充電式バッテリーを用いて IoT デバイスを最初は動作させていたが、合理的なサイズの一次電池で達成可能なセンサノードの寿命が短いため、動作を維持するためにノードのために頻りに電池を交換する必要があるとしている。その解決策としてエネルギーハーベスティングに焦点を当てている。IoT デバイスでのバッテリー問題に着目しているものの、具体的な稼働方法や、エネルギー管理方法について記述されていない。IoT デバイスの省電力化について、処理の増加による消費電力増加を抑えるため処理の分散手法を述べている [13]。バッテリー依存を抑えるためにソーラーハーベスティング (太陽光発電) を用いて駆動させることを目的としている論文である [14]。太陽光発電を用いたものであるが、恒久的な動作時間の延長ではなく、あくまで一時的な動作時間の延長としており長期間動作としての電力管理は提案されていない。IoT デバイスにおける太陽光発電の利用については先行研究がなされており電力パフォーマンスの改善を提案しているが、最も重要なコンポーネントであるユーザーに対して焦点を当てていない。

3. 提案

ユーザとシステムにてセンサーデータの送信間隔を合意した上で決定する。そのために、システムはユーザに対し「定められた期間において、定められたセンサーデータの送信間隔を送ることが出来る確率」を提示する。これを本稿

では「センサーデータの送信可能確率」と呼ぶものとする。例として、「今から3日後まで、10分間隔でセンサーデータの送信を行うと80%の確率で送信することが出来る」といったようなものである。

センサーデータの送信可能確率は今後発電される電力量を予測を行い求め出す。発電量の予測は天気予報を用いる。そして、センサーデータの送信間隔は予測された発電量とIoTデバイスの電力状況(2次電池の残量, 発電量)を考慮した上で適切な消費電力となるように複数個設定され、システムからユーザへ提示される。したがって、ユーザは提示された複数のセンサーデータの送信間隔から最も求めているものに近い間隔を選択することでシステムと送信間隔を合意する。そしてシステムは設定されたセンサーデータの送信間隔での送信可能確率を満たす。

全体の流れ

提案における全体の流れを図3へ示す。図3にて示されている数字は下記にて説明をする。

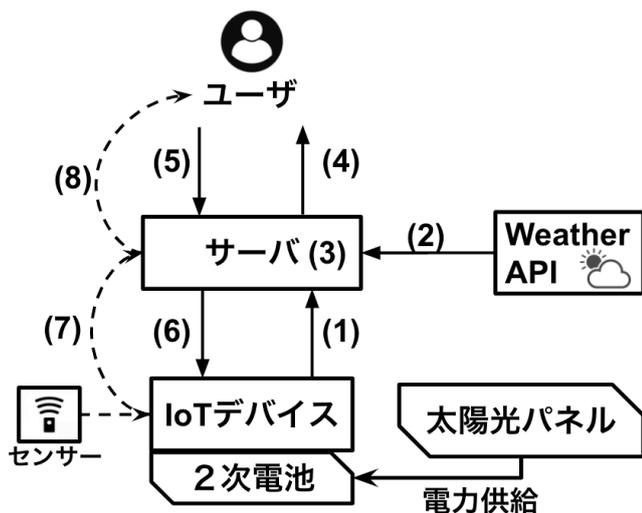


図3 全体アーキテクチャ図

- (1) IoTデバイスからサーバへ発電量, 2次電池の残量が送信される。
- (2) 天気予報をWeatherAPIから取得する。
- (3) サーバにて天気予報から発電量の予測を行い, IoTデバイスの2次電池の残量をもとにセンサーデータの送信間隔を複数決定する。
- (4) サーバにてセンサーデータの送信間隔をユーザへ提示する。
- (5) ユーザはサーバから提示されたセンサーデータの送信間隔の中から1つを選択する。
- (6) ユーザが選択したセンサーデータの送信間隔をIoTデバイスへ設定する。
- (7) IoTデバイスは設定された送信間隔でサーバへセンサーデータを送信する。

