

気温の変動に応じたデータ送信頻度の増減による IoTデバイスの省電力化

五味 滉人¹ 中川 翔太² 串田 高幸¹

概要: コンセントに接続できない場所に置いている IoT デバイスは、一次電池を用いて電力供給を行い、稼働している。しかし、一次電池は再充電を行えない電池であり、電池残量がなくなった際に交換しなければならず、交換する手間が発生してしまう。その問題を解決するために本研究では、一次電池から電力供給をしている IoT デバイスの電力消費量を削減することを目的とし、IoT デバイスに取り付けられた温度センサーから取得した気温をサーバーに送信して、送信した気温をもとにそれ以降の1時間ごとの気温を予測する。予測した気温を用いて、データの送信頻度を減らすアルゴリズムを提案する。この提案では、気温をサーバーに送信する回数を抑制すること、気温をサーバーに送信してから次に気温を測定するまでの時間を Deep Sleep モードにすることで消費電力を削減した。実験結果では、ビニールハウス内外の予測気温精度の適合率は、実測気温と比較して、ビニールハウス内では 80.2%、ビニールハウス外では 72.6% になった。また、提案方式を適応した場合の消費電力量と Deep Sleep モードなしでコネクションを維持続けた場合の消費電力量を比較した結果、ビニールハウス内では 94.4%削減できた。ビニールハウス外では 93.8%削減することができた。

1. はじめに

背景

IoT(Internet of Things) は、インターネットを介して通信でき、物理的な世界を感知し、観測データを送信し、作動する [1]。近年 IoT の普及が進んでおり、McKinsey の推定では、2025 年までに 250 億~500 億台のデバイスが存在し、経済効果は年間 3.9 兆~11.1 兆ドルの範囲になると推定されている [2]。

IoT で最も研究されている分野は、世界人口が急速に増加している中、食の安全を確保するために非常に重要である農業分野である。農業現場のセンサーから生成されたデータはデータ分析に利用することで、農業の収穫量を向上することができ、農業にまつわる省力化やノウハウ継承の問題を解決することができる [3]。農業における IoT の主なアプリケーションは、精密農業、畜産、温室で、監視領域にグループ化されている。これらのアプリケーションは、IoT デバイスを介して土壌、空気、気温、湿度、水といった関連データを収集するのに役立つワイヤレスセンサーネットワークを使用して監視を行っている。 [4]

センサーネットワークノードは、充電できないバッテリー(一次電池)を利用していたが、一次電池ではセンサーノードの寿命が短いため、ノードの動作を維持するために頻繁に一次電池を交換する必要があった [5]。一次電池とは、一度完全に放電してしまったら捨ててしまうことになる、使い切りのタイプでマンガン乾電池やアルカリ乾電池のことを指す。IoT アプリケーションでは、センサーが単 4 電池 1 本で何年も稼働できるシステムが理想的である。一般的に、接続機器の無線技術を選択する際には、消費電力、最大の距離範囲を考慮する必要がある [6]。

IoT デバイスは一般にコンセントに接続されていないため、バッテリーや環境から得られる電力で存続しなければならない、IoT デバイスは一般的に長い寿命を必要とするため、消費電力がさらに制限されてしまう [7]。本研究で用いている IoT デバイスの ESP32 の消費電力の一覧を表 1 に示す。例として、ESP32 の Wi-Fi 接続時に平均 100mA 消費した場合、4000mAh バッテリーは 40 時間しか接続できない。

表 1 ESP32 の消費電力

カテゴリ	消費電力 (mW)
Wi-Fi 接続時	330
Wi-Fi 非接続時	142
Deep Sleep	30

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

² 東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻
〒192-0982 東京都八王子市片倉町 1404-1

課題

本研究の課題図を図1に示す。図1では、IoT デバイスは一次電池から電力を受け取っている。IoT デバイスにはセンサー類が取り付けられていて、気温や電力を取得している。取得しているデータをサーバーに送信している。IoT デバイスにおいて無線通信を行う際の消費電力は大きいので、無線通信を頻繁に行うと電池残量を消費して、動作することができなくなる。そのため、長期間運用するためには、無線通信を削減する必要がある。例としてビニールハウス内にIoT デバイスを設置して気温を管理している時に、消費電力を抑えるために無線通信の頻度を減らした場合、センサーからデータを取得し、サーバーとの無線通信する頻度が減ってしまい、管理が適切に行えない。消費電力を削減するためには、無線通信を減らすこととIoT デバイスの待機時に Deep Sleep モードにするといった対策が考えられる。IoT デバイスには、継続的に電力を供給する必要があるため、コンセントに接続して電力供給を行っている。一次電池を用いている場合、再充電することはできず、電池残量がなくなった際に交換する手間やコストがかかってしまう。本研究の課題を解決するために、IoT デバイスの消費電力を削減する手法が必要になる。

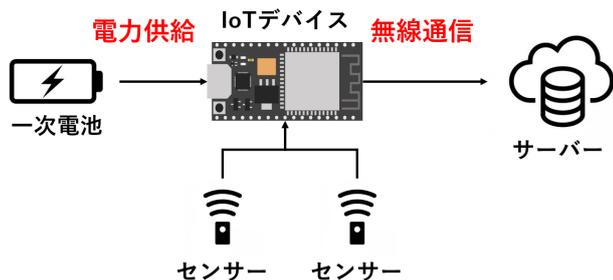


図1 課題図

各章の概要

第2章では、本研究の関連研究について説明する。第3章では、第1章で説明した課題を解決するシステムの提案について説明する。第4章では、本研究で提案するシステムの実装と実験環境について説明する。第5章では、本研究で提案したシステムの評価と分析について説明する。第6章では、本研究で提案したシステムの議論を行う。第7章では、本研究のまとめを行う。

2. 関連研究

Wail Mardini らは、IoT デバイスのデータによって引き起こされる電力消費に役立つ提案した [8]。この提案では、IoT デバイスの新しい読み取り値が古い読み取り値と異なる場合にのみ、IoT デバイスがデータを CoAP(Constrained Application Protocol) サーバーに新しい読み取り値を送信

することができる。論文では、バッテリーの容量が限られていることを課題としている。実験では、標準的な CoAP プロトコルに対して 33 % から 155 % のデバイスの寿命の延長を可能とした。しかし、デバイスが故障してデータを送れていないのか判断できない問題がある。

Mukesh Taneja らは、M2M/IoT デバイスとゲートウェイノードの消費電力を削減するためのフレームワークを提案した [9]。この提案では、IoT アプリケーション、IoT ゲートウェイ、およびデバイスでデータをバッファリングして、デバイスとゲートウェイノードを可能な限りスリープモードに保っている。論文では、アクティブモードの電流が数十 mA でスリープモードが数 μA になり、アクティブモードのエネルギー消費量と電力消費量が多いことを課題としている。しかし、使用されるワイヤレスアクセステクノロジーに新しい同期要件を導入することができない問題がある。

Ashish Kumar Sultania らは、NB-IoT(Narrowband IoT) の省電力スキームの ns-3 設計および実装の詳細を紹介している [10]。この紹介では、NB-IoT の省電力スキームの ns-3 設計と実装の詳細を示し、デバイスのエネルギー消費とレイテンシーを評価している。論文では、バッテリーで動作しているデバイスの消費電力を削減し、デバイスの寿命を延ばすことを課題としている。しかし、タイマーとパラメーターを設定するための 3GPP 規格で提起されている手順が含まれていないことや、メッセージベースでの実装ができていないという問題がある。

Ben Abdel Ouahab Ikram らは、実際の IoT から収集したデータに基づいて、周辺の温度を予測できるネットワークを構築し、訓練する提案した [11]。この提案では、超短記憶 (LSTM) アーキテクチャを持つリカレントニューラルネットワーク (RNN) を採用し、周辺の環境を予測している。しかし、この提案では、エネルギー問題を考慮できていない。

Ahmad F. Subahi らは、センサー機器の使用における IoT の新しい進歩に基づいて、内部の温室温度を監視および制御するインテリジェントなエネルギー効率 (EE) システムの構築を提案した [12]。この提案では、一貫した成長環境を構築するだけでなく、システム全体を自動化し、エネルギーと生産コストを節約している。論文では、消費エネルギーが大きいことを課題としている。しかし、処理するデータを分割できていないので、エネルギー消費が大きくなる問題がある。

3. 提案方式

本研究は、一次電池から電力供給をしている IoT デバイスの電力消費量を削減することが目的とし、温度センサーから取得した気温をサーバーに送信して、送信した気温を

もとに1時間ごとの予測点を決め、それ以降のデータの送信頻度を減らすアルゴリズムを提案する。

基礎実験

提案を確定する際に実際のビニールハウス内外の気温を知る必要があるため基礎実験としてビニールハウス内外の気温の測定を行った。図2に2021年11月15日(晴れ)のビニールハウス内外の気温グラフ、図3に2021年11月16日(晴れ)のビニールハウス内外の気温グラフ、図4に2021年11月19日(晴れ)のビニールハウス内外の気温グラフを示す。縦軸は気温、横軸は時刻である。グラフの結果からビニールハウス内の気温の方がビニールハウス外の気温より高くなっていることがわかった。また、14時から16時の間に最高気温になっている。また最高気温になるまで上昇傾向で、最高気温になってから緩やかな下降傾向になることがわかった。