- (8) サーバからユーザへセンサーデータが送信される。

3.1 センサーデータ送信間隔のユーザとの合意

サーバはセンサーデータの送信間隔の変更をする際には必ずユーザからの合意を得る必要がある。サーバが定めた複数のセンサーデータ送信間隔の中からユーザが選択する、もしくはユーザが任意の送信間隔を設定することでユーザが合意したとみなす。ユーザが選択できる項目は2つある。“センサーデータ送信間隔”と“センサーデータ取得日数”である。センサーデータ送信間隔はセンサーデータをユーザへ送信する間隔である(例:1分間に1回,1時間に1回)。センサーデータ取得日数はセンサーデータをどれほどの期間取得したいか設定する項目である。また、システムは設定された送信間隔でデータを送信する期間をユーザへ保証する。

3.2 天気予報を用いた発電量予測を基にしたセンサーデータ送信間隔の決定

IoTデバイスにて取得された発電量をもとに今後の発電量の予測を行う。これはIoTデバイスによって発電の特徴が違うことにより、一律した予測を行うことができないため1台1台のIoTデバイスでの予測とする。したがって一律で予測より精度の高い予測となる。また発電量予測には天気予報を用いるが本稿では雲量データをもとに予測を行う。そして予測された発電量からユーザへ提示するためのセンサーデータ送信間隔の決定を行う。具体的な手順を図4へ示す。

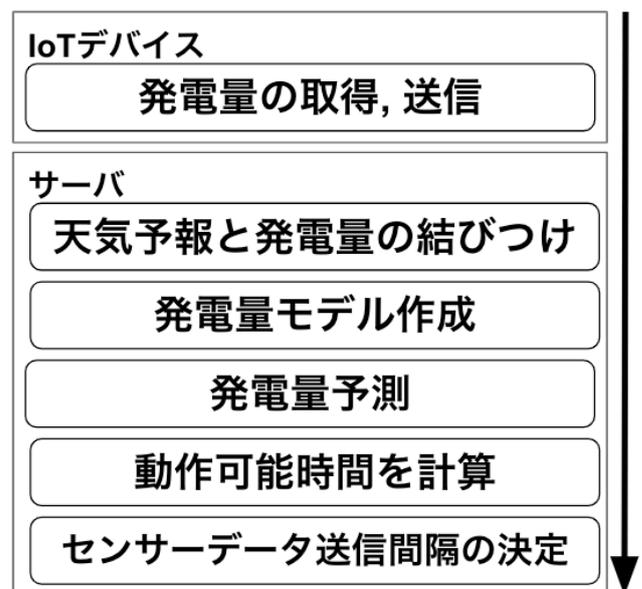


図4 センサーデータ送信間隔の決定手順

発電量の取得, 送信はIoTデバイスにて行う。発電量の取得は電力計, もしくは当該機能を持っているセンサーモ

ジュールで行う。IoT デバイスにて取得後、サーバへの送信を行う。

天気予報と発電量の結びつけでは、IoT デバイスで取得され送信された発電量と、取得した同時刻での天気予報との関係性を求めるために天気予報データと発電量データを1つのデータセットとする。図5はデータセットの方法を示している。IoT デバイスから発電量 $P1$ が時刻 $T1$ で

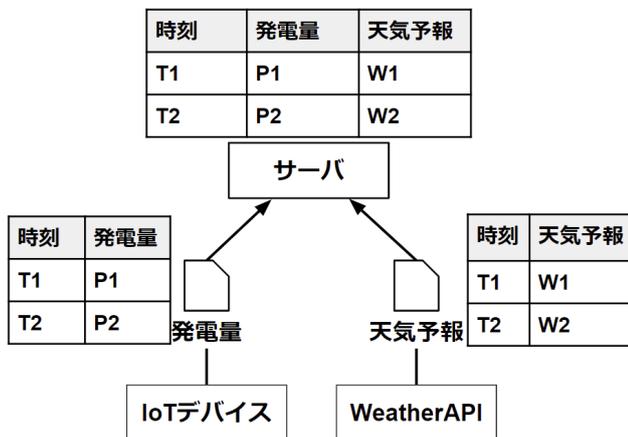


図5 発電量と天気予報のセット化

取得された場合、サーバは WeatherAPI にて発電量が取得された同時刻 $T1$ の天気予報 $W1$ を取得する。そしてサーバにて時刻 $T1$ の発電量 $P1$ と天気予報 $W1$ をセット化しデータベースへと格納される。同様に時刻 $T2$ で取得された発電量 $P2$ 、天気予報 $W2$ もデータセットとしてデータベースへ格納される。発電量が取得された際、サーバは常にこの処理を行う。