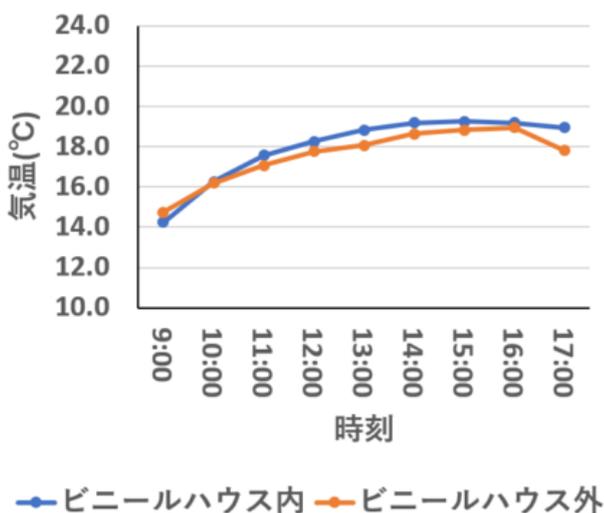


図2 ビニールハウス内外の気温 (2021年11月15日(晴れ))

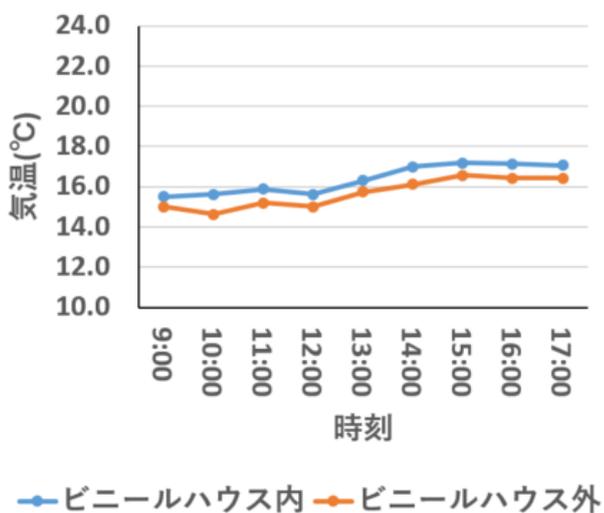


図3 ビニールハウス内外の気温 (2021年11月16日(晴れ))

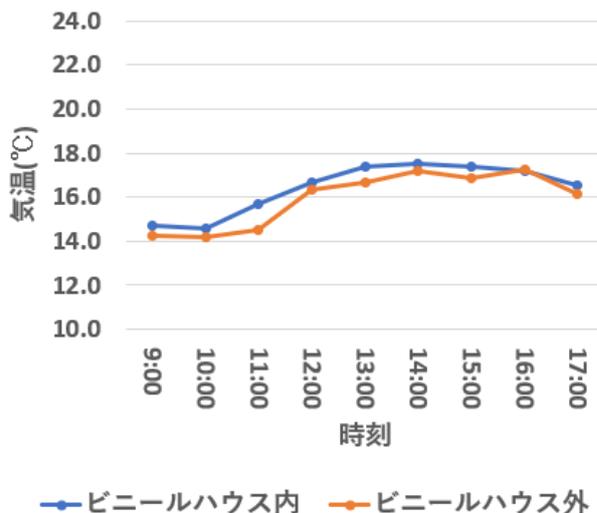


図4 ビニールハウス内外の気温 (2021年11月19日(晴れ))

次に、センシング頻度を基礎実験から決定する。表2,3,4,5に2021年11月15日(晴れ)の1時間間隔と30分間隔のビニールハウス内外の気温と上がり幅を示す。午前中は上がり幅が高いが気温が低い。また午後には上がり幅が温度センサーの気温計測精度である $\pm 0.5^\circ\text{C}$ の範囲内である事が多い。30分間隔では、ビニールハウス内が $\frac{13}{16}$ 、ビニールハウス外が $\frac{11}{16}$ の確率で上がり幅が温度センサーの気温計測精度である $\pm 0.5^\circ\text{C}$ の範囲内であった。そのため、センシング頻度を1時間と決定した。

表2 1時間間隔のビニールハウス内気温

時刻	ビニールハウス内の気温 (°C)	前の時間からの上り幅 (°C)
9:00	14.3	
10:00	16.3	2.0
11:00	17.6	1.3
12:00	18.3	0.7
13:00	18.8	0.6
14:00	19.2	0.4
15:00	19.3	0.1
16:00	19.2	-0.1
17:00	18.9	-0.3

表3 1時間間隔のビニールハウス外気温

時刻	ビニールハウス内の気温 (°C)	前の時間からの上り幅 (°C)
9:00	14.8	
10:00	16.2	1.4
11:00	17.1	0.9
12:00	17.8	0.7
13:00	18.1	0.3
14:00	18.6	0.6
15:00	18.8	0.2
16:00	18.9	0.1
17:00	17.8	-1.1