サーバにて発電量と天気予報のデータセット作成後、雲量ごとに発電量を区別し、特徴量を求め出す。特徴量は雲量ごとに分けられた発電量の確率分布をとることで求め出す。発電量は太陽光の照射量に依存するため、雲量と発電量が必ずしも一致する事や常に同じ発電量となることはない。したがって「発電される可能性」をモデルとして定義しておくことにより、確率として発電量を予測する。

発電量予測は天気予報の雲量データと雲量ごとに分けられた発電量の確率分布を用いて行う。予測される発電量は一定値ではなく、今までの発電量データに基づく確率値となる。そのため予測発電量の最大値と最小値の幅は、予測期間が長くなるに従い広がっていく。発電量予測の疑似コードを下記に示す。

- 発電量予測の疑似コード

```

1 Set prediction_model to probability
  distribution
2 Set h to number of estimated time
3 prediction_probability = list()
4 for counter = 0 to number of elements in
  prediction_model at h do

```

```

5   for ... (Repeat as many times as h) do
6     add to the prediction_probability
7   end do
8 end do

```

疑似コードでは、発電量の確率分布モデルを用いて予測する時間分の計算を行っている。発電される可能性を全て足し合わせて計算し、その結果を再度確率分布とすることで予測する時間分の合計発電量の予測を行う。

動作可能時間はIoT デバイスが稼働し続けることのできる時間のことを指す。この数値が低いほどより短い期間しか動作することができない。つまり、2次電池の残量が少ない場合や予測発電量が低い場合となる。逆に数値が高いほど、2次電池の残量が多い場合や予測発電量が高い場合となり、より長期間動作することが可能となる。しかしあくまで予測値であるため必ずしもこの数値が正しいとは限らない。動作可能時間の計算では、予測された発電量と現在の2次電池残量、またIoT デバイスでの消費電力の3つの要素を考慮し、算出する。現在の2次電池の残量を V 、予測発電量の合計を P 、IoT デバイスの1時間の平均消費電力を C 、動作可能時間 T は以下の計算式で求める。

$$T = \frac{V + P}{C}$$

センサーデータ送信間隔の決定は、ユーザに提示するセンサーデータの送信間隔の候補を決める処理である。ユーザはこの決定された候補の中から送信間隔を設定し、システムと合意をしたことになる。予測された動作可能時間より、ユーザが求めるセンサーデータの送信期間内にIoT デバイスが電力不足で動作が停止しないような消費電力となる送信間隔が決定される。しかし、予測されている動作可能時間も予測発電量によるものであるため、予測されている最小値、もしくは最小値以下の発電量になった場合IoT デバイスが停止する可能性が存在する。そのため、ユーザが指定した期間内におけるセンサーデータ送信可能確率も同時に提示する必要がある。

4. 実装

4.1 実装

ここでは提案を基にした実装について記述する。全体の構成図を図6に示す。IoT デバイスとしてESP32を用いる。ESP32に電力供給をしているモジュールは寸法 $188 \times 251 \times 17\text{mm}$ の太陽光パネルと 4000mAh のリチウムイオン電池である。また発電量とリチウムイオン電池の電源電圧の取得には電力計センサーのINA226PRCを用いた。サーバにはUbuntu18.04、データベースにはMongoDB4.5を用いた。天気予報の取得にはOpenWeatherAPIを利用した。ESP32とサーバはネットワークにより接続されている。ユーザはサーバに立てられているWebサーバ(Apache2)内に設けられているブラウザからアクセスする。

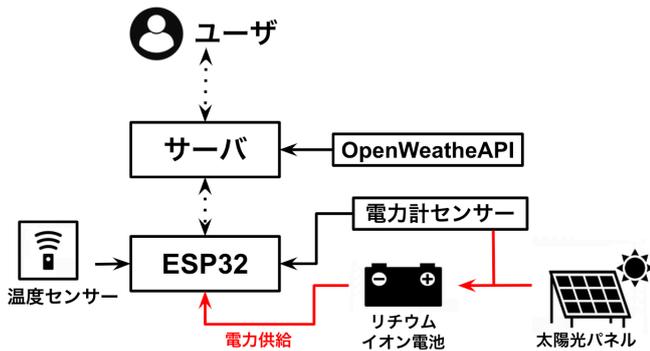


図 6 全体の実装構成図

4.2 ユーザインタラクション

ユーザは、まずリクエストを受け付けるサーバに対して、どのセンサーからどの頻度でデータを送信するかを設定する。サーバはそのリクエストを受け取った後、3.2で提案した手法を用いてIoTデバイスの現在の動作可能時間にユーザリクエストを反映して計算をし、設定されたリクエストでIoTデバイスがどれほどの確率でセンサーデータを送信できるかを5日後先まで計算し、ユーザへ提示する。

ユーザを操作画面を図7と図8に示す。図7はユーザがセンサーデータの受信間隔を設定する際の操作画面である。ここではシステムから提示された送信間隔を選択する。しかしユーザの要求が選択肢以外になることを想定し、任意の受信間隔を設定することが可能である。またセンサーデータの取得期間の設定では1日～5日間を選択できる。これは本実装で天気予報の取得を最大5日後までとしており、5日後以降の発電量予測が行えないためである。

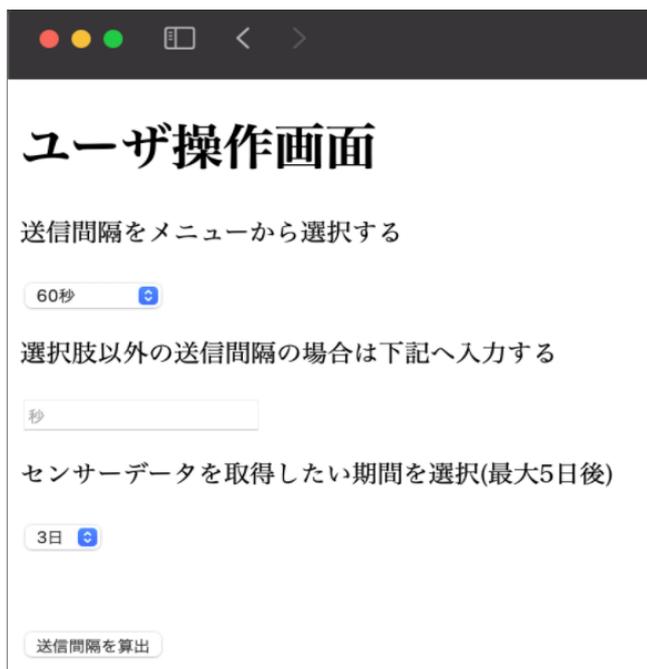


図 7 ユーザ操作画面 (センサーデータ受信間隔設定)



ユーザ操作画面

入力された送信間隔は 60 秒です
取得日数の期間は 3日です

データ送信確率は80%です

▼ 詳細はこちらをクリック

日数	天気予報	送信確率
1日後	曇り	50%
2日後	曇り	75%
3日後	晴れ	90%
4日後	快晴	100%
5日後	晴れのち曇り	85%

この間隔で設定する やり直す

図 8 ユーザ操作画面 (送信可能確率提示)

4.3 発電量予測

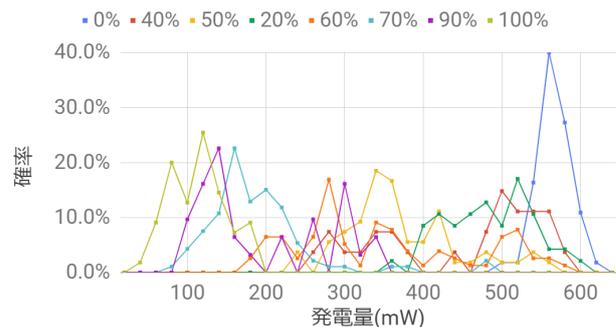


図 9 雲量ごとに分けられた発電量の確率分布

図9は12時台に計測された発電量を雲量ごとに分け、確率分布としたものである。発電能力は最大で約600mWであり、雲量の数値が上がるにつれて低下していく。図9は12時台のみの確率分布だが、実際は発電量を1時間ごとに区切り、それぞれの時間帯にて図9のような雲量ごとに分けられた発電量の確率分布が出されている。なお相関係数は-0.86である。