表 4 30 分間隔のビニールハウス内気温

時刻	ビニールハウス中気温 (°C)	前の時間からの上り幅 (°C)
9:00	14.3	
9:30	15.5	1.3
10:00	16.3	0.8
10:30	17.3	1.0
11:00	17.6	0.3
11:30	17.8	0.2
12:00	18.3	0.5
12:30	18.6	0.3
13:00	18.8	0.3
13:30	18.9	0.1
14:00	19.2	0.3
14:30	19.3	0.1
15:00	19.3	0.0
15:30	19.4	0.1
16:00	19.2	-0.2
16:30	19.1	-0.1
17:00	18.9	-0.2

て比較する。予測気温と実測気温の差が閾値内の場合、再度 Deep Sleep モードにする。予測気温と実測気温の差が閾値外の場合、未送信部分の実測気温、消費電力、日時を IoT デバイスからサーバーに送信し、測定時間用の予測気温を気温モデルを用いて再決定する。

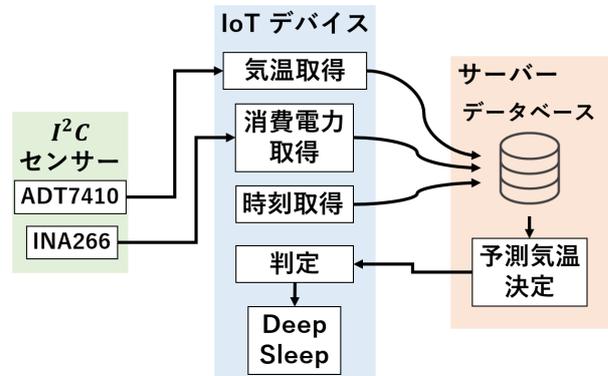


図 5 全体構成図

表 5 30 分間隔のビニールハウス外気温

時刻	ビニールハウス外の気温 (°C)	前の時間からの上り幅 (°C)
9:00	14.8	
9:30	15.8	1.1
10:00	16.2	0.4
10:30	17.3	1.1
11:00	17.1	-0.2
11:30	17.3	0.2
12:00	17.8	0.5
12:30	17.7	-0.1
13:00	18.1	0.4
13:30	18.0	-0.1
14:00	18.6	0.6
14:30	18.9	0.3
15:00	18.8	-0.1
15:30	19.0	0.2
16:00	18.9	-0.1
16:30	18.4	-0.6
17:00	17.8	-0.6

本提案で用いる気温モデルは、過去に測定した 9 時から 17 時までの気温を取得し、1 時間ごと構築する。構築した気温モデルをサーバー上に構築してあるデータベースに保存している。気温モデルは、予測気温を決定するときに用いる。

気温モデルが保存されているデータベースの例を表 6 に示す。気温が保存されているデータベースは時刻ごとにテーブルを作成している。データベースの更新は、実測気温と予測気温が閾値外の場合に、IoT デバイスからサーバーに未送信部分の気温を送信してモデルに追加する。

表 6 気温モデルが保存されているデータベース例 (9 時)

日時	気温 (°C)
11-24	11.5
11-25	15.0
11-26	13.5
...	...

提案方式

システムの全体構成図を図 5 に示す。IoT デバイスでは、取り付けた I2C センサーの ADT7410 から気温、INA226PRC から電流と電圧を取得し消費電力を求めている。また、気温と消費電力を取得した日時を保存する。取得した気温、消費電力、日時を IoT デバイスからサーバーに送信し、サーバー上に構築してあるデータベースに IoT デバイスから送信されたデータを保存する。保存した気温をもとに測定時間用の予測気温を気温モデルを用いて決定し、次の測定時刻まで IoT デバイスを Deep Sleep モードにする。そして測定時間になったら気温を測定し、実測気温と予測気温の閾値はセンサー精度の ± 0.5 °C を閾値とし

本提案の流れとして、測定を始める時間で 100 回気温を取得し、最頻値を求めてこの値をその時間の気温とする。100 回気温を取得して最頻値を求める理由として、温度センサーの外れ値やばらつきを抑える必要があるからである。取得した気温を IoT デバイスからサーバーに送信する。送信した気温とあらかじめデータベースに保存してある気温モデルを用いて 1 時間ごとの予測気温を決定する。決定した予測気温をサーバーから IoT デバイスに送信し保存する。次の気温取得時間まで Deep Sleep モードにし、次の気温取得の時間になったら Deep Sleep モードを解除し、気温を取得し、実測気温と予測気温とを比較する。実測気温と予測気温の閾値はセンサー精度の ± 0.5 °C を閾値として比較する。実測気温と予測気温の差が閾値内の場合、実測

表 7 モデル選択

		9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
モデル (°C)	11 月 19 日	14.7	14.6	15.7	16.7	17.4	17.5	17.4	17.2	16.6
	11 月 24 日	13.1	13.8	14.7	15.4	16.1	16.3	16.3	15.4	14.8
実測値 (°C)	11 月 25 日	12.1	13.3	14.4	...					