図10は発電量予測の計算モデルを示している。図10に示されているH1~H4の意味は、H1が予測1時間目における発電量確率分布、H2が予測2時間目における発電量確率分布といったように、予測時間によって雲量や時間帯にあった確率分布を使用する。例えば、6時間の合計発電量の予測を行う場合にはH1~H6までとなる。そして発電される可能性を全て算出するために、確率分布のデータ全ての足し合わせを毎時間行う。そして、結果を確率分布にすることで最終的な発電量の合計も確率として求め出す。

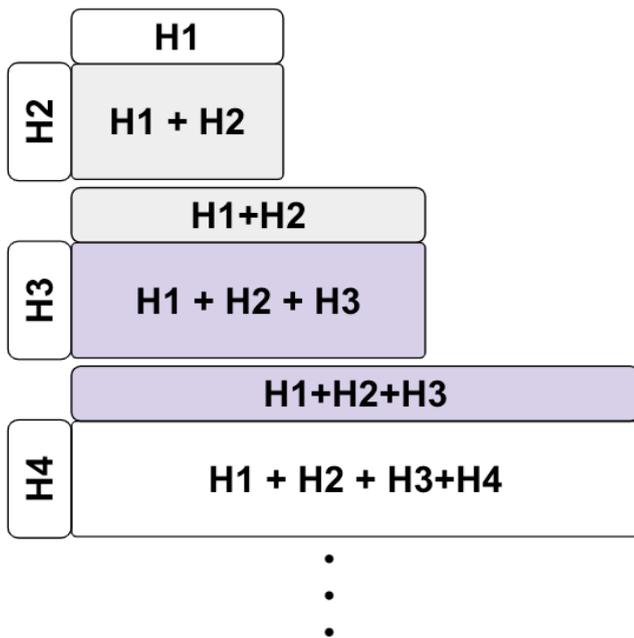


図 10 発電量予測計算モデル

4.4 バッテリー残量の算出

2次電池として 4000mAh のリチウムイオン電池を用いる。ESP32 の動作可能時間を求めるために現在のバッテリー残量を算出する必要がある。リチウムイオン電池の電源電圧が実際の静電容量と比例するものではないため定数を与えて測定することはできない。リチウムイオン電池の電圧の降下の特性から、バッテリー残量を求め出す。図 11 は本実験で用いたリチウムイオン電池を最大まで充電した後、ESP32 が電力不足で停止する電圧まで低下していく電圧のグラフである。図 11 のグラフは一定に低下するのではなく、傾きに緩急がある。電圧の値から電池残量を求め出すためには、このグラフを線形回帰させることで電源電圧から残量を求める。

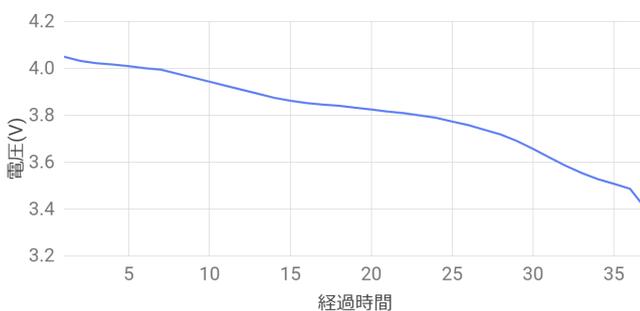


図 11 実験にて用いたリチウムイオン電池の電圧と時間の関係

5. 評価と分析

5.1 評価方法

評価方法は以下の方法で実施する。

- (1) ESP32 から発電量とバッテリー出力電圧を計測し、

サーバへ送信する。

- (2) 発電量と天気予報の雲量データから発電量の予測を行う。
- (3) 実際の発電量と予測発電量との誤差を求め出す。

5.2 評価と分析

発電量予測

OpenWeatherAPI からは 4 日前から 3 時間毎の天気予報の取得が可能である。表 1~表 3 は図 12~図 14 の発電量予測を行うための雲量を示している。1 日前~3 日前にかけて取得した雲量となっている。

表 1 2020 年 12 月 23 日の 10:00~16:00 の雲量

日時	10:00~	12:00~	15:00~
1 日前雲量	0	0	0
2 日前雲量	8	29	88
3 日前雲量	27	64	95

表 2 2021 年 1 月 15 日の 9:00~16:00 の雲量

日時	9:00~	12:00~	15:00~
1 日前雲量	73	76	79
2 日前雲量	77	78	79
3 日前雲量	75	98	98

表 3 2021 年 1 月 27 日 12:00~15:00 の雲量

日時	12:00~	15:00
1 日前雲量	18	73
2 日前雲量	99	98
3 日前雲量	100	100

図 12 は 2020 年 12 月 23 日の 10:00~16:00、図 13 は 2021 年 1 月 15 日の 9:00~16:00、図 14 は 2021 年 1 月 27 日 12:00~15:00 において予測された発電量の合計を確率分布で示したものである。黄色面が 1 日前、赤色面が 2 日前、青色面が図 12 では 3 日前、図 13 と図 14 では 3、4 日前の天気予報から求め出された発電量の合計である。また緑色の縦棒は当日の同じ時間に実際の発電量の合計である。

それぞれのグラフでは、天気予報の取得した日にちが離れるほど予測発電量が下がっていくことが分かる。これは OpenWeatherAPI の天気予報を求める過程における何かしらの影響がある。そして、日にちが近づくほど実際に発電された電力へと分布が近寄っていることが分かる。

図 15 は図 12~図 14 における実際の発電量と予測された確率分布における最大確率時での発電量との誤差を示している。平均誤差は 1 日前が 9.06%、2 日前が -29.63%、3 日前が -48.86%であった。2 日前、3 日前で誤差が大きく広がるのが分かる。また 1 日前になると実際に発電される電力と比べ予測される発電量が多い傾向にあることが分かる。

発電量予測を行うにあたり、より精度の高い予測を行う

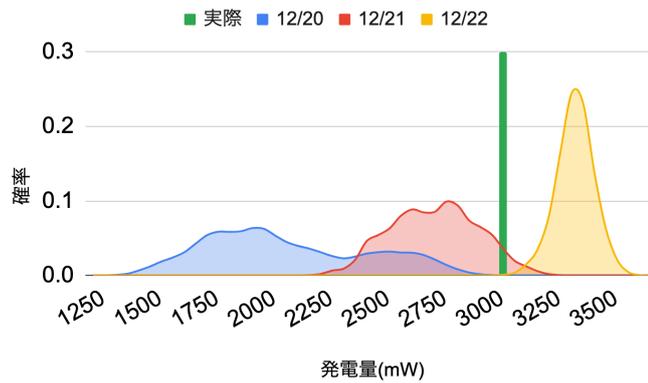


図 12 2020 年 12 月 23 日 10:00~16:00 での予測された発電量の合計

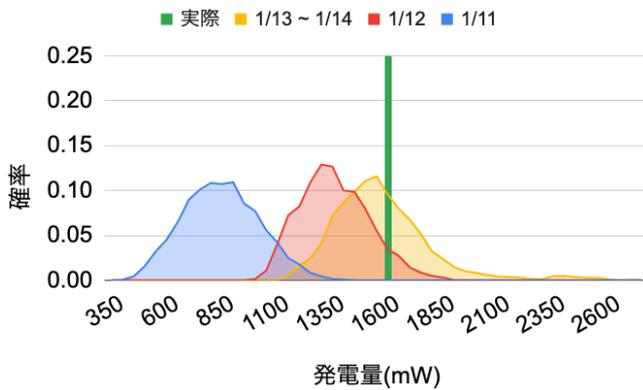


図 13 2021 年 1 月 15 日 9:00~16:00 での予測された発電量の合計

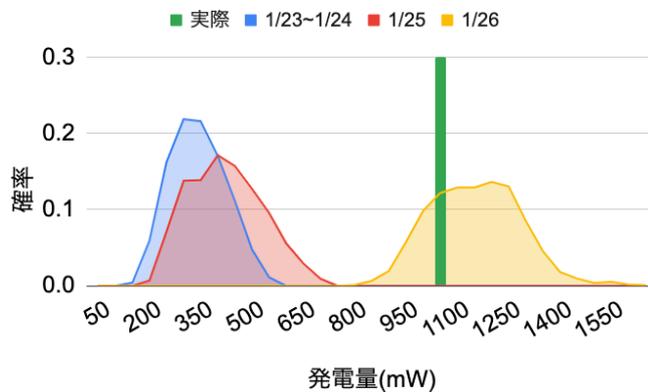


図 14 2021 年 1 月 27 日 12:00~15:00 での予測された発電量の合計

ためには 1 日前の天気予報データを用いることが望ましい。2 日前, 3 日前である場合, 誤差範囲が大きいものとなり, 実際の発電量と約 1000mW 違うといった場合も見受けられる。そのため最低でも 1 日に 1 回は天気予報の更新を行い, 最新のデータでの予測を行うことで, 精度を高めていくことに繋がる。