気温をファイルに書き込み、次の気温取得時間まで Deep Sleep モードにする。実測気温と予測気温の差が閾値外の場合、未送信部分の実測気温、消費電力、日時を IoT デバイスからサーバーに送信し、予測気温を再決定し、サーバーから IoT デバイスに送信し保存してから Deep Sleep モードにする。

予測気温は IoT デバイスからサーバーに送信された気温と気温モデルを用いて決定する。気温モデルは、実測日より 1 ヶ月前までの 9 時から 17 時までの気温を用いる。1 ヶ月前までの気温モデルを用いる理由として、実測日に近い気候や気温変化の傾向が似ていると考えているため用いる。9 時の気温が IoT デバイスからサーバーに送信された時、9 時の気温とあらかじめデータベースに保存されている 9 時の気温モデルを比較して、気温が一番近い日の一日の気温のを予測気温とする。9 時の気温と近い気温を予測気温として選択する理由は、気温変化の傾向が似ていると考えているため選択している。

予測気温の決定手順を以下に示す。

- IoT デバイスで気温を取得し、サーバーに送信
- サーバーに送った実測気温とサーバー上に構築してあるデータベースに保存している気温のモデルを比較する。比較は、最新の時刻の実測気温と同時刻の各気温モデルの値の差を求める。差が最小である気温モデルの時刻ごとの気温を予測気温とする

- 予測気温を IoT デバイスに送信し、保存する

予測気温のモデル選択を表 7 に示す。表 7 では、11 月 25 日を实測日、11 月 19 日、24 日がモデルとなっている。例として 11 月 25 日 9 時の実測気温が 12.1 °C だった場合、9 時の気温のモデルと比較して、一番近い気温は 11 月 24 日の気温になる。9 時以降の予測気温は 11 月 24 日の各時間の気温を予測気温とする。

ユースケース・シナリオ

ユースケースシナリオ図を図 6 に示す。ユースケースとして、図のような電力供給を持ち合わせてないビニールハウスが挙げられる。電力供給が難しい理由として、設備にかかるコストや時間がかかってしまうからである。作物の例として、レタスをビニールハウスで水耕栽培する場合、気温を 15 °C 以上 25 °C 以下の範囲に保つ必要があり、IoT デバイスに温度センサーを取り付け、サーバーに温度データを送信してサーバー側で温度管理を行う。IoT デバイスに電力供給が行えないビニールハウスでは、一次電池を使

用することになり、電池の交換がユーザにコストや手間がかかるようになる。したがって、消費電力を抑えることで、長期間運用することができ、一次電池の交換する手間とコストを抑えることができる。

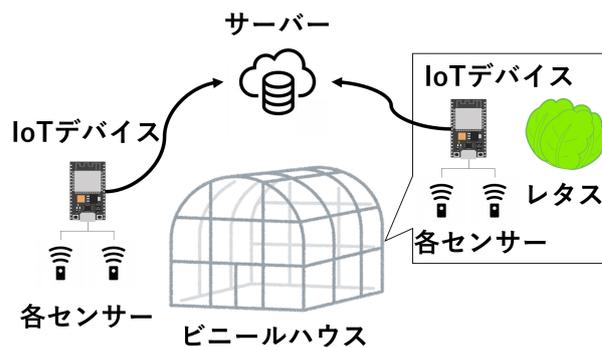


図 6 ユースケース

4. 実装と実験方法

実装

図 7 にソフトウェアの処理の流れを示す。

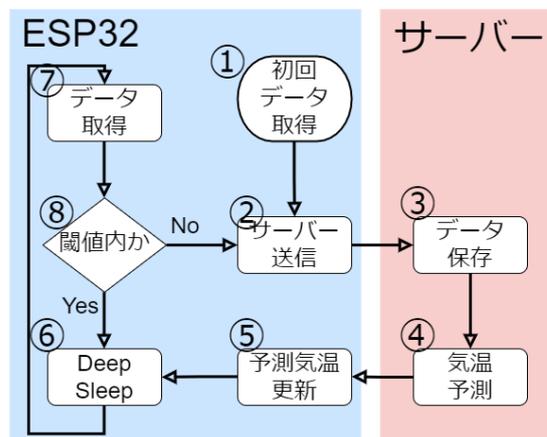


図 7 ソフトウェアの処理の流れ

図 7 に示されているソフトウェアの流れを以下に示す。

- I2C センサーの ADT7410 から気温、INA226PRC から電流・電圧を取得し電流・電圧を用いて消費電力、日時を取得する
- ①で取得した気温・消費電力・日時を ESP32 からサーバーに送信
- ESP32 から送信された気温・消費電力・日時をサーバー上に構築したデータベースに保存する