6. 議論

本研究では太陽光パネルを IoT デバイスの電力供給源として消費電力の制御や発電量の天気予報からの予測や, 太陽光パネルを用いることで発生するユーザへのインタラ

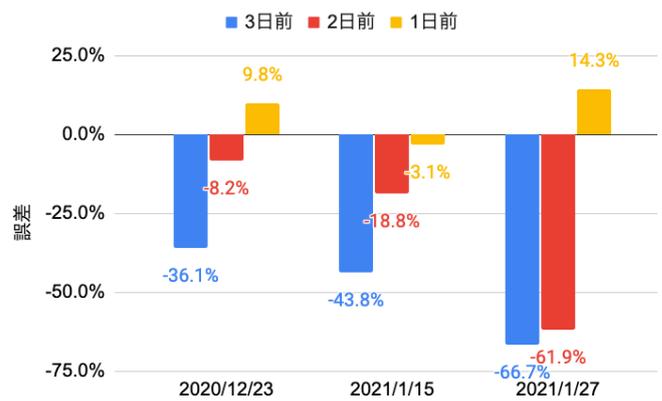


図 15 実際の発電量と最大確率時での予測発電量との誤差

クションの改善についての提案を行った。IoT の発展において電力供給の問題については課題であるということは様々な論文でも取り上げられ, IoT デバイスの増加に伴う消費電力の増加や, 設置場所の制限について議論がかわされている [15]。自然エネルギーである太陽光発電を IoT に活用する仕組みは先行研究でもいくつか取り上げられている [16][17][18][19]。しかし, 小型 IoT デバイスでの, より具体的な消費電力やバッテリー残量, 太陽光パネルの発電量の監視まで行っている論文は少ない。また, 本研究では太陽光パネルを利用することで発生するユーザへのアプローチも目を向ける事で, より具体的なサービスを構築する上で重要なユーザ目線提案を行っている。これは他の論文でもユーザに目を向けているものは少ない。

本提案では, ユーザと IoT デバイスが 1 対 1 である関係の場合のみを想定したが, 実際にはユーザが 1 に対して IoT デバイスが多 (1 対多), ユーザが多で IoT デバイスが多 (多対多) であるモデルの方が一般的である。IoT デバイスが多である場合, ユーザが各々の IoT デバイス 1 つ 1 つに対してセンサーデータ送信間隔を設定することは非常に負荷が高いことになる。そのため, センサーデータの種類や設置した場所やバッテリー残量でグルーピングした上でまとめて操作することで負荷を下げる事ができ, ユーザへの負担を少なくまとめてセンサーデータの取得を行うことが可能となる。

7. おわりに

本稿では, 天気予報から発電量予測を行い, その予測をもとにセンサーデータの送信間隔をユーザとの合意の上で変更する手法を提案した。これらの提案は IoT デバイスを太陽光発電にて稼働させるために, ユーザが望むセンサーデータ送信間隔を最大限対応することと IoT デバイスが電力不足にならないよう発電量を予測し, 適切な消費電力制御をすることを可能にした。ユーザに提示するセンサーデータ送信可能確率を求めるための発電量予測については実際の発電量との誤差が天気予報を取得するタイミングに

より変化することが分かり、平均誤差は1日前が9.06%、2日前が-29.63%、3日前が-48.86%であった。1日前では精度が10%以内に収まっており、予測へ大きな影響は無いものの、2日前より後であると精度が途端に悪くなる。本稿では雲量のみからの予測であったが、雲量のみならず太陽高度や季節、予想計算モデルの改善を行うことでより精度の高い予測が行える。

8. 謝辞

本研究は、JSPS 科研費 JP20K11776 の助成を受けたものです。

参考文献

- [1] Chi, Q., Yan, H., Zhang, C., Pang, Z. and Xu, L. D.: A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment, *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1417–1425 (2014).
- [2] Bellavista, P., Cardone, G., Corradi, A. and Foschini, L.: Convergence of MANET and WSN in IoT Urban Scenarios, *IEEE Sensors Journal*, Vol. 13, No. 10, pp. 3558–3567 (2013).
- [3] Zanella, A., Bui, N., Castellani, A., Vangelista, L. and Zorzi, M.: Internet of Things for Smart Cities, *IEEE Internet of Things Journal*, Vol. 1, No. 1, pp. 22–32 (2014).
- [4] Alamri, A., Ansari, W. S., Hassan, M. M., Hossain, M. S., Alelaiwi, A. and Hossain, M. A.: A Survey on Sensor-Cloud: Architecture, Applications, and Approaches, *International Journal of Distributed Sensor Networks*, Vol. 9, No. 2, p. 917923 (2013).
- [5] Wan, T., Karimi, Y., Stanačević, M. and Salman, E.: Perspective Paper—Can AC Computing Be an Alternative for Wirelessly Powered IoT Devices?, *IEEE Embedded Systems Letters*, Vol. 9, No. 1, pp. 13–16 (2017).
- [6] Jolly, B.: The Last Thing IoT Device Engineers Think About: End of Battery Life Behavior for IoT Devices, *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 837–840 (online), DOI: 10.1109/MWSCAS.2019.8885125 (2019).
- [7] Chu, M., Li, H., Liao, X. and Cui, S.: Reinforcement Learning-Based Multiaccess Control and Battery Prediction With Energy Harvesting in IoT Systems, *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp. 2009–2020 (2019).
- [8] Bekaroo, G. and Santokhee, A.: Power consumption of the Raspberry Pi: A comparative analysis, *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, pp. 361–366 (2016).
- [9] Vestenicky, M., Matuska, S., Hudec, R. and Kamencay, P.: Sensor network proposal based on IoT for a prediction system of the power output from photovoltaic panels, *2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–4 (online), DOI: 10.1109/RADIOELEK.2018.8376390 (2018).
- [10] Ahn, H. K. and Park, N.: Deep RNN-Based Photovoltaic Power Short-Term Forecast Using Power IoT Sensors, *Energies*, Vol. 14, No. 2 (2021).
- [11] Kraemer, F. A., Ammar, D., Braten, A. E., Tamkittikhun, N. and Palma, D.: Solar Energy Prediction for Constrained IoT Nodes Based on Public Weather Forecasts, *Proceedings of the Seventh International Conference on the Internet of Things*, New York, NY, USA, Association for Computing Machinery (2017).
- [12] Jackson, N., Adkins, J. and Dutta, P.: Reconsidering Batteries in Energy Harvesting Sensing (2018).
- [13] Kul, B. and Şen, M.: Energy saving IoT-based advanced load limiter, *2017 XXVI International Scientific Conference Electronics (ET)*, pp. 1–5 (2017).
- [14] Gummeson, J.: Energy Harvesting is Charging Up, Vol. 22, No. 4 (2019).
- [15] Sahraei, N., Watson, S., Sofia, S., Pennes, A., Buonassisi, T. and Peters, I. M.: Persistent and adaptive power system for solar powered sensors of Internet of Things (IoT), *Energy Procedia*, Vol. 143, pp. 739–741 (2017). Leveraging Energy Technologies and Policy Options for Low Carbon Cities.
- [16] Spanias, A. S.: Solar energy management as an Internet of Things (IoT) application, *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, pp. 1–4 (2017).
- [17] Shrihariprasath, B. and Rathinasabapathy, V.: A smart IoT system for monitoring solar PV power conditioning unit, *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, pp. 1–5 (2016).
- [18] Deshmukh, N. S. and Bhuyar, D. L.: A Smart Solar Photovoltaic Remote Monitoring and Controlling, *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 67–71 (2018).
- [19] Kaur, N. and Sood, S. K.: An Energy-Efficient Architecture for the Internet of Things (IoT), *IEEE Systems Journal*, Vol. 11, No. 2, pp. 796–805 (2017).