- ④ データベースに保存した気温をもとに、その日の1時間ごとの予測気温を決め、ESP32に送信する
- ⑤ ④で決定した予測気温をESP32で受け取る
- ⑥ 次の測定時間までDeep Sleepモードにする
- ⑦ 測定時間にDeep Sleepモードを解除し、I2CセンサーのADT7410から気温、INA226PRCから電流・電圧を取得し電流・電圧を用いて消費電力、日時を取得する
- ⑧ センサーの精度の ± 0.5 を閾値として、⑦で取得した気温と予測気温を比較し、閾値内の場合には、ESP32内に気温・消費電力・日時を保存し、Deep Sleepモードにする。閾値外の場合には、再度ESP32からサーバーに未送信部分の気温・消費電力・日時を送信する。

図7に示しているソフトウェアそれぞれの説明を以下に示す。

- 初回データ取得、データ取得
IoTデバイスに取り付けたセンサーから気温、電流・電圧データから消費電力を求め取得する役割である。本研究では、気温を取得できるADT7410から気温を取得、電流・電圧データを取得できるINA226PRCから電流・電圧を用いて消費電力、日時を取得するものとする。
- サーバー送信
センサーから取得した気温・消費電力・日時をESP32からサーバーに送信する役割である。
- データ保存
ESP32から送られてきた気温・消費電力・日時をデータベースに保存する役割である。本研究で用いているデータベース構成例を表8に示す。このデータベースはMySQLで構築している。

表8 データベース構成例

日時	気温 (°C)	消費電力 (mW)
2021-11-22 9:00	11.5	0.60
2021-11-22 10:00	12.5	0.65
...

- 予測気温決定
予測気温はIoTデバイスからサーバーに送信された気温と気温モデルを用いて決定する役割である。3章の提案方式で述べたアルゴリズムが実装されている。
- 予測気温更新
予測した気温をサーバーからESP32に送り、受け取る役割である。
- Deep Sleep
気温を取得して次の気温を取得するまでIoTデバイスのDeep Sleepモードにし、消費電力を抑える役割である。
- 閾値内か
サーバーから送られてきた時刻ごとの予測気温をIoT

デバイス内に保存して、実測気温と比較する役割である。センサーの精度の ± 0.5 °Cを閾値とする。予測気温と実測気温の差が閾値内の場合には、IoTデバイス内で実測気温をテキストファイルに書き込む。予測気温と実測気温の差が閾値外の場合には、サーバーに未送信部分の実測気温、消費電力、日時を送り、予測気温を再度絞りなおす。

実験環境

本研究の実験用のハードウェアを表9に示し、ソフトウェアを表10に示す。本研究のIoTデバイスとしてESP-WROOM-32を使用した。ファームウェアとしてMicroPython1.13をインストールしている。センサーモジュールは温度データを取得できる温度センサーのADT7410と電流・電圧・電力データを取得できるINA226PRCを使用した。ビニールハウスはSOCKER 90191726を使用した。サーバーはUbuntu 20.04.2 LTSを使用した。データベースにMySQLを使用した。

表9 ハードウェア

カテゴリ	ハードウェア
IoT デバイス	ESP-WROOM-32
センサーモジュール	ADT7410
センサーモジュール	INA226PRC
ビニールハウス	SOCKER 90191726

表10 ソフトウェア

カテゴリ	ソフトウェア
ESP-WROOM-32	MicroPython1.13
サーバー	Ubuntu 20.04.2 LTS
データベース	MySQL 8.0.25-0ubuntu0.20.04.1

実験に使用した構成を図8に示す。実験はビニールハウス内外にIoTデバイスとADT7410とINA226PRCを置き、気温と消費電力を取得する。



図8 実験環境

5. 評価と分析

本研究では、予測気温の精度、消費電力量を評価する。

予測気温精度の適合率

ビニールハウス内の実測気温と予測気温との適合率を図9に示し、ビニールハウス外の実測気温と予測気温との適合率を図10に示す。縦軸に予測気温と実測気温の適合率、横軸に実測気温を取得した日付である。評価では、提案通りに実測気温と予測気温が閾値外だった場合に予測気温を変更している。予測気温は実測日以前の計測データをモデルとして利用している。ビニールハウス内では、平均一致率は80.2%になった。また、ビニールハウス外では、平均一致率は72.6%となった。図9、図10より一致率が低い日があることがわかる。これはモデル数が少ないため、低くなっている。モデル数が増えることによって一致率を上げることができると思う。

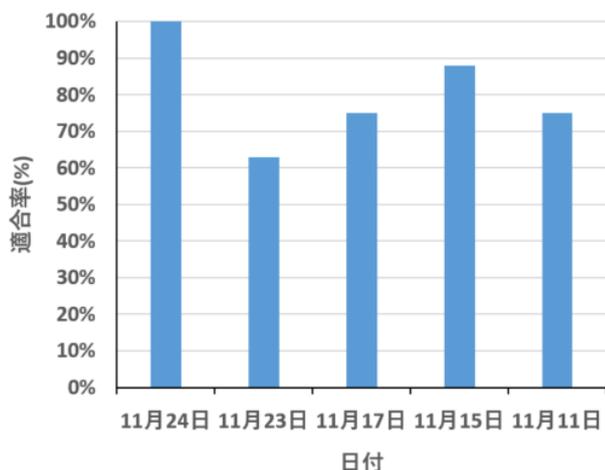


図9 ビニールハウス内の一致率

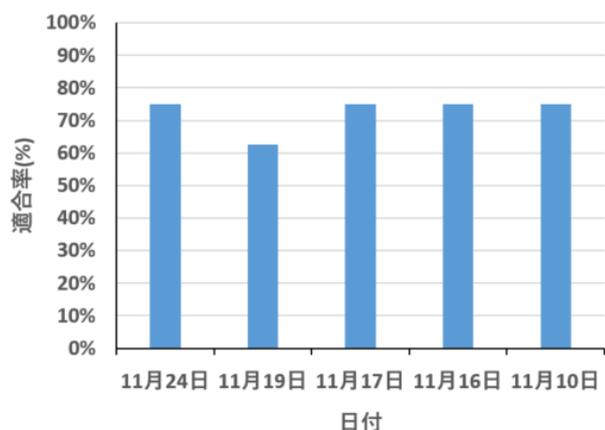


図10 ビニールハウス外の一致率

消費電力量

サーバーとの無線通信回数ごとの消費電力量を図11に示し、サーバーとの無線通信回数1回目と2回目以降の無線通信回数の削減率を表11に示す。図と表に記載している消費電力量は8時間分の消費電力量となっている。図と表から消費電力量はサーバーとの通信回数によって上昇していることがわかる。

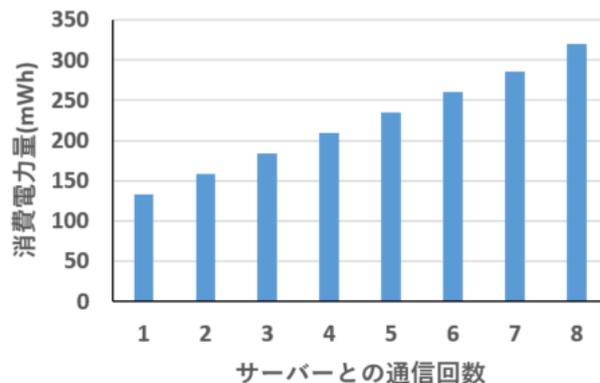


図11 無線通信ごとの消費電力量

表11 無線通信回数の削減率

比較対象 (回数)	削減率 (%)
1 と 2	16
1 と 3	28
1 と 4	37
1 と 5	44
1 と 6	49
1 と 7	54
1 と 8	58

また、Deep Sleep モードなしで接続を維持し続けた場合の消費電力量は2640mWhとなっている。実験結果でサーバーとの無線通信回数は1~3回であった。無線通信が1回の際は132.8mWhの消費電力量と比較すると、Deep Sleep モードなしで接続を維持し続けた場合(2640mWh)では95.0%削減している。無線通信が2回の際は158.4mWhの消費電力量と比較すると、Deep Sleep モードなしで接続を維持し続けた場合(2640mWh)では94.0%削減している。無線通信が3回の際は184mWhの消費電力量と比較すると、Deep Sleep モードなしで接続を維持し続けた場合(2640mWh)では93.0%削減している。平均すると、Deep Sleep モードなしで無線通信を常時行っている場合は94.0%削減している。

また、ビニールハウス内で提案方式を適応した場合の消費電力量(148.2mWh)とDeep Sleep モードなしで接続を維持し続けた場合の消費電力量(2640mWh)と比較すると、94.4%削減している。ビニールハウス外で提

案方式を適応した場合の消費電力量 (163.8mWh) と Deep Sleep モードなしでコネクションを維持し続けた場合の消費電力量 (2640mWh) を比較すると, 93.8%削減している。

6. 議論

本研究では, 温度センサーから取得した気温をサーバーに送信して, 送信した気温をもとに1時間ごとの予測点を決め, それ以降のデータの送信頻度を減らし, 気温をサーバーに送信してから次に気温を計測するまで Deep Sleep モードに切り替えるアルゴリズムを提案した。しかし, 本研究の提案では, 課題となる点も残されている。

まず, 季節ごとの気温の変化についてである。季節によって気温の変化が異なっていて, 季節ごとの気温変化の差を確認できていない。今後は, 基礎実験からわかっている14~16時まで気温が上昇傾向, それ以降は下降傾向になることと過去の気温がわかるサイトを用いて再検証していく必要がある。

次に, 予測気温のためのモデルの量についてである。本研究では, 予測気温を決めるために, 過去に計測した1日の気温をモデルとして, 予測気温を決定している。しかし, 気温は一日に複数計測することができないため, モデルの量が少ない。今後は, 計測日数を増やし, モデルが増えた場合の予測気温と実際の気温の正解率がどのように変化するのか, 再検証していく必要がある。

7. おわりに

本研究の課題として, IoT デバイスが充電不可能な一次電池で動作している場合, 徐々に電池残量が減ってしまい, IoT デバイスに安定した電力供給が行えなくなり動作できなくなることを課題とした。本研究の課題を解決するために, 温度センサーから取得した気温をサーバーに送信して, 送信した気温をもとに1時間ごとの予測気温を決め, それ以降のデータの送信頻度を減らし, 気温をサーバーに送信してから次の気温を測定するまで Deep Sleep モードに切り替えるアルゴリズムを提案した。実験結果では, ビニールハウス内外の予測気温精度の適合率は, 実測気温と比較して, ビニールハウス内では80.2%, ビニールハウス外では72.6%になった。また, 提案方式を適応した場合の消費電力量と Deep Sleep モードなしでコネクションを維持続けた場合の消費電力量を比較した結果, ビニールハウス内では94.4%削減できた。ビニールハウス外では93.8%削減することができた。

参考文献

[1] Fizza, K., Banerjee, A., Mitra, K., Jayaraman, P. P., Ranjan, R., Patel, P. and Georgakopoulos, D.: QoE in IoT: a vision, survey and future directions, *Discover Internet of Things*, Vol. 1, No. 1, pp. 1–14 (2021).

[2] Zhang, J., Li, G., Marshall, A., Hu, A. and Hanzo, L.: A new frontier for IoT security emerging from three decades of key generation relying on wireless channels, *IEEE Access*, Vol. 8, pp. 138406–138446 (2020).

[3] Doshi, J., Patel, T. and Kumar Bharti, S.: Smart Farming using IoT, a solution for optimally monitoring farming conditions, *Procedia Computer Science*, Vol. 160, pp. 746–751 (2019).

[4] Farooq, M. S., Riaz, S., Abid, A., Umer, T. and Zikria, Y. B.: Role of IoT technology in agriculture: A systematic literature review, *Electronics*, Vol. 9, No. 2, p. 319 (2020).

[5] Jackson, N., Adkins, J. and Dutta, P.: Reconsidering batteries in energy harvesting sensing, *Proceedings of the 6th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*, pp. 14–18 (2018).

[6] Mahmoud, M. S. and Mohamad, A. A.: A study of efficient power consumption wireless communication techniques/modules for internet of things (IoT) applications (2016).

[7] Blaauw, D., Sylvester, D., Dutta, P., Lee, Y., Lee, I., Bang, S., Kim, Y., Kim, G., Pannuto, P., Kuo, Y.-S. et al.: IoT design space challenges: Circuits and systems, *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*, IEEE, pp. 1–2 (2014).

[8] Mardini, W., Yassein, M. B., AlRashdan, M., Alsmadi, A. and Amer, A. B.: Application-based power saving approach for IoT CoAP protocol, *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*, pp. 1–5 (2018).

[9] Taneja, M.: A framework for power saving in IoT networks, *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, pp. 369–375 (2014).

[10] Sultania, A. K., Delgado, C. and Famaey, J.: Implementation of nb-iot power saving schemes in ns-3, *Proceedings of the 2019 Workshop on Next-Generation Wireless with ns-3*, pp. 5–8 (2019).

[11] Ikram, B. A. O., Abdelhakim, B. A., Abdelali, A., Zafar, B. and Mohammed, B.: Deep Learning architecture for temperature forecasting in an IoT LoRa based system, *Proceedings of the 2nd International Conference on Networking, Information Systems & Security*, pp. 1–6 (2019).

[12] Subahi, A. F. and Bouazza, K. E.: An intelligent IoT-based system design for controlling and monitoring greenhouse temperature, *IEEE Access*, Vol. 8, pp. 125488–125500 (2020